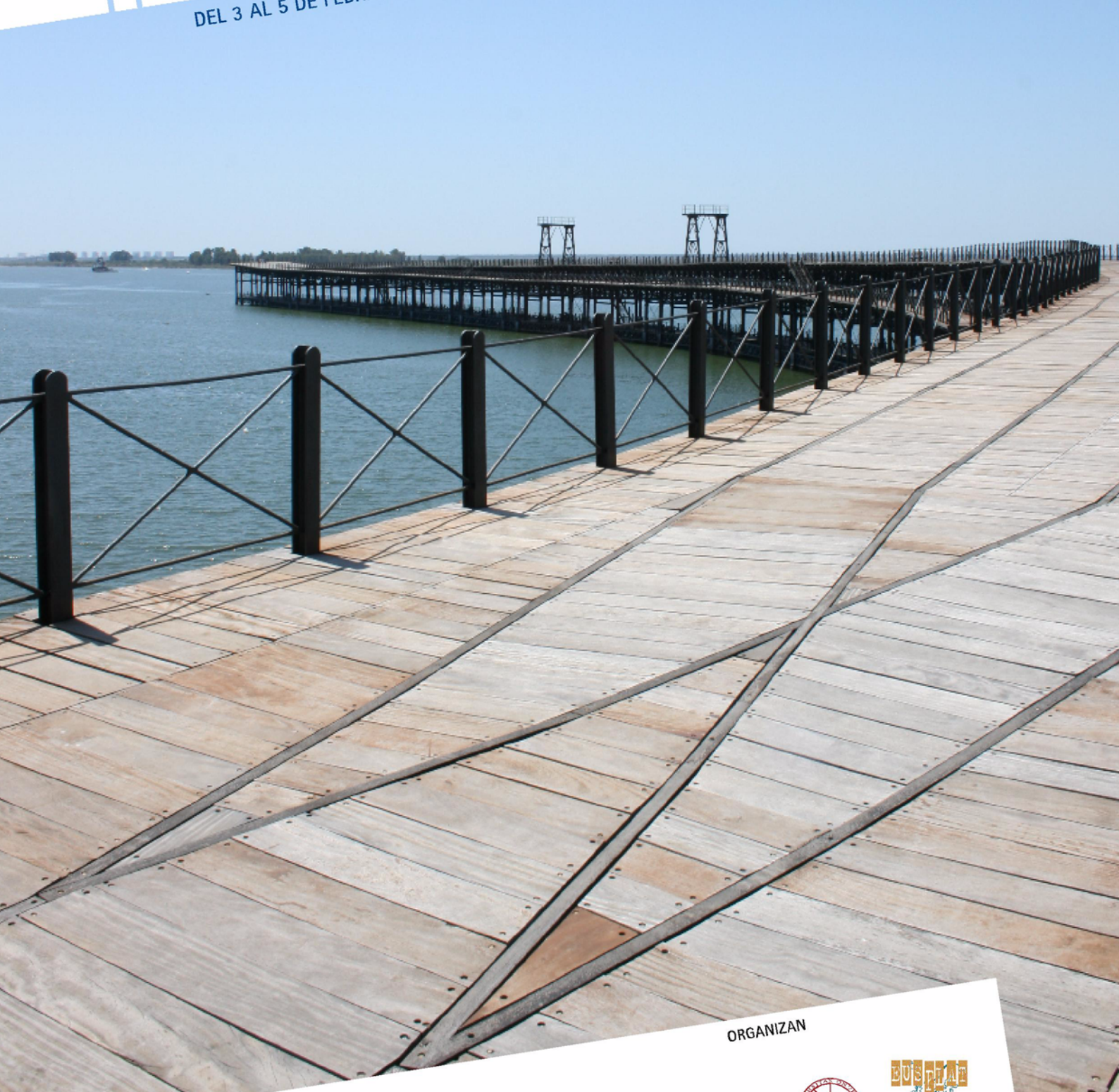


XV CONGRESO ESPAÑOL SOBRE TECNOLOGÍAS Y LÓGICA FUZZY ESTYLEF 2010 HUELVA

DEL 3 AL 5 DE FEBRERO · HOTEL BARCELÓ CONVENTION CENTER PUNTA UMBRÍA · HUELVA



ORGANIZAN

PATROCINAN



Actas del

XV Congreso Español sobre

Tecnologías y Lógica Fuzzy

Huelva, 3-5 de febrero de 2010

ORGANIZADORES:

Universidad de Huelva
European Society for Fuzzy Logic and Technology

EDITORES:

Antonio Peregrín Rubio
Jesús Alcalá Fernández
Francisco José Moreno Velo
Francisco Alfredo Márquez Hernández

ENTIDADES COLABORADORAS:

Ministerio de Ciencia e Innovación. Gobierno de España.
Consejería de Innovación, Ciencia y Empresa. Junta de Andalucía.
Vicerrectorado de Extensión Universitaria. Universidad de Huelva.
Consejo Social de la Universidad de Huelva.
Fundación Cajasol.

ISBN: 978-84-92944-02-6

Presentación

Este volumen *electrónico* recoge las comunicaciones presentadas a la XV Edición del Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF 2010) celebrado en Huelva entre los días 3 y 5 de Febrero de 2010. Huelva acoge a los investigadores españoles en el campo de las Tecnologías Fuzzy, cuyas aportaciones muestran una vez más, por su afluencia y calidad científica, la buena salud de la que goza la comunidad española en Lógica Fuzzy y Soft Computing.

La organización quiere agradecer a la European Society for Fuzzy Logic and Technology (EUSFLAT) el habernos confiado a Huelva la organización de esta edición que han acogido anteriormente diversas ciudades españolas. Lo hemos organizado con gran ilusión y compromiso para mantener tanto la habitual acogida y excelencia organizativa de todas las sedes anteriores del congreso como la habitual hospitalidad de las tierras onubenses.

En esta edición continuamos con la iniciativa puesta en marcha dos ediciones atrás de entregar una muestra de reconocimiento a los compañeros que cuentan con una dilatada carrera investigadora en el ámbito de la Lógica Fuzzy, así como también el del mejor trabajo presentado por un estudiante.

Deseamos agradecer a todos los que han contribuido a la elaboración de las actas, en primer lugar a los autores que han sometido sus trabajos y que son el actor principal del congreso, a los organizadores de las sesiones especiales ya que la buena prodigalidad participativa y calidad científica de los trabajos presentados anteriormente referidos se debe también a su activo trabajo y dedicación durante estos meses, y extensivamente a la organización del evento.

Finalmente queremos agradecer también a los patrocinadores, Ministerio de Ciencia e Innovación, Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía, Fundación Cajasol, Universidad de Huelva y su Consejo Social, por su compromiso con el mismo que nos ha permitido realizar todo aquello que planteamos inicialmente.

Esperamos que todos los asistentes al evento disfruten de unos días de encuentro científico con colegas y amigos, de la hospitalidad onubense, y disfruten de la lectura de los trabajos que se encuentran en las presentes actas que muestran el buen hacer de nuestra comunidad científica.

Antonio Peregrín Rubio

Presidente del Comité de Programa de ESTYLF2010

Presidente Honorífico

Enric Trillas

Comité Organizador

Presidente: Francisco Herrera

Jesús Alcalá

Javier Aroba

Antonio Ángel Márquez

Francisco Alfredo Márquez

Francisco José Moreno Velo

Nieves Pavón

Miguel Ángel Rodríguez

Ana M^a Roldán

Comité de Programa

Presidente: Antonio Peregrín (UHu)

Rafael Alcalá (UGr)

José M. Alonso (ECSC)

Claudi Alsina (UPC)

Senen Barro (USC)

Edurne Barrenechea (UPNa)

Angel Barriga (IMSE-CSIC)

Iluminada Baturone (IMSE-CSIC)

José Manuel Benítez (UGr)

Alberto Bugarín (USC)

Pedro Burillo (UPNa)

Humberto Bustince (UPNa)

José Manuel Cadenas (UM)

Tomas Calvo (UAH)

Pablo Carmona (UEx)

Jorge Casillas(UGr)

Juan Luis Castro (UGr)

José Jesús Castro-Schez (UCLM)

Elena Castiñeira (UPM)

Oscar Cerdón (ECSC)

Inés Couso (UO)

Susana Cubillo (UPM)

M^a Teresa De Pedro (IAI-CSIC)

M^a José Del Jesús (UJa)

Miguel Delgado (UGr)

Francesc Esteve (IIIA-CSIC)

Pere Garda (IIIA-CSIC)

José L. Garda-Lapresta (UVa)

Luis Garmendia(UCM)

M^a Ángeles Gil (UO)

Lluís Godó (IIIA-CSIC)

Daniel Gómez (UCM)

Antonio González (UGr)

Sergio Guadarrama (ECSC)

Julio Gutiérrez (UPM)

Francisco Herrera (UGr)

Enrique Herrera-Viedma (UGr)

Joan Jacas (UPC)

Luis Jiménez (UCLM)

M^a Teresa Lamata (UGr)

Vicente Liern (UV)

Enrique López (ULe)

Bonifacio Llamazares (UVa)

Luis Magdalena (ECSC)

Roque Marín (UM)

María José Martín (UGr)

Luis Martínez (UJa)

Francisco Mata (UJa)

Gaspar Mayor(UIB)

José M^a Merigó (UB)

José M. Molina (UC3M)

Javier Montero (UCM)

Susana Montes (UO)

Eduard Montseny (UPC)

Claudio Moraga (ECSC)

Francisco José Moreno (UHu)

Manuel Mucientes (USC)

Manuel Ojeda (UMa)

José A. Olivas (UCLM)

José Ignacio Peláez (UMa)

Raúl Pérez (UGr)

Héctor Pomares (UGr)

Ana Pradera (URJC)

Jordi Recasens (UPC)

Adolfo R. deSoto (ULe)

Ignacio Rojas (UGr)

Gregorio Sainz (UVa)

Santiago Sánchez (IMSE-CSIC)

Luciano Sánchez (UO)

Daniel Sánchez (UGr)

Miguel Angel Sicilia (UAH)

Pilar Sobrevilla (UPC)

Alejandro Sobrino (USC)

Vicenc Torra (IIIA-CSIC)

Joan Torrens (UIB)

Aida Valls (URV)

José Luis Verdegay (UGr)

Amparo Vila (UGr)

Pedro Villar (UGr)

ÍNDICE

Sesión Regular: Aplicaciones de Sistemas Difusos

- 1 .- Aplicación de lógica difusa en el control de contraste de imágenes 1-6
(Nashaat M.H., A. Barriga)
- 2 .- Towards people indoor localization combining WiFi and human motion recognition..... 7-12
(J. M. Alonso, A. Álvarez, G. Trivino, N. Hernández, F. Herranz, M. Ocana)
- 3 .- Calidad y seguridad en explotaciones mineras de carbón. Gestión de conformidades mediante Fuzzy-Logic 13-18
(L. Argüelles)
- 4 .- Extracción de Conocimiento en el Desarrollo de Software mediante el Análisis de los Registros de Tiempos de Trabajo 19-24
(A. Peralta, F. P. Romero, J. A. Olivas, M. Polo)

Sesión Regular: Aplicaciones de la Lógica Fuzzy

- 1 .- Obtención de un número difuso utilizable en aplicaciones a partir de datos ordenados 25-30
(J. Moreno-Garcia, L. Jiménez, L. Rodríguez, F. Andrés)
- 2 .- Sistemas de afinación borrosos: un estudio del temperamento igual de 12 notas..... 31-36
(T. León, V. Liern)
- 3 .- Una primera aproximación al problema de asignación de atraques con tiempos de llegada difusos..... 37-42
(B. Melián-Batista, J. M. Moreno-Vega, J. L. Verdegay)
- 4 .- Algoritmo genético con evaluación borrosa multicriterio: aplicación a la planificación de movimientos en manipuladores paralelos 43-48
(D. A. López García, F. Gómez-Bravo, J. M. Martín Ramos, I. López Quintero, G. Carbone)
- 5 .- Aprendizaje de la sintaxis para la descripción de escenas compuestas por figuras geométricas..... 49-54
(D. Perez, S. Guadarrama)
- 6 .- A prototype of a machine speaking with emotions 55-60
(A. van der Heide, G. Trivino)
- 7 .- Detección de idioma de sitios web mediante redes neuronales 61-66
(P. Garrido, F. J. Martinez, F.J. Vela, J. Tramullas, I. Plaza)
- 8 .- Sistema Fuzzy aplicado al CAGD basado en relaciones de similitud 67-72
(A. Monreal, J. Recasens, J. Jacas)

Sesión Regular: Sistemas Difusos. Estudios sobre Interpretabilidad

- 1 .- Índice de Interpretabilidad Semántica para el Ajuste de Sistemas Basados en Reglas Difusas y Selección de Reglas Mediante un Algoritmo Evolutivo Multi-Objetivo..... 73-78
(M. J. Gacto, R. Alcalá, F. Herrera)
- 2 .- Mejora del equilibrio precisión-interpretabilidad en modelos difusos 79-84
(M. Galende, G. I. Sainz)
- 3 .- Introducción de un mecanismo para la mejora de la interpretabilidad en sistemas difusos lingüísticos con defuzzificación adaptativa 85-90
(A. Márquez, F. A. Márquez, A. Peregrin)
- 4 .- Criterios para el empleo de transformaciones ortogonales en la mejora de modelos difusos. Aplicación a modelos basados en ART 91-96
(M. I. Rey, M. Galende, G. I. Sainz)

Sesión Regular: Sistemas Difusos. Extracción de Conocimiento

- 1 .- Resolución de Problemas Multi-clase con Sistemas de Clasificación Basados en Reglas Difusas Lingüísticos Basados en Aprendizaje por Parejas y Relaciones de Preferencia. 97-102
(A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, F. Herrera)
- 2 .- Una metodología para generar bases de conocimiento difusas basada en agrupamiento difuso, selección y ajuste evolutivos 103-108
(R. Toscano, J. Aroba, A. Peregrin)
- 3 .- EVFUZZYSYSTEM: Evolución de sistemas difusos para problemas de regresión multi-dimensionales 109-114
(M. Martínez-Ballesteros, V. M. Rivas)
- 4 .- Un algoritmo genético para el ajuste de Sistemas de Clasificación Basados en Reglas Difusas con Conjuntos Intervalo-Valorados Difusos 115-120
(J. A. Sanz, A. Fernández, H. Bustince, F. Herrera)
- 5 .- FUZZYCN2: Un algoritmo de extracción de listas de reglas difusas 121-126
(P. Martín, F. J. Moreno-Velo)

Sesión Regular: Teoría de Lógica Fuzzy

- 1 .- Reflexividad y T-transitividad aproximadas 127-132
(D. Boixader, J. Recasens)
- 2 .- On the relation between Fuzzy Preorders, Fuzzy Closing Morphological Operators and Fuzzy Consequence Operators 133-138
(J. Elorza, R. Fuentes-González, J. Bragard, P. Burillo)
- 3 .- An explicit expression for k-negations 139-144
(E. de Amo, J. Fernández-Sánchez)

4 .- First Order SMTL Logic and Quasi-witnessed Models	145-150
(M. Cerami, F. Esteva)	
5 .- Interpretación de las soluciones de $X \rightarrow R = X$ como bloques de tolerancia borrosa y su relación con el análisis de conceptos borrosos	151-156
(C. Alcalde, A. Burusco, R. Fuentes-González)	
6 .- On the Notion of Coherence in Fuzzy Answer Set Semantics	157-162
(N. Madrid, M. Ojeda-Aciego)	
7 .- Grado de parentesco entre predicados	163-168
(I. García-Honrado, E. Trillas, S. Guadarrama)	
8 .- From computer algebra to discretized continuous logic	169-174
(E. Roanes-Lozano, J. Montero, A. Hernando, L.M. Laita)	
9 .- On fuzzy homomorphisms between hyperrings	175-180
(I. P. Cabrera, P. Cordero, G. Gutiérrez, J. Martínez, M. Ojeda-Aciego)	
10 .- Estudio de la probabilidad de Zadeh para t-normas arquimedianas	181-186
(J. Hernández, I. Montes, D. Martinetti, S. Montes)	
11 .- On the logics of similarity-based approximate and strong entailments	187-192
(F. Esteva, L. Godo, R. O. Rodríguez, T. Vetterlein)	

Sesión Regular: Lógica Fuzzy en Sistemas de Información

1 .- Consultas con comparadores difusos en algoritmos de minería de datos en SQL Server 2008	193-198
(A. Urrutia, C. Gutiérrez-Soto, J. Méndez)	
2 .- Implementación de una base de datos difusa con FIRST-2 y PostgreSQL	199-204
(A. Urrutia, J. Galindo, A. Sepúlveda)	
3 .- A complete logic for fuzzy functional dependencies over t-norms	205-210
(P. Cordero, M. Enciso, A. Mora, I. Pérez de Guzmán)	
4 .- Inclusión de conjuntos borrosos en el núcleo del sistema bousi~prolog	211-216
(P. Julián-Iranzo, C. Rubio-Manzano, J. Gallardo-Casero)	
5 .- EMD: una metodología de recuperación para sistemas de lenguaje natural basados en casos	217-222
(A. Moreo, J.L. Castro, V. López, J.M. Zurita)	
6 .- Aplicación de Lógica Difusa en el Proceso de Expansión de Folksonomías	223-228
(C. Martínez-Cruz, I. J. Blanco, M.A. Vila)	

Sesión Especial: Soft Computing en minería y resumen de datos

Organizadores:

Daniel Sánchez

María José Martín Bautista

- 1 .- Speeding Up Evolutionary Learning Algorithms using GPUs 229-234
(A. Cano, A. Zafra, S. Ventura)
- 2 .- Determinando Automáticamente los Dominios de Competencia de un Sistema de Clasificación Basado en Reglas Difusas: Un Caso de Estudio con FH-GBML 235-240
(J. Luengo, F. Herrera)
- 3 .- Minería de reglas de asociación con programación genética gramatical 241-248
(J. M. Luna, J. L. Olmo, J. R. Romero, S. Ventura)
- 4 .- Un Modelo de Clasificación basado en Reglas de Asociación Difusas para Problemas de Alta Dimensionalidad y Selección Evolutiva de Reglas.....249-254
(J. Alcalá-Fdez, R. Alcalá, F. Herrera)
- 5 .- Linguistic summarization of data with probabilistic fuzzy quantifiers 255-260
(F. Díaz-Hermida, A. Bugarín)
- 6 .- Construcción de conjuntos de reglas difusas para la clasificación de objetos 261-266
(C.J. Solana-Ciprés, J. Albusac, J.J. Castro-Schez, J. Moreno-García, L. Rodríguez-Benítez)
- 7 .- Linguistic comparison of time series: a fuzzy approach 267-272
(R. Castillo-Ortega, N. Marín, D. Sánchez)

Sesión Especial: Funciones de agregación

Organizadores:

Tomasa Calvo Sánchez

Humberto Bustince Sola

Javier Montero de Juan

- 1 .- On the satisfaction of the functional equation $A(x, N(x))=c$ 273-278
(A. Pradera)
- 2 .- Una generalización de la migratividad 279-284
(H. Bustince, J. Fernández, B. De Baets, J. Montero, R. Mesiar)
- 3 .- Using heavy aggregations in a unified model between the weighted average and the OWA operator 285-290
(J. M. Merigó, M. Casanovas)
- 4 .- Decision making with distance measures, weighted averages and induced OWA operators 291-296
(J. M. Merigó, A. M. Gil-Lafuente)

5 .- Regular Multidistances	297-302
(J. Martín, G. Mayor)	
6 .- Sobre funciones y reglas de agregación	303-308
(J. Montero, D. Gómez, V. López, J. Tinguaro Rodríguez, B. Victoriano)	
7 .- Aggregation and arithmetic operations on fuzzy natural number-valued multisets...	309-314
(J. Casasnovas, J. V. Riera)	
8 .- Means on discrete product lattices.....	315-320
(G. Beliakov, H. Bustince, J. Fernández, D. Paternain)	
9 .- Specificity for interval-valued fuzzy sets using aggregation operators.....	321-324
(R. González del Campo, L. Garmendia, V. López)	

Sesión Especial: Soft Computing en visión artificial

Organizadores:

Humberto Bustince Sola

Jesús Chamorro Martínez

Miguel Pagola Barrio

1 .- Clustering con conjuntos intervalo-valorados difusos para la segmentación de imágenes de ultrasonidos.....	325-330
(A. Jurio, M. Pagola, E. Barrenechea, C. Lopez-Molina)	
2 .- Reducción de imagen utilizando operadores OWA de dimensión 2	331-336
(D. Paternain, H. Bustince, G. Beliakov, M. Galar)	
3 .- Coarseness Fuzzy Partitions applied to Texture Image Retrieval	337-342
(J. Chamorro-Martínez, P. Martínez-Jimenez, J.M. Soto-Hidalgo)	
4 .- Caracterización del color del citoplasma en imágenes de citología cervico-vaginal ..	343-348
(S. Román, P. Sobrevilla, E. Montseny, E. Lerma)	

Sesión Especial: Aplicaciones del Soft Computing

Organizadores:

Jorge Casillas Barranquero

Manuel Mucientes Molina

1 .- Fuzzy steering control of a hydraulic tractor and laser perception in obstacle avoidance strategies	349-356
(D. Martín, D.M. Guinea, M.C. García-Alegre, E. Villanueva, D. Guinea)	
2 .- Reglas Difusas para el Control de un Vehículo en la 2009 Simulated Car Racing Competition: un Enfoque Ganador	357-362
(E. Onieva, D. A. Pelá, J. Pérez, J. Alonso, V. Milanés)	
3 .- Sistema basado en reglas difusas para el mapeo de ontologías	363-368
(S. Fernández, J. R. Velasco, M.A. López-Carmona)	

- 4 .- Evaluating the impact of sewage sludge application on agricultural soils 369-374
(J. Pijuan, A. Valls, A. Pasuello, M. Schuhmacher)
- 5 .- Generación de emociones en agentes de recomendación usados en portales de comercio electrónico B2C.....375-380
(V. Herrera Tirado, J.J. Castro-Schez, J. Albusac, C. Gonzalez Morcillo)
- 6 .- Softcomputing para la elaboración de modelos predictivos de calidad superficial en fresado a alta velocidad 381-386
(V. M. Flores, M. Correa, J. R. Alique)
- 7 .- Aplicación de técnicas de soft computing para la aceleración de la transcodificación de vídeo.....387-392
(C. Solana-Ciprés, G. Fernández-Escribano, L. Rodríguez-Benítez, J. Moreno-García, L. Jiménez-Linares)

Sesión Especial: Software para Soft Computing

Organizador:

Francisco José Moreno Velo

- 1 .- Aplicación de Xfuzzy 3 al procesado de imágenes basado en reglas 393-398
(I. Baturone, P. Brox, R. Arjona)
- 2 .- GUAJE - A Java environment for generating understandable and accurate models ...399-404
(J. M. Alonso, L. Magdalena)
- 3 .- Software didáctico sobre control difuso 405-410
(E. León González, J. Galindo)
- 4 .- Nip1.5 : una herramienta software para la generación de conjuntos de datos con imperfección para minería de datos 411-416
(J. M. Cadenas, J.V. Carrillo, M. C. Garrido, E. Muñoz)
- 5 .- KEEL: una herramienta software para el análisis de sistemas difusos evolutivos 417-422
(J. Derrac, A. Fernández, J. Luengo, S. García, L. Sánchez, J. Alcalá, F. Herrera)
- 6 .- XFHL: una herramienta de inducción de sistemas difusos jerárquicos 423-428
(S. Cala Cordero, F. J. Moreno-Velo)

Sesión Especial: Redes neuronales difusas y modelos RBF

Organizadores:

Ignacio Rojas Ruiz

Héctor Pomares Cintas

Luis Javier Herrera Maldonado

- 1 .- Modelado y control neuro-difuso de procesos de taladrado de alto rendimiento 429-434
(A. Gajate, P. Vega, R. Haber)

- 2 .- Hibridación de paradigmas de clasificación en el contexto del reconocimiento automático de actividades físicas cotidianas..... 435-440
(O. Baños, H. Pomares, I. Rojas)
- 3 .- Mejoras en el diseño multiobjetivo de redes de funciones de base radial..... 441-446
(P. L. López, A. J. Rivera, C. J. Carmona, M. D. Pérez-Godoy)
- 4 .- Sistema neuro-difuso para el modelado de la evolución del color. Modelado de un proceso de blanqueamiento dental 447-452
(L. J. Herrera, H. Pomares, I. Rojas, J. Santana, R. Pulgar, R. Ghinea, M. M. Perez)

Sesión Especial: Soft Computing y Bioinformática

Organizadores:

Rocío C. Romero Zaliz

David Pelta

- 1 .- Tecnología difusa aplicada a la comparación de motivos de ADN. 453-458
(F. García Alcalde, C. Cano Gutiérrez; F. J. López Domingo, A. Blanco Morón)
- 2 .- Importancia de las Propiedades Físico-Químicas de los Aminoácidos en la Predicción de Estructuras de Proteínas usando Vecinos más Cercanos. 459-464
(G. Asencio Cortés, J. S. Aguilar-Ruiz)
- 3 .- Definición de Umbral Mínimo para la Predicción de Estructura Secundaria de Proteínas. 465-470
(A. E. Márquez Chamorro , J. S. Aguilar Ruiz, E. Anguiano Rey)
- 4 .- Cis-cop: Multiobjective identification of cis-regulatory modules based on constrains 471-476
(R. Romero-Zaliz, M. Martinez Ballesteros, I. Zwir, C. del Val)
- 5 .- Optimization of multi-clasiffiers using a fuzzy logic approach: an aplication to the gene prediction problem. 477-482
(R. Romero-Zaliz, C. del Val, I. Zwir)

Sesión Especial: Extracción de conocimiento a partir de datos de baja calidad

Organizadores:

Luciano Sánchez Ramos

Inés Couso Blanco

- 1 .- A coevolutionary genetic algorithm able to learn a FRBS and detect the less informative instances in low quality problems..... 483-488
(L. Sánchez)
- 2 .- Extracción de conocimiento a partir de reglas difusas ponderadas que soportan datos imprecisos 489-494
(A. M. Palacios)
- 3 .- Cómo hacer inferencia estadística a partir de datos de baja calidad 495-500
(I. Couso)

- 4 .- Uso de Marcadores Difusos para Solucionar el Problema de la Coplanaridad en la Calibración de la Cámara en 3D. Aplicación en Identificación Forense por Superposición Craneofacial 501-506
(O. Ibañez, O. Cordon, S. Damas, J. Santamaría)
- 5 .- Nuevos diseños experimentales para algoritmos de extracción de conocimiento con datos de baja calidad.....507-512
(J. Otero)
- 6 .- Statistical preference as a comparison method of two imprecise fitness values.....513-518
(I. Montes, D. Martinetti, S. Díaz, S. Montes)

Sesión Especial: Soft Computing y recuperación de información en Internet

Organizadores:

José Ángel Olivas Varela
 María José Martín Bautista
 Alejandro Sobrino
 Enrique Herrera Viedma

- 1 .- Representación de relaciones causales a través de un proceso semi-automático de recuperación 519-524
(C. Puente, A. Sobrino, J. A. Olivas)
- 2 .- FOG: Arquitectura flexible para la generación automática de ontologías 525-530
(E. Fdz-Viñas, J. Serrano-Guerrero, J. A. Olivas, J. de la Mata, A. Soto)
- 3 .- Representación Múltiple de Consultas: Una alternativa a la Expansión de Consultas en Sistemas de Recuperación de Información 531-536
(J. Ruiz-Morilla, J. Serrano-Guerrero, J. A. Olivas, E.F. Viñas)
- 4 .- Ontologías borrosas para representar perfiles de usuario en una herramienta de gestión de objetos de aprendizaje 537-542
(M. Ferreira-Satler, V. H. Menéndez, F. P. Romero, A. Zapata, M. E. Prieto)
- 5 .- Una aproximación declarativa a la clasificación de documentos 543-548
(F. P. Romero, P. Julián Iranzo, M. Ferreira-Satler, J. Gallardo)
- 6 .- Trabajando con relaciones de preferencia lingüísticas difusas incompletas para recomendar recursos de investigación en una Biblioteca Digital Universitaria.....549-554
(C. Porcel, E. Herrera-Viedma)
- 7 .- Ordered weighted averaging approaches for aggregating gradual trust and distrust..555-560
(C. Cornelis, P. Victor, E. Herrera-Viedma)

Sesión Especial: Toma de decisiones con información difusa y/o lingüística

Organizadores:

José Luís García Lapresta

Luís Martínez López

- 1 .- A distance function to assess the similarity of words using ontologies 561-566
(M. Batet, A. Valls, K. Gibert)
- 2 .- Probabilistic decision making with the owa operator and the 2-tuple linguistic approach 567-574
(J. M. Merigó, M. Casanovas, M. Espinilla, L. Martínez)
- 3 .- A mobile linguistic decision support system to help users in their e-commerce activities 575-580
(I.J. Pérez, S. Alonso, F.J. Cabrerizo, E. Herrera-Viedma)
- 4 .- Licitación de proyectos públicos con valoraciones lingüísticas 581-586
(E. Falcó Díaz de Cerio, J. L. García Lapresta)
- 5 .- Decisiones mayoritarias con umbral de apoyo. Análisis de la coherencia colectiva... 587-592
(J. L. García Lapresta, B. Llamazares, P. Pérez Asurmendi)
- 6 .- Toma de decisión lingüística. Modelos computacionales simbólicos y su aplicación en el tratamiento de percepciones 593-600
(R.M. Rodríguez, L. Martínez, F. Herrera)
- 7 .- Uso de jerarquías lingüísticas extendidas para mejorar la representación de información lingüística no balanceada 601-606
(M. Espinilla, L. Martínez)
- 8 .- A Multigranular Hierarchical linguistic model for pharmacy and therapeutics committees 607-612
(J. M. Doña, A. J. García, J. I. Peláez)
- 9 .- Fuzzy decision making with the probability, the weighted average and the OWA operator 613-618
(J. M. Merigó)

Sesión Especial: Optimización y evaluación para la toma de decisiones fuzzy

Organizadores:

Jose Manuel Cadenas Figueredo

Maria del Carmen Garrido Carrera

Enrique Muñoz Ballester

- 1 .- Nueva aproximación al método TÓPSIS difuso con etiquetas lingüísticas 619-624
(M. S. Garcia-Cascales, M. T. Lamata)

2 .- Optimization of route distribution problems with fuzzy objectives.....	625-630
(J. Brito, F. J. Martinez, J. A. Moreno, J.L. Verdegay)	
3 .- Algoritmos para abordar modelos de inversión afectados de incertidumbre	631-636
(C. Calvo, J. M. Cadenas, J. V. Carrillo, M. C. Garrido, C. Ivorra, V. Liern)	
4 .- Toma de decisiones en ensambles basados en árbol fuzzy.....	637-642
(P. Bonissone, J. M. Cadenas, M. C. Garrido, R.A. Díaz-Valladares, A. Muñoz)	
5 .- Mejoras en el uso de aprendizaje activo con árboles fuzzy: un ejemplo de su aplicación en la toma de decisiones de un sistema coordinado	643-648
(G. Acampora, G. Fenza, E. Muñoz, B. Romera)	

APLICACIÓN DE LÓGICA DIFUSA EN EL CONTROL DE CONTRASTE DE IMÁGENES

Nashaat M. Hussein

Angel Barriga

Instituto de Microelectrónica de Sevilla (CNM-CSIC)/ Universidad de Sevilla,
{nashaat,barriga}@imse.cnm.es

Resumen

En esta comunicación se aplican los operadores del álgebra de Lukasiewicz para modificar el contraste de imágenes. El control del contraste se realiza mediante mecanismos de inferencia basados en lógica difusa. De esta manera el sistema de control de contraste que se obtiene presenta características que hacen que sea muy adecuado para su implementación hardware ya que da lugar a circuitos de bajo coste y alta velocidad de procesado.

Palabras Clave: control de contraste, lógica difusa.

1 INTRODUCCIÓN

El concepto de contraste en una imagen fue desarrollado con el propósito de describir los cambios de la energía lumínica. La luminancia describe la energía del estímulo más bien que cambios de la energía, así que no es bastante por sí mismo. Existen muchas propuestas de medida del contraste. Básicamente el contraste puede definirse como el cambio de la luminancia relativa de los elementos de una imagen. Por lo tanto corresponde a la diferencia de luminancia que existe entre dos puntos de una imagen. El histograma de la imagen es una herramienta útil para examinar el contraste en la imagen [1].

Nuestro interés en esta comunicación se centra en describir un mecanismo del control del contraste. Esta técnica se basa en la aplicación de los operadores del álgebra de Lukasiewicz (suma-acotada y producto-acotado) y mecanismos de inferencia basados en lógica difusa con objetivo de realizar una simplificación del diseño de los circuitos que controlen el contraste en las imágenes (circuitos con bajo coste y una alta velocidad).

Esta comunicación se organiza en cuatro apartados. En el apartado siguiente se muestran las técnicas del control del

contraste. A continuación se describe el efecto de los operadores del álgebra de Lukasiewicz en el contraste de una imagen. Finalmente se considera la aplicación de sistemas de inferencia difusos en el control del contraste de imágenes.

2 TÉCNICAS DE CONTROL DE CONTRASTE

Una definición de contraste es el contraste de Weber que es comúnmente utilizada en el contexto de la iluminación. Consiste en la diferencia entre dos luminancias dividido por la luminancia menor.

$$C = \frac{L_{\max} - L_{\min}}{L_{\min}}$$

Otra definición de contraste es la medida de la varianza. Viene dada por la siguiente expresión:

$$\sigma^2 = \frac{1}{MN} \sum_{k=1}^L (k - \bar{k})^2 n_k$$

donde M y N son el tamaño de la imagen, k es el valor de la luminancia en el rango $[1, L]$, n_k es la frecuencia del nivel de luminancia k y $\bar{k}$ es el valor medio de la distribución de luminancia,

$$\bar{k} = \frac{1}{MN} \sum_{k=1}^L k n_k$$

Cuando todos los píxeles tienen el mismo nivel de luminancia su varianza es cero, y cuando la diferencia entre todos los posibles pares de píxeles es mayor la varianza es mayor.

Por otro lado los valores ($p_k = n_k/MN$; $k=1, 2, \dots, L$) constituyen una distribución de probabilidad sobre el conjunto de los valores de luminancia de manera que $\sum_{k=1}^L p_k = 1$. Es posible utilizar la entropía como una medida de contraste [2]:

$$H = - \sum_{k=1}^L p_k \ln p_k$$

Cuando la distribución de tonos de luminancia es uniforme ($p_k = 1/L$), la entropía alcanza su valor máximo (que es $\ln(L)$) y que corresponde a una imagen con el máximo contraste. Esto sugiere que una medida estándar en el intervalo $[0,1]$ del contraste de una imagen es $H/\ln(L)$.

Conviene tener en cuenta que la entropía es una medida de incertidumbre. Cuando vale cero corresponde al mínimo contraste y para una imagen con distribución uniforme, lo que corresponde al máximo de contraste, la incertidumbre o falta de información es máxima.

Debido al proceso de digitalización de imágenes los píxeles están codificados por un número de bits limitado. Por ejemplo en el caso de imágenes monocromas de 8 bits por píxel supone distinguir 256 niveles de gris. Si el rango de variación en el brillo de la imagen es mucho menor que el rango dinámico de la cámara entonces el verdadero rango de valores va a ser mucho más pequeño que toda la gama de 0 a 255. Es decir la imagen obtenida en la salida de los sensores de la cámara no cubriría toda la gama. Por lo tanto en muchas situaciones la imagen tiene un menor rango de valores de brillo. Estos valores se pueden encontrar en la gama media (los valores intermedios de grises) o en los extremos de la gama (brillantes u oscuros).

La visibilidad de los elementos que forman una imagen se puede mejorar realizando un estiramiento del contraste con el fin de reasignar los valores de los píxeles para cubrir toda la gama disponible. Esto significa que los píxeles son interpolados entre los valores extremos de la gama dinámica.

Un mecanismo habitual de mejora del contraste es realizando una interpolación lineal [1], [3], [4], [5]. Esta técnica de expansión lineal del contraste permite aumentar la discriminación visual y es útil cuando las variaciones de luminancia en la imagen permiten distinguir entre los elementos que lo componen.

Existen implementaciones hardware para realizar el control de contraste. Así, en [4] se describe un circuito implementado en una tecnología CMOS de $0.25\mu\text{m}$. El método descrito en [3] se aplica en imágenes de vídeo y se basa en aproximación de funciones lineales a tramos de densidad acumulativa (CDF, *Cumulative Density Function*).

Otras técnicas se basan en las transformaciones locales de los píxeles y se denominan operaciones de punto. Las operaciones de punto o funciones punto a punto requieren en cada paso conocer el valor de la intensidad de un solo

píxel sobre el que se aplica la transformación deseada. Después del procesado ya no se requiere el píxel por lo que este tipo de operación se denomina de memoria cero.

Las operaciones de punto se realizan de manera más eficiente con tablas de búsqueda (LUT). La LUT es un vector simple que utiliza el valor del píxel actual como índice del vector. El nuevo valor es el elemento del vector almacenado en esa posición. La nueva imagen se construye repitiendo el proceso para cada píxel. El uso de LUT evita repetitivos e innecesarios cálculos. Cuando se trabaja con imágenes de, por ejemplo, 8 bits sólo es necesario calcular 256 valores. En este caso el tamaño de la imagen es irrelevante, puesto que el valor de cada píxel de la imagen es un número entre 0 y 255 y el resultado de la tabla produce otro número entre 0 y 255. Estos algoritmos se pueden aplicar sin necesidad de utilizar ninguna memoria intermedia puesto que la imagen de salida puede almacenarse en el mismo espacio de memoria que la imagen de entrada.

Una de las transformaciones no lineales más utilizada es la transformación Gaussiana que viene dada por:

$$g(i,j) = \frac{\phi\left(\frac{f(i,j)-0.5}{\sigma\sqrt{2}}\right) + \left[\frac{0.5}{\sigma\sqrt{2}}\right]}{\phi\left(\frac{0.5}{\sigma\sqrt{2}}\right)}$$

donde los corchetes en la expresión $[x]$ representa función truncación de x y además:

$$\phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy$$

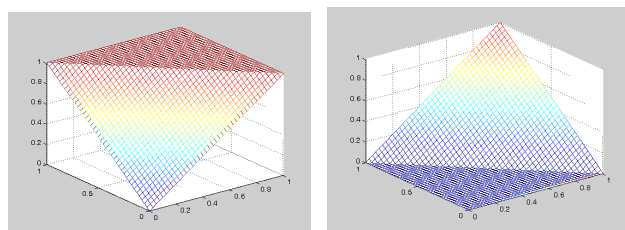
Esta transformación aumenta el contraste de la imagen haciendo que las partes oscuras sean más oscuras y las partes brillantes sean más claras.

3 EFECTO DE LOS OPERADORES DE ŁUKASIEWICZ EN EL CONTRASTE

El desarrollo de los conceptos teóricos de las lógicas multivaluadas se inició en la década de los 20s por Jan Łukasiewicz quien estableció la generalización de la lógica clásica a la lógica multivaluada. Más tarde, a finales de los años 50, C.C. Chang formalizó el álgebra multivaluada sobre la base de la lógica de Łukasiewicz. La definición de los operadores básicos es (figura 1):

suma acotada: $x \oplus y = \min(1, x + y)$

producto acotado: $x \otimes y = \max(0, x + y - 1)$



a) b)
Figura 1: Superficie de a) suma acotada, b) producto acotado.

La aplicación de los operadores de Łukasiewicz en una imagen da lugar a una transformación de la distribución de los niveles de los píxeles. Esta transformación produce un cambio de los niveles bajos a valores altos o de niveles bajos a valores altos, es decir con la aplicación de los operadores de Łukasiewicz la mayoría de los niveles de luminancia de la imagen son objeto de un desplazamiento en el histograma.

El operador suma acotada actúa como un filtro paso bajo y realiza un desplazamiento de los píxeles a niveles altos. De este modo se obtiene una imagen más clara. La figura 2 muestra el efecto de la aplicación de la suma acotada a píxeles consecutivos de la imagen original. Es posible observar el desplazamiento de los píxeles hacia el blanco. El control de contraste usando la suma acotada se puede hacer mediante la introducción de un parámetro adicional que permite regular el desplazamiento de la frecuencia:

$$x \oplus y \oplus C$$

donde C es el parámetro de control del contraste. La gama de valores que puede tomar C (codificado con 8 bits) se encuentra en el intervalo $[-128, 127]$. La figura 2 muestra el efecto de la suma acotada con diferentes valores de los parámetros sometidos a control ($C=0$ y $C=30$).

La operación complementaria a la suma acotada corresponde al producto acotado. Este operador da lugar a un desplazamiento del histograma hacia el negro. Este efecto se observa en la figura 3 que muestra el resultado de la aplicación del producto acotado y de su histograma. El control del contraste aplicando el producto acotado se realiza por medio del parámetro C en la siguiente expresión:

$$x \otimes y \otimes C$$

La figura 3 muestra la aplicación del producto acotado con diferentes valores del parámetro de control C .

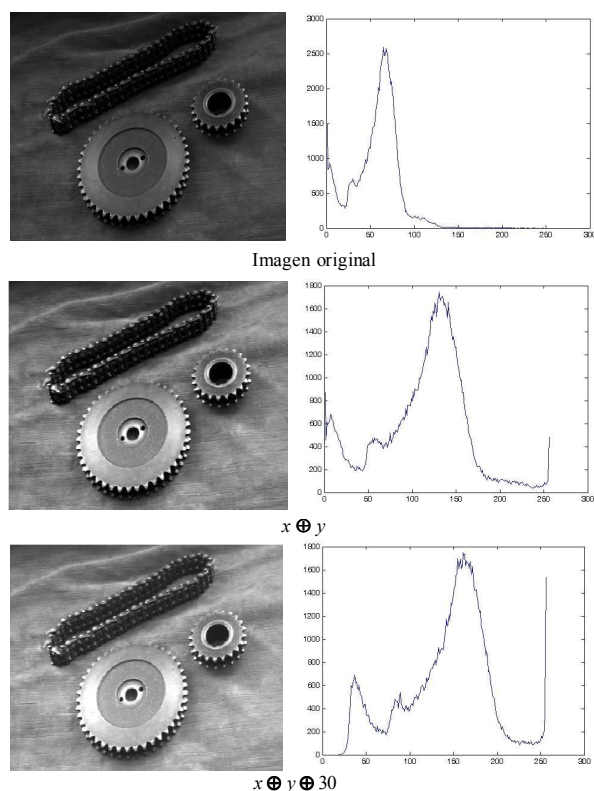


Figura 2: Control del contraste mediante la suma acotada junto con el histograma de las imágenes

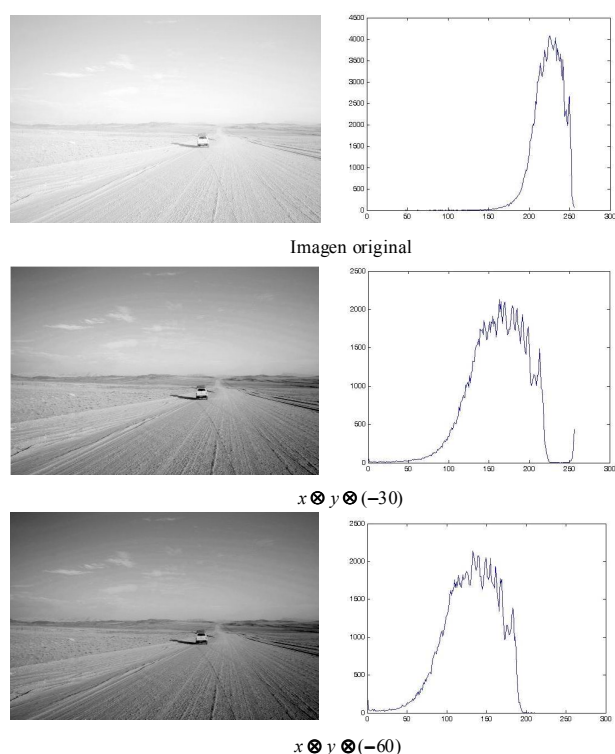


Figura 3: Control del contraste mediante el producto acotado junto con el histograma de las imágenes

4 CONTROL BASADO EN LÓGICA DIFUSA

La técnica de control del contraste que se ha presentado se basa en realizar una transformación del histograma de la imagen aplicando los operadores suma acotada y producto acotado. Estos operadores dan lugar a un desplazamiento y expansión de los valores del histograma. El control de este efecto se realiza mediante un parámetro C que permite regular la intensidad de la transformación. La variación del contraste en una imagen no tiene porque ser uniforme. Así pueden existir regiones donde el contraste sea menor que en otras zonas de la imagen. Por ello el parámetro C debería adaptarse a cada región de la imagen con objeto de mejorar la calidad de la transformación. Así la expresión que regula el contraste mediante la suma acotada viene dada por la siguiente expresión:

$$x \oplus y \oplus f(x, y)$$

donde x e y son píxeles de la imagen y el parámetro de control es la función $f(x, y)$.

La función de control de contraste $f(x, y)$ depende de las características de cada imagen y permite adaptar la operación de control de contraste de manera local. En nuestro caso se ha optado por aplicar una heurística que determina dicha función aplicando un criterio de decisión mediante un mecanismo de inferencia basado en lógica difusa. Así el sistema de toma de decisiones se basa en criterios de proximidad, esto es, si los valores de los píxeles están muy cercanos (poco contraste) la función $f(x, y)$ debe ser alta mientras que si los píxeles están alejados la función debe ser baja.

La figura 4 muestra las especificaciones del sistema difuso para el control de contraste. Las funciones de pertenencia corresponden a cinco funciones triangulares equiespaciadas y con grado de solapamiento de dos. La salida del sistema está compuesta por 9 funciones de pertenencia de tipo *singleton*. La base de regla detalla la heurística descrita anteriormente, es decir,

- Si x es Bajo e y es Bajo entonces $f(x, y)$ es Muy Bajo (F1)
- Si x es Bajo e y es Medio entonces $f(x, y)$ es Bajo (F2)
- Si x es Bajo e y es Alto entonces $f(x, y)$ es Medio (F3)
- ...

La figura 5 muestra la superficie correspondiente a la función de control de contraste. La figura 6 muestra un ejemplo de aplicación del control de contraste. El caso de la figura 6b corresponde a la suma acotada, la figura 6c corresponde a la función de control basada en el sistema difuso. Puede observarse en la figura 6 que en la zona de la imagen correspondiente a la columna se puede apreciar los efectos del control del contraste. Se observa que cuando no se establece control los valores de la columna

se saturan (toman el valor blanco) por lo que se pierde contraste. Sin embargo cuando se aplica un control local (caso c) se mejora el contraste en la zona de la columna.

En el caso de la aplicación del producto acotado la expresión que regula el contraste viene dada por la siguiente expresión:

$$x \otimes y \otimes f(x, y)$$

De la misma manera que en el caso de la suma acotada el cálculo de la función de control de contraste se basa en un motor de inferencia difuso basado en la base de conocimientos mostrada en la figura 7.

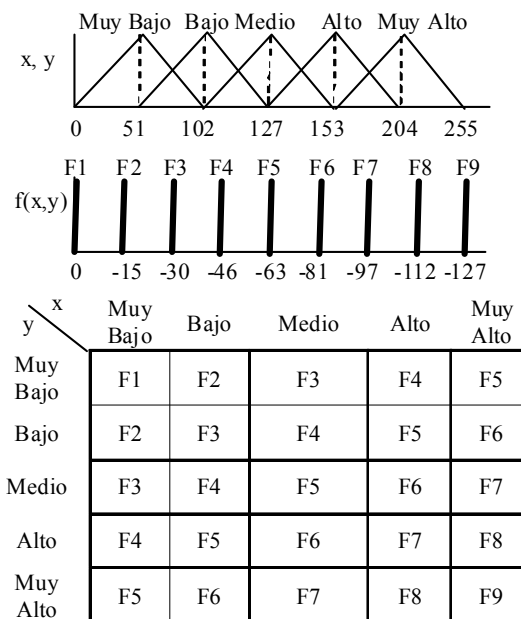


Figura 4: Sistema para la función de control de contraste $f(x, y)$ asociada a la suma acotada

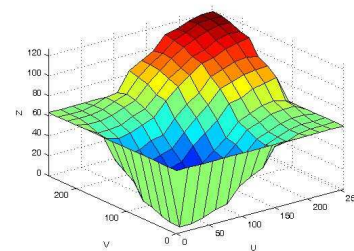


Figura 5: Superficie correspondiente a la función de control de contraste.

Los resultados que se obtienen de la aplicación del producto acotado se muestran en la figura 8. En el caso de la figura 8b se muestran los resultados corresponden al producto acotado sin adaptación mientras que la figura 8c corresponde al control usando lógica difusa.

En la técnica que proponemos para el control del contraste se aplica cada uno de los dos operadores (suma acotada y producto acotado) dependiendo de las características de la imagen. Así la suma acotada se utiliza en el caso de las imágenes oscuras mientras que el producto acotado debe ser aplicado en imágenes claras. Sin embargo, en general, las imágenes pueden tener zonas con diferentes características. Es decir pueden coexistir zonas oscuras y zonas claras en la misma imagen. Por ello conviene adaptar el mecanismo de control de contraste a las características locales de la imagen. Para ello se ha considerado un sistema de toma de decisión que determine el tipo de operador que debe aplicarse en cada momento (suma acotada en la zona oscura de la imagen y el producto acotado en la zona clara).

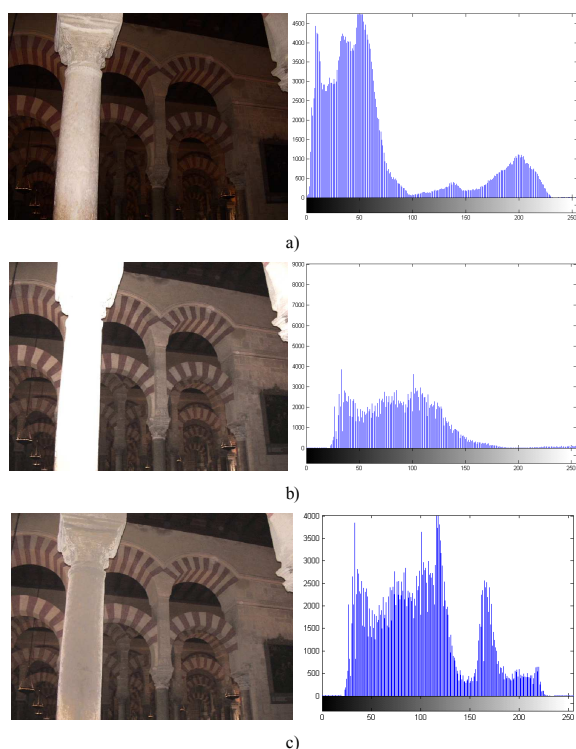
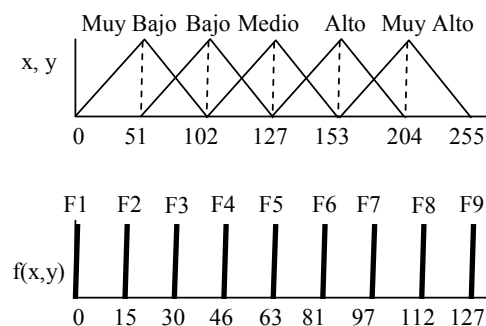


Figura 6: a) Imagen original, b) $x \oplus y$,
c) $x \oplus y \oplus f(x, y)$.

El sistema de toma de decisiones se basa en un mecanismo de inferencia basado en lógica difusa. Las especificaciones del sistema difuso se muestra en la figura 9. Las funciones de pertenencia del consecuente son tres funciones de tipo *singleton* ($Z1$, $Z2$ y $Z3$). La etiqueta $Z1$ indica realizar la suma acotada mientras que $Z3$ supone aplicar el producto acotado. El caso de que la salida sea $Z2$ significa que no se realiza ningún cambio de contraste y por lo tanto la salida corresponde al valor de la entrada. La base de reglas contiene 9 reglas. Cuando el contraste es bajo se aplica la suma acotada o bien el producto acotado mientras que si el contraste es alto la salida no cambia respecto a la entrada.



$x \backslash y$	Muy Bajo	Bajo	Medio	Alto	Muy Alto
Muy Bajo	F9	F8	F7	F6	F5
Bajo	F8	F7	F6	F5	F4
Medio	F7	F6	F5	F4	F3
Alto	F6	F5	F4	F3	F2
Muy Alto	F5	F4	F3	F2	F1

Figura 7: Sistema para la función de control de contraste $f(x, y)$ asociada al producto acotado

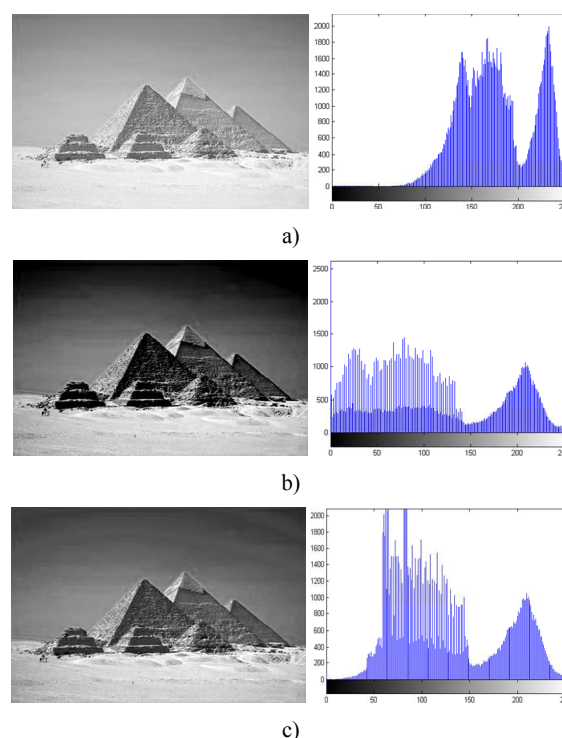


Figura 8: a) Imagen original, b) $x \otimes y$,
c) $x \otimes y \otimes f(x, y)$.

De acuerdo con nuestra estrategia de control de contraste el sistema que proponemos se basa en aplicar una máscara que recorre la imagen. En función del contraste local el sistema decide aplicar el operador más adecuado. Esta toma de decisión se realiza con el sistema difuso discutido anteriormente cuya base de conocimiento se muestra en la figura 9. De acuerdo con esta estrategia el sistema global se compone de 3 motores de inferencia difusos como se ilustra en la figura 10. Los sistemas FIM1 y FIM2 generan las funciones de control de contraste asociadas a la suma acotada y el producto acotado respectivamente. El sistema FIM3 corresponde al sistema de toma de decisión que selecciona el mejor operador. Finalmente es posible añadir un parámetro adicional C que permita al usuario realizar un control específico. De esta manera la funcionalidad del sistema viene dada por la siguiente expresión:

$$x' = \begin{cases} x \oplus y \oplus f(x, y) \oplus C & \text{if } z = Z1 \\ x & \text{if } z = Z2 \\ x \otimes y \otimes f(x, y) \otimes C & \text{if } z = Z3 \end{cases}$$

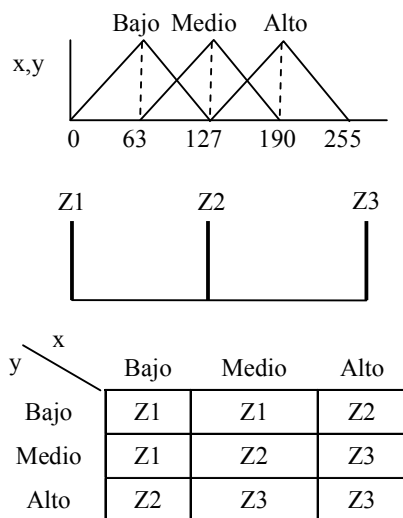


Figura 9: Sistema de toma de decisión.

Agradecimientos

Este trabajo ha sido financiado en parte por los proyectos TEC2008-04920/MICINN del Ministerio de Educación y Ciencia, P08-TIC-03674 de la Junta de Andalucía.

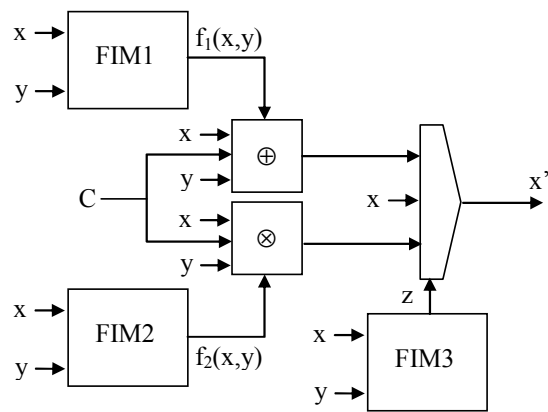


Figura 10: Esquema del sistema de control de contraste

Referencias

- [1] Z.Y. Chen, B.R. Abidi, D.L. Page, M.A. Abidi, "Gray-Level Grouping (GLG): An Automatic Method for Optimized Image Contrast Enhancement—Part I: The Basic Method", *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2290-2302, Aug. 2006.
- [2] Amar Khellaf, Azeddine Beghdadi, and Henri Dupoisot, "Entropic Contrast Enhancement", *IEEE Transactions On Medical Imaging*, vol. 10, no. 4, pp. 589-592, Dec. 1991.
- [3] S.Y. Kim, D. Han, S.J. Choi and J.S. Park, "Image Contrast Enhancement Based on the Piecewise-Linear Approximation of CDF", *IEEE Transactions on Consumer Electronics*, Vol. 45, No. 3, pp. 828-834, Aug. 1999.
- [4] H.C Kim, B.H. Kwon, M.R. Choi "An Image Interpolator with Image Improvement for LCD Controller", *IEEE Transactions on Consumer Electronics*, vol. 47, no. 2, pp. 263-271, May 2001.
- [5] H.H. Cho, C.H. Choi, B.H. Kwon and M.R. Choi: "A Design of Contrast Controller for Image Improvement of Multi-Gray Scale Image", *IEEE Asia Pacific Conference on ASICs*, pp. 131-133, Aug 2000.
- [6] N.M. Hussein, A. Barriga: "Image Contrast Control based on Łukasiewicz's Operators", *IEEE International Symposium on Intelligent Signal Processing (WISP'09)*, Sept. 2009.
- [7] N.M. Hussein, A. Barriga: "Image Contrast Control based on Łukasiewicz's Operators and Fuzzy Logic", *International Conference on Intelligent Systems Design and Applications (ISDA'09)*, Dec. 2009.

TOWARDS PEOPLE INDOOR LOCALIZATION COMBINING WIFI AND HUMAN MOTION RECOGNITION

José M. Alonso¹ Alberto Alvarez¹ Gracian Trivino¹
N. Hernández² F. Herranz² Manuel Ocaña²

¹ European Centre for Soft Computing (ECSC), Mieres (Asturias)
{jose.alonso,alberto.alvarez,gracian.trivino}@softcomputing.es

² Department of Electronics, University of Alcalá (UAH), Madrid
{noelia.hernandez,fernando.herranz,mocana}@depeca.uah.es

Abstract

This work presents a general framework for people indoor localization. Firstly, a WiFi localization system implemented as a fuzzy rule-based classifier (FRBC) is used to deal with the intrinsic uncertainty of such environments. It consists of a set of linguistic variables and rules automatically generated from experimental data. As a result, it yields an approximate position at the level of discrete zones (room, corridor, toilet, etc). Secondly, a Fuzzy Finite State Machine (FFSM) mainly based on expert knowledge is used for human motion (activity, body posture and step length) recognition. The goal is finding out whether people is (or not) moving, in which direction, at which pace, etc. Finally, another FFSM combines both WiFi localization and human motion recognition with the aim of obtaining a robust, reliable, and easily understandable human-oriented localization system.

Keywords: WiFi localization, Human motion recognition, Fuzzy rule-based classifier, Fuzzy finite state machine.

1 INTRODUCTION

People localization systems provide interesting applications in many areas [12, 14, 16]: personal navigation assistance, medical assistance, finding and rescuing emergency first responders, personal security, etc. We are mainly interested in security applications (for instance sending warnings when someone gets into a dangerous area in order to reduce the occupational health and safety risk) and/or people assistance (for instance helping elderly or handicapped people).

There are three main types of localization systems: (1) satellite based systems, (2) local network based systems and (3) sensor based systems.

First, the satellite based systems, e.g., Global Positioning System (GPS) [9], are widely used in outdoor applications with great success. However, they do not provide precise indoor localization (nor even in cities with high buildings), making this problem an open challenge.

Second, local network based systems use the network infrastructure to estimate user's location. There are systems based on pre-existing networks like ZigBee networks designed for home control applications [7]. However, the most used systems are based on WiFi networks. The main advantage opposite to satellite based systems is that they are able to provide indoor absolute localization. In contrast, the principal drawback is the need of a complete network infrastructure in the whole building where we want to localize a person. Luckily, this technology is quickly growing of coverage. Currently, there are WiFi Access Points (APs) in most public buildings like hospitals, libraries, universities, museums, etc. In addition, measuring the WiFi signal level is free even for private WiFi networks. As a result, WiFi technology is a good choice for indoor global localization systems yielding a good accuracy-cost trade-off [2].

Third, sensor based systems provide absolute information (e.g., magnetic compass, ultrasonic or infrared sensors) or relative information (e.g., inertial measurement units or pressure sensors). One low-cost inertial sensor is the accelerometer, based on the Micro Electro Mechanical Systems (MEMS) technology that has allowed its integration in small and low energy consumption devices. Accelerometers can be used as step length estimators; furthermore they let us to obtain some information about body posture [21]. Human activity can also be analyzed by means of combining one accelerometer with a skin conductivity meter [22].

Our localization system is mainly based on Fuzzy Logic (FL) [25] because it combines several heterogeneous sources of knowledge (mainly expert knowledge and knowledge automatically extracted from experimental data provided by sensors), dealing with vague information, and its interaction with humans demands building an easily understandable system. There are other recent works [4, 8] that show the advantages of using FL for modeling and monitoring human activity. They are mainly based on fusing video sensors what means installing additional hardware (HW) like video cameras in the environment under study. Our approach takes profit of the available HW (WiFi APs) and avoids adding new devices to the environment.

This paper contributes to the field of people indoor localization by providing a fusion framework that yields the estimated location of a person thinking on maximizing its accuracy and interpretability. We propose a general framework for people indoor localization which is made up of three main modules as illustrated in Fig. 1. Each block will be described in the following sections. Section 2 focuses on WiFi localization and how it contributes to this work. Section 3 presents some previous works related to human motion recognition. Section 4 describes the proposed fusion framework (human-oriented indoor localization), explaining how it can be applied in a real-world scenario. Finally, section 5 draws some conclusions and future works.

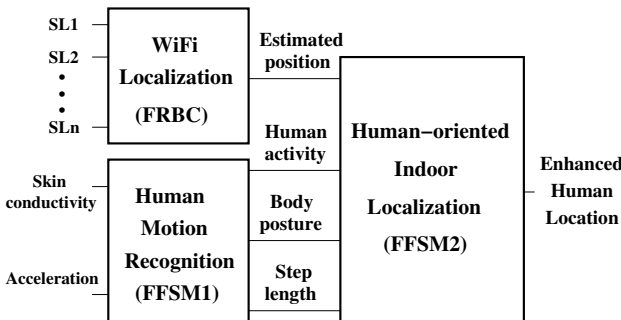


Figure 1: Scheme of the proposed fusion framework.

2 WIFI LOCALIZATION

WiFi localization systems use 802.11b/g network infrastructure to estimate a device position without using additional hardware. The received signal level (SL) from each AP depends on the distance and the obstacles between APs and the receiver. Therefore, the simplest method for estimating the device position consists of applying a triangulation algorithm [18]. Unfortunately, in indoor environments SL is strongly affected by the well-known multipath effect [17] that comprises reflection, refraction and diffraction. Thus,

SL becomes a complex function of the distance that dynamically changes with time because it is affected by every modification made in the environment.

Only approximate solutions are able to get nice results. Authors of [5] propose the use of a priori radio map storing the received SL of each AP belonging to an interest region. The radio map is built during the training stage. Then, in the estimation stage, a vector with received SL of each AP is created and compared with the radio map to obtain the estimated position. We have previously proposed the use of fuzzy classification for WiFi robot localization, handling the signal measure uncertainty and getting small localization errors [2]. In this contribution we propose to use an enhanced version of such system devoted to people localization. It yields zone-level localization. Notice that, a zone may be only one room or a set of rooms.

A fuzzy rule-based classifier (FRBC) is designed following the Highly Interpretable Linguistic Knowledge (HILK) methodology [1]. Useful pieces of knowledge are automatically extracted from experimental data and represented by means of linguistic variables [26] and linguistic rules [15] under the fuzzy logic formalism. Rules are of form: **IF** *Condition* **THEN** *Conclusion*, where both *Condition* and *Conclusion* use linguistic terms.

For instance, **IF** *Signal received from AP_i is High* AND *Signal received from AP_j is Low* **THEN** *We are close to Position P which belongs to zone Z*. Notice that, the output of the FRBC will be one zone of the environment along with an activation degree which is understood as a degree of confidence on the system output. It is computed as the result of a fuzzy inference that takes into account all defined variables and rules. We have selected the usual fuzzy classification structure and the winner rule fuzzy reasoning mechanism. Of course, several fuzzy rules can be fired (by the same input vector) at the same time yielding several output classes (zones) with non-zero activation degree.

As an illustrative example, let us suppose that two zones A and B are one close to the other (with a common wall) and one person is inside zone A but near the wall, the FRBC may give as output an activation degree of 0.7 related to zone A and 0.3 regarding zone B. Hence, the interpolation ability of fuzzy systems makes possible to define a hierarchical localization system where the position can be refined as much as desired. In a first level it is possible to identify the floor of the building, in a second level it points out the room where the person is located, but in a third level (depending on the application) it may be interesting giving also the position inside the room. Thus, thanks to this approach, a FRBC made up of a small

number of rules is used for each level, keeping a good accuracy-interpretability trade-off.

3 HUMAN MOTION RECOGNITION

The use of accelerometers in personal navigation or localization is not new. Accelerometers can be used as the basic sensors of a technique called Dead Reckoning (DR). It is the process of estimating one's current position based upon a previously determined position, and advancing that position based upon known or estimated speeds over elapsed time. Pedestrian Dead Reckoning (PDR) consists of estimating the position of a person based on his/her step length and heading [6].

In theory, the walking speed and distance could be obtained by integrating the acceleration signals once and twice respectively. However, the double integration causes an error in position which grows cubically in time [11]. For this reason, some authors [10, 23] use the accelerometer, in dead reckoning algorithms, to provide an estimation of the step length whereas the heading is obtained from gyroscopes or magnetic compass. The easiest way to get an estimation of the step length is to count the number of steps provided by the vertical axis acceleration. Then, a fix step length can be assigned to every person. The main drawback of this method is that the step length is not constant for a certain person; it varies significantly with the walking speed. There are some models that relate the step length to the vertical acceleration, like [24] that uses an empirical equation to provide step length estimation:

$$\text{Step Length} = \sqrt[4]{a_{max} - a_{min}} \cdot C \quad (1)$$

where:

- a_{min} and a_{max} are the minimum and the maximum of the vertical acceleration.
- C is an experimental parameter practically invariant for a certain person. It is calculated making a person to walk a known distance.

In previous works, we have shown how a Fuzzy Finite State Machine (FFSM) can deal with signals which evolve following an approximately repetitive pattern [3]. It is able to synchronize with the acceleration signal produced during the human gait and to extract the relevant characteristics suitable for our purpose. It can provide the maximum and minimum values of the vertical acceleration in each complete gait cycle, and using Equation 1, we are able to calculate the step

length in each gait cycle and therefore, the distance walked.

In our approach, not only have we used accelerometers as step length estimators, but also we have successfully used a FFSM for processing data provided by accelerometers yielding useful information about human activity [22] as well as body posture [21]. The FFSM in charge of human motion recognition (called FFSM1 in Fig. 1) was thoroughly described in [3]. The interested reader is referred to it to get further details. As an introduction to what a FFSM is and how it is used, we define a FFSM as a tuple:

$$\{Q, U, f, Y, g, Q_0\}$$

where:

- Q is a set of fuzzy states $\{q_1, \dots, q_i, \dots, q_{N_{states}}\}$. When the system evolves, it could stay simultaneously in several fuzzy states. The degree with which the system is in state q_i is called degree of activation of the state q_i .
- U is the input vector $(u_1, u_2, \dots, u_{N_{inputs}})$.
- f is the state activation transition function defined by $Q[t+1] = f(U[t], Q[t])$, which is implemented using a set of Takagi-Sugeno-Kang (TSK) fuzzy rules [20]. We distinguish between rules (R_{ii}) to remain in a state q_i and rules (R_{ij}) to change from the state q_i to the state q_j . The allowed transitions have associated fuzzy rules and simply the forbidden ones have not associated fuzzy rules.
- Y is the output vector $(y_1, y_2, \dots, y_{N_{outputs}})$. The output provided by a FFSM can be expressed as a linguistic summary of relevant characteristics of the system while it remains in q_i .
- g is the output function $Y[t] = g(U[t], Q[t])$. When a transition occurs, the values of the output variables are typically obtained applying, e.g., the average and the standard deviation of the values of the input variables while the signal remains in the considered state.
- Q_0 is the initial degree of activation of the set of states.

4 FUSION FRAMEWORK: HUMAN-ORIENTED INDOOR LOCALIZATION

The general framework illustrated on Fig. 1 suggests combining WiFi localization data and human motion

data, including the distance, the human activity and the body posture. On the one hand, FRBC is in charge of estimating the user position taking as inputs the SL received from each AP in the current position. Then, FFSM1 translates data captured by one accelerometer and one skin conductivity meter into useful information summarizing the user motion such as human activity, body posture, and distance walked. Finally, FFSM2 combines all available information to obtain a robust, reliable, and easily understandable localization system. The final output is called *Enhanced human location* because it is two-fold human-oriented. First, it yields the estimated location of a person. Second, the whole system design is made thinking on maximizing its interpretability. Thus, the output explanation is based on a set of linguistic variables and rules quite close to the human natural language.

The rest of the section gives some details on how FFSM2 was designed for a specific real-world scenario. It summarizes the preliminary experiments made as part of this work, with the aim of showing how it can be applied to the chosen scenario. The test-bed was established at the European Centre for Soft Computing (ECSC) premises whose layout is shown in Fig. 2. The topological map of the whole environment is illustrated at the top picture of the figure. It has a surface of 49 x 9 meters, and it has been discretized into seven main zones as it can be appreciated at the middle picture: (1) main corridor, (2) coffee area, (3) working area A, (4) working offices, (5) working area B, (6) administration area, and (7) head unit offices.

These seven zones determine the set of fuzzy states of the FFSM2 introduced in Fig. 1. The designed diagram of states for FFSM2, with the allowed transitions rules, is shown at the bottom part of the Fig. 2. The whole design includes defining the input vector, the state activation transition function (fuzzy rules), the output vector through its output function and the initial degree of activation of the set of states. The input vector has four components related to the following variables (that take values in the interval $[0,1]$):

- The WiFi estimated position (WEP). It is computed as the output of the FRBC yielding a degree of membership for each of the seven zones.
- The degree of activity (A). As explained in [22], it comprises four different states (*Idle*, *Working*, *Walking*, and *Relaxing*) with their corresponding degree of activation.
- The body posture (P). As the degree of activity, it has four different states (*Seated*, *Standing up*, *Upright*, and *Sitting down*) [21] also with their degree of activation.

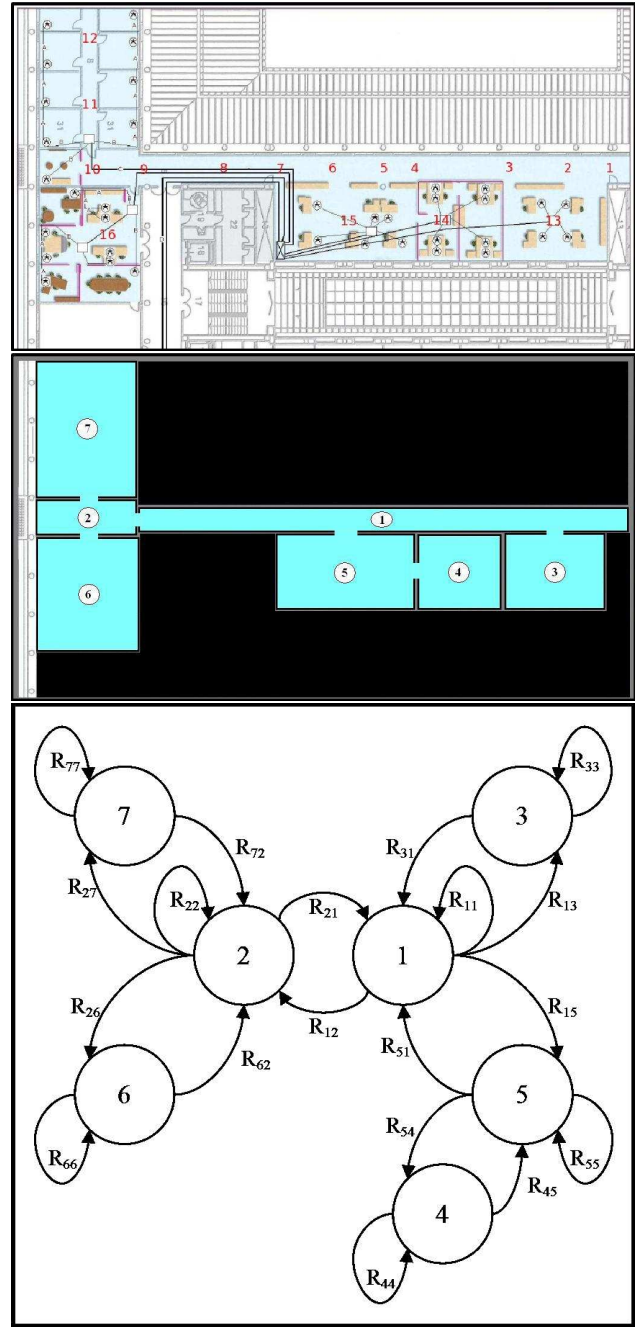


Figure 2: Discretization and diagram of states of the ECSC environment.

- The distance (D). It is calculated from the step length, and it is modeled as a linguistic variable that comprises several linguistic terms (*None*, *Short*, *Medium*, *Long*, etc.) with their attached membership functions.

The state activation function, as can be deduced from the diagram shown in Fig. 2, is made up of 19 fuzzy rules. As an example, below you can find four of them

to have an idea about how they look like:

R_{11} : IF ($Q[t]$ is *Zone 1*) AND (WEP is *Zone 1*)
AND (A is *Walking*) AND (P is *Upright*)
AND (D is *Short*) THEN $Q[t + 1]$ is *Zone 1*.

R_{12} : IF ($Q[t]$ is *Zone 1*) AND (WEP is *Zone 2*)
AND (A is *Relaxing*) AND (P is *Sittingdown*)
AND (D is *Long*) THEN $Q[t + 1]$ is *Zone 2*.

R_{51} : IF ($Q[t]$ is *Zone 5*) AND (WEP is *Zone 1*)
AND (A is *Walking*) AND (P is *Standing up*)
AND (D is *Medium*) THEN $Q[t + 1]$ is *Zone 1*.

R_{33} : IF ($Q[t]$ is *Zone 3*) AND (WEP is *Zone 3*)
AND (A is *Working*) AND (P is *Seated*)
AND (D is *None*) THEN $Q[t + 1]$ is *Zone 3*.

Previous rules are interpreted as follows. R_{11} means that the user was (and still continues) walking through the main corridor (*Zone 1*). R_{12} is going to be fired if the user goes from the main corridor to the coffee area (*Zone 2*). R_{51} means that the user is currently walking with the aim of leaving the working area A (*Zone 5*) to pass through the main corridor. Finally, according to R_{33} the user remains working at his/her usual place (*Zone 3*). Notice that, several rules can be fired at the same time. As a result, the final output depends on the activation degree of each rule as well as on the aggregation of all the rules.

The output vector of the FFSM provides the zone where the user is. In this way, it validates (confirming and/or correcting) the WEP. One criterion could be choosing that zone corresponding with the state with the maximum degree of activation. However, this selection will make the FFSM very sensitive to noise and spurious in the signal, and that is precisely what we want to avoid. Therefore, the output is designed as the zone corresponding to the state which has had the maximum average degree of activation over the last minute. Notice that, the initial degree of activation of the set of states Q_0 is taken as the computed WEP for each zone when the application was started.

For testing our WiFi localization system in the ECSC environment, we have built a training data set (LRN) with the SL measures (from all visible APs) carried out in the center of all the 16 remarked positions at the top picture in Fig. 2. One of the main problems of WiFi localization is the SL small-scale variations that happen when moving in a small distance (under the wavelength λ) [19]. They are generated by multipath effect producing a lot of uncertainty because they may arise up to 10 dBm for positions around the same location. For the 802.11b networks working at the 2.4 GHz range, λ is 12.5 cm. Thus, with the aim of checking the robustness of our system we have generated a test data set (TST) taking measures around all the sixteen

positions (always under λ distance).

Then, HILK methodology [1] has been applied on LRN in order to build a FRBC with a good accuracy-interpretability trade-off. It includes six input variables (one per each AP in the environment) which are characterized by strong fuzzy partitions of nine linguistic terms (extremely low, very low, low, etc). In addition, linguistic rules are automatically generated from data by means of the neuro-fuzzy ID3[13], a version of the popular fuzzy decision trees (FDT). Finally, the simplification procedure provided by HILK was run getting a more compact and general FRBC, keeping high accuracy while increasing even more its interpretability. Main results are presented in Table 1 considering two cases: (1) training-test with raw data and (2) training-test with averaged data in blocks of four samples. Notice that we use 4 samples per block because the maximum acquisition frequency of the WiFi interface is 4Hz, i.e., we are capturing 4 samples per second.

Table 1: Experimental results for WiFi localization.

	ROW DATA (1 sample)			AVERAGED DATA (4 samples)		
	LRN	TST	RULES	LRN	TST	RULES
MP	98.9	97.04	-	100	97.64	-
C45	99.74	96.59	28	100	91.61	8
FRBC	99.3	94.2	46	100	96.3	14

In addition to our solution (FRBC), for comparison purpose, the table includes results provided by other methods implemented in Weka¹: MP (Multilayer Perceptron) which yields very accurate neural network classifiers (disregarding comprehensibility), and C45 (Quinlan's decision trees) which usually provides good interpretability-accuracy trade-offs. The most accurate solution, i.e., the highest classification rate for training and test, is provided by MP with averaged data. C45 yields the most interpretable solution with the smallest number of rules. However, FRBC achieves the best trade-off with a larger accuracy than C45 but at the cost of a larger number of rules.

5 CONCLUSIONS

So far, we have already developed separately two of the three main blocks belonging to the proposed framework: (1) the WiFi localization module and (2) the human activity recognition module. In both cases, as explained in this paper but also in cited works, achieved results are encouraging. In consequence, the integration of both modules by means of the third one, which is currently under development, should yield a really powerful global system. Of course, more experiments

¹<http://www.cs.waikato.ac.nz/ml/weka/>

are needed in order to show the utility of our fusion framework because it is part of an ongoing research project.

Acknowledgements

This work has been funded by the Foundation for the Advancement of Soft Computing (Mieres, Asturias, Spain), by the Spanish government (CICYT) under project TIN2008-06890-C02-01, by the grant CCG08-UAH/DPI-3919 (SISLOPEWI) from CAM/UAH, and by S-0505/DPI/000176 (Robocity2030 Project) from the Science Department of Community of Madrid.

References

- [1] J. M. Alonso, L. Magdalena, and S. Guillaume. HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems*, 23(7):761–794, 2008.
- [2] J. M. Alonso, M. Ocaña, M. A. Sotelo, L. M. Bergasa, and L. Magdalena. WiFi localization system using fuzzy rule-based classification. *Computer Aided Systems Theory - LNCS 5717 - EUROCAST 2009*, pages 383–390, 2009.
- [3] A. Alvarez and G. Trivino. Comprehensible model of a quasi-periodic signal. In *Proc. of the 9th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2009.
- [4] D. Anderson et al. Modeling human activity from voxel person using fuzzy logic. 17(1):39–49, 2009.
- [5] P. Bahl and V.N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. of the IEEE Infocom*, pages 775–784, 2000.
- [6] S. Beauregard and H. Haas. Pedestrian dead reckoning: A basis for personal positioning. In *3rd Workshop on Positioning, Navigation and Communication*, pages 27–36, 2006.
- [7] R. Blasco et al. Indoor localization based on neural networks for non-dedicated zigbee networks in AAL. *5517 Lecture Notes in Computer Science*, pages 1113–1120, 2009.
- [8] C. S. Chan and H. Liu. Fuzzy qualitative human motion analysis. *IEEE Transactions on Fuzzy Systems*, 17(4):851–862, 2009.
- [9] P. Enge and P. Misra. Special issue on GPS: The global positioning system. In *Proc. of the IEEE*, volume 87, pages 3–172, 1999.
- [10] L. Fang et al. Design of a wireless assisted pedestrian dead reckoning system - the navmote experience. *IEEE Transactions on Instrumentation and Measurement*, 54(6):2342–2358, 2005.
- [11] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46, 2005.
- [12] B. Huang and N. Liu. Mobile navigation guide for the visually disabled. *Transportation Research Record*, pages 28–34, 2004.
- [13] H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi. Neuro-fuzzy ID3: A method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning. *Fuzzy Sets and Systems*, 81:157–167, 1996.
- [14] E. López, R. Barea, L.M. Bergasa, and M.S. Escudero. A human-robot cooperative learning system for easy installation of assistant robots in new working environments. *Journal of Intelligent and Robotic Systems*, 40(3):233–265, July 2004.
- [15] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic systems. *IEEE Transactions on Computers*, 26(12):1182–1191, 1977.
- [16] A. Marco et al. Location-based services for elderly and disabled people. *Computer communications*, 31(6):1055–1066, 2008.
- [17] V. Matellán, J. M. Cañas, and O. Serrano. WiFi localization methods for autonomous robots. *Robotica*, 24(4):455–461, 2006.
- [18] O. Serrano, J.M. Cañas, V. Matellán, and L. Roderio. Robot localization using wifi signal without intensity map. In *Proc. of the V Workshop Agentes Físicos (WAF'04)*, pages 79–88, 2004.
- [19] M. A. Sotelo et al. Low level controller for a POMDP based on WiFi observations. *Robotics and Autonomous Systems*, 55(2):132–145, 2007.
- [20] M. Sugeno. *Industrial Applications of Fuzzy Control*. Elsevier Science Inc., New York, NY, USA, 1985.
- [21] G. Trivino and G. Bailador. Linguistic description of human body posture using fuzzy logic and several levels of abstraction. In *IEEE Int. Conference on Computational Intelligence for Measurement Systems and Applications (CIMSIA)*, pages 105–109, 2007.
- [22] G. Trivino and A. van der Heide. Linguistic summarization of the human activity using skin conductivity and accelerometers. In *Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU)*, pages 1583–1589, 2008.
- [23] H. Wang et al. WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors. In *4th Workshop on Positioning, Navigation and Communication (WPNC)*, pages 1–7, April 2007.
- [24] H. Weinberg. Using the ADXL202 in pedometer and personal navigation applications. Technical report, Analog Devices, 2002.
- [25] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [26] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Parts I, II, and III. Information Sciences*, 8, 8, 9:199–249, 301–357, 43–80, 1975.

Calidad y seguridad en explotaciones mineras de carbón. Gestión de conformidades mediante Fuzzy-Logic

Luis Argüelles Méndez¹

Carbonar S.A., Departamento I+D
Cardenal Cienfuegos, 8., 33007 - Oviedo
¹ arguelles@carbonar.es

Abstract

Los sistemas de calidad en la industria se encuentran tradicionalmente ligados a la satisfacción del cliente. Sin embargo, en minería del carbón la calidad es obtenible atendiendo a los procesos de producción. En este contexto, la utilización de técnicas Fuzzy-Logic permite modelizar adecuadamente las conformidades y no-conformidades de los subprocesos mineros.

Keywords: Quality management, TQM, EFQM, calidad en minería, fuzzy modelling.

1 Introducción

La dualidad existente entre calidad y seguridad en una explotación subterránea de carbón se explica cuando se entiende la calidad no como un algo abstracto sino como un sistema de organización interno enfocado a procesos. Teniendo en cuenta que una explotación minera, al igual que toda instalación industrial, está constituida por un conjunto relacionado de procesos, se puede ver que si identificamos calidad con la idea de “hacer bien las cosas” se obtendrá directamente un aumento en la seguridad de los mismos y por ende, en la seguridad global de la explotación minera.

Este enfoque a los procesos difiere del tradicional enfoque de los sistemas de calidad dirigidos hacia el cliente [1]. De esta forma, una empresa cuya actividad sea la explotación de carbón que implemente un sistema de calidad no va a conseguir una mayor satisfacción del cliente final, ya que independientemente de las mejoras físicas, técnicas y organizativas que se introduzcan, la calidad del mineral viene dada en un muy alto grado por el comportamiento aleatorio de la propia geología del yacimiento. Especial atención en un sistema de calidad es la gestión de conformidades y no-conformidades de un proceso dado, donde la frontera entre estos dos estados es borrosa. Este trabajo presenta una metodología matemático-computacional basada en Fuzzy-Logic para

resolver aquellas situaciones en las cuales la frontera conformidad / no-conformidad de un subproceso no es nítida.

2 Formalismo Matemático

Bajo el enfoque que se presenta, todo proceso está formado por un conjunto finito de subprocesos de forma que un subproceso se conceptúa como “proveedor” del subproceso siguiente, el cual se convierte en “cliente” del anterior. Visto bajo esta óptica, es aplicable la idea de la satisfacción del “cliente” cuando existe calidad en la realización de un determinado subproceso[2]. Veamos esto con un ejemplo.

Podemos describir el proceso P de arranque de carbón mediante un conjunto de subprocesos S1, S2, S3, de forma tal que:

- S1 = Roza completa en el frente de explotación
- S2 = Desplazamiento del pancer del tajo
- S3 = Avance de los escudos

S1 es proveedor de S2 en el sentido de que el proceso de roza del frente de explotación debe de efectuarse siguiendo una serie de procedimientos establecidos que permitan al cliente S2 desarrollar a continuación su operación de la forma adecuada. Si todos estos procedimientos (velocidad correcta de la rozadora, regado continuo del carbón, etc.) se desarrollan de la forma correcta, S1 está transfiriendo calidad a su cliente S2 (calidad transferida). A partir de entonces, S2 se convierte en proveedor de S3 y S3 en cliente de S2.

Es imprescindible en un sistema de calidad una buena gestión de las conformidades y no-conformidades relacionadas con un proceso o subproceso. En líneas generales, un proceso que transfiere calidad a otro genera una situación de conformidad, mientras que una merma en la calidad transferida genera una situación de no-conformidad que ha de ser recogida en un informe (report). En función de la magnitud de la disminución de la calidad transferida del proveedor de un subproceso a su cliente hablaremos de no-conformidad leve o grave.

Como elemento de medida de la calidad de un subproceso suelen emplearse lo que en términos de calidad se

denominan “indicadores”, los cuales suelen ser descriptores numéricos de un subproceso o proceso. Un ejemplo de indicador es el avance en metros/día del sistema de rozadora y entibación autodesplazable o el porcentaje de tiempo diario que el sistema de cintas transportadoras se encuentran en situación de carga. Por otra parte, se denominan “estándares” a los niveles admitidos por una organización para los indicadores que describen su sistema de calidad. Una empresa puede adoptar un estándar de 1.5 metros de avance diario para su sistema de rozadora y entibación autodesplazable, un porcentaje del 60% para la situación en carga del sistema de transporte, etc.

Una vez expuestos estos conceptos iniciales, pasamos a describir un modelo propio de calidad que será la base de los desarrollos matemático-computacionales de nuestro interés.

Todo proceso industrial P puede enunciarse como una sucesión de subprocesos S_i de forma que:

$$P = \{S_1, S_2, \dots, S_n\}$$

En lo tocante a calidad, el concepto de proveedor/cliente se aplica de forma tal que:

$$S_{i-1} \rightarrow S_i \rightarrow S_{i+1}$$

Donde S_{i-1} es un subproceso proveedor de S_i y este a su vez lo es de S_{i+1} . De igual forma, S_i es cliente del subproceso S_{i-1} y S_{i+1} es cliente de S_i .

Todo subproceso S_i recibe una cierta calidad mensurable de S_{i-1} , desarrolla una serie de fenómenos o transformaciones internas y transfiere, como resultado, otra calidad mensurable a S_{i+1} . Estas calidades pueden expresarse numéricamente mediante números complejos:

$$S = a + bi$$

En este modelo, a , la parte real, representa la calidad recibida del proveedor, y b , la parte imaginaria, representa la calidad transferida al cliente. A partir de aquí podemos expresar estas calidades como números reales definidos en el rango $[0,1]$ de forma que un valor 0 significa una ausencia total de calidad y un valor 1 significa una calidad óptima, es decir:

$$\begin{aligned} 0 &\leq a \leq 1 \\ 0 &\leq b \leq 1 \end{aligned}$$

En la representación gráfica en el plano complejo, el eje imaginario Y se convierte en la calidad transferida al cliente o siguiente subproceso, mientras que el eje real X se convierte en la calidad recibida del proveedor o subproceso previo, tal y como se aprecia en la figura #1:

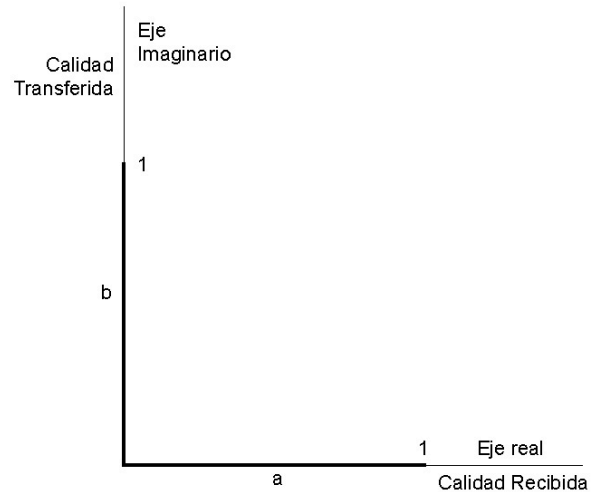


Figura #1: Calidades en el plano complejo

Como es lógico, todas las posibles representaciones de los números complejos definidos de esta forma se circunscriben al primer cuadrante del plano complejo, las cuales pueden dividirse en cuatro casos posibles, en función de los valores a y b , tal y como se analiza a continuación.

Caso #1: $a = b = 1$

Esta situación representa un caso donde tanto la calidad recibida como la calidad transferida es máxima, es decir, no existe pérdida de calidad en el subproceso. Representa la situación óptima, con un valor $\theta = 45^\circ$ y un módulo $|S_i| = \sqrt{2}$

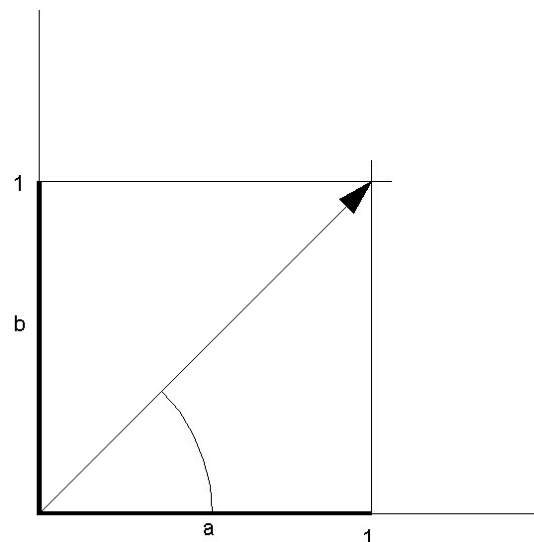


Figura #2: Representación primer caso

Caso #2: $a < 1; b = 1; \theta > 45^\circ$

En este caso, la calidad recibida es menor que uno, mientras que la calidad transferida es máxima. Esto significa que el subproceso S_i está subsanando los

problemas de calidad de S_{i-1} . En esta situación es probable que el subproceso S_i esté requiriendo más tiempo del habitual. Esta situación se muestra gráficamente en la figura #3:

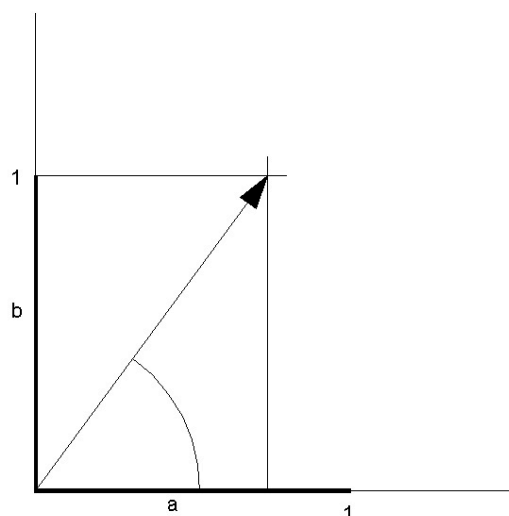


Figura #3: Representación segundo caso

Caso #3: $a = 1$; $b < 1$; $\theta < 45^\circ$

En este caso la calidad recibida es máxima (el proveedor ha realizado perfectamente su subproceso), pero el proceso actual S_i ha experimentado algún tipo de anomalía, con lo cual no puede transferir una calidad igual a uno al siguiente subproceso, S_{i+1} . Esta situación se muestra gráficamente en la figura #4:

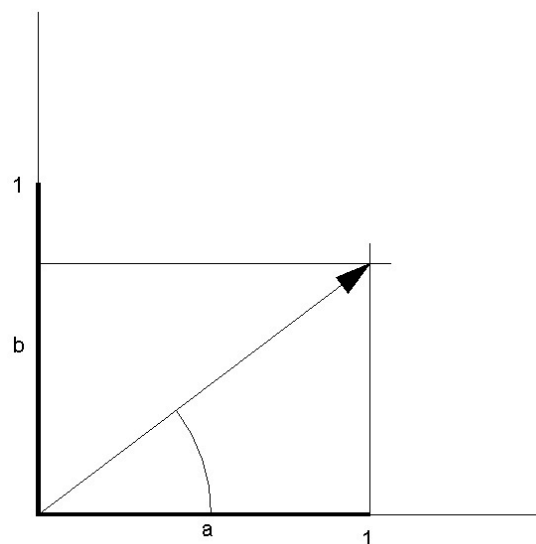


Figura #4: Representación tercer caso

Caso #4: $a < 1$; $b < 1$; $0^\circ < \theta < 90^\circ$; $|S_i| < \sqrt{2}$

Este caso representa la situación más desfavorable. El subproceso S_i tiene problemas con la calidad recibida de su proveedor y no es capaz de subsanar el problema generado anteriormente. En función de sus operaciones

específicas mayorará o minorará la calidad, pero no podrá transferir una calidad óptima. Esta situación se muestra gráficamente en la figura #5:

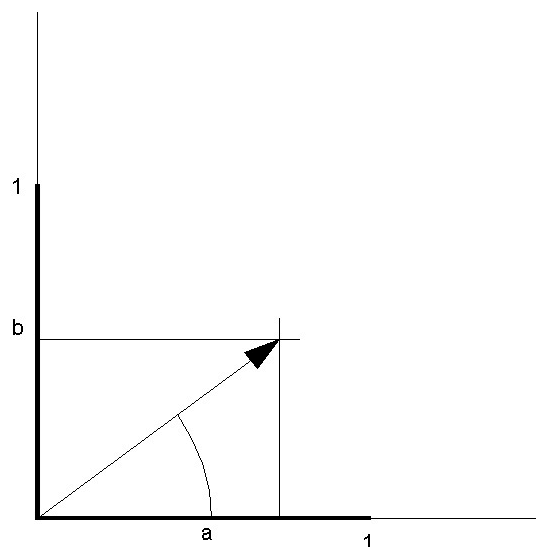


Figura #5: Representación cuarto caso

Corolarios: Del análisis gráfico mostrado en los anteriores párrafos se desprende:

- Si $\forall S_i, |S_i| = \sqrt{2}$, se da la condición suficiente para que todos los estándares de calidad de un proceso se satisfagan.
- Si $\forall S_{i,j}, |S_{i,j}| = \sqrt{2}$, entonces se alcanza la calidad total, ya que todos los subprocesos de todos los procesos de la mina reciben y transfieren una calidad $a=b=1.0$

De todo lo expuesto hasta ahora, y en lo tocante a calidad, una explotación minera puede representarse por una matriz rectangular M de la siguiente forma:

$$M = \begin{pmatrix} S_{1,1} & S_{1,2} & \dots & S_{1,n} \\ S_{2,1} & S_{2,2} & \dots & S_{2,n} \\ \dots & \dots & \dots & \dots \\ S_{m,1} & S_{m,2} & \dots & S_{m,n} \end{pmatrix}$$

Como es obvio, los diferentes procesos (filas de la matriz) suelen tener un diferente número de subprocesos. Por completitud en la exposición matemática podemos crear los subprocesos virtuales necesarios con valores $a = b = -1$, con lo que los corolarios anteriores siguen siendo validos, ya que el valor del módulo de estos subprocesos virtuales es siempre $\sqrt{2}$

No-conformidades: Al principio de esta sección decíamos que todo subproceso S_i recibe y transfiere una cierta calidad mensurable. La cantidad de calidad puede determinarse mediante varias vías, en función principalmente de la naturaleza de cada subproceso:

- Mediante expresiones matemáticas cerradas
- Mediante valores determinados por la experiencia de la gerencia o los directores de procesos (percepciones expertas)
- Mediante sistemas basados en Fuzzy-Logic

Especialmente interesante para este trabajo es la definición de calidad mediante técnicas Fuzzy-Logic. A partir de un conjunto de indicadores x_i, y_i se puede construir un conjunto f_m, f_n de reglas borrosas donde:

$$\begin{aligned} f_m(x_i) &\rightarrow a \\ f_n(y_i) &\rightarrow b \end{aligned}$$

Esto está íntimamente ligado a los conceptos de conformidad y no-conformidad. De hecho, tanto a como b (calidad recibida y transferida, respectivamente) pueden expresarse como variables lingüísticas formadas por la variable lingüística:

$$V = (\text{"conforme"}, \text{"no-conformidad leve"}, \text{"no conformidad grave"})$$

Las cuales pueden representarse gráficamente en el plano complejo de la forma expuesta en la figura #6:

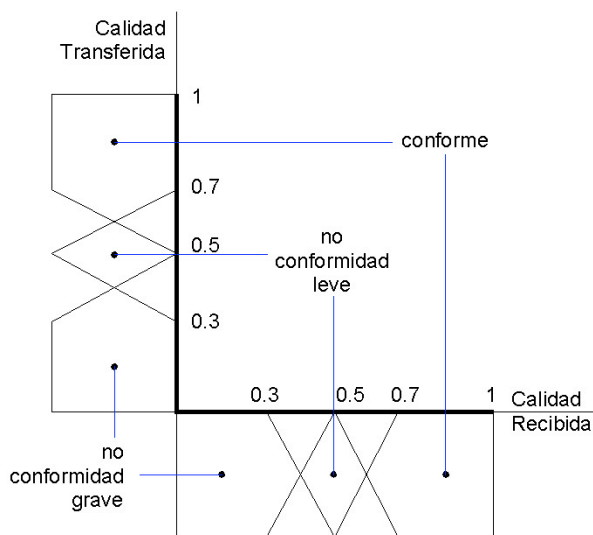


Figura #6: Variables lingüísticas de conformidades

Es decir, tanto el eje real como el imaginario se convierten en el universo de discurso de la variable lingüística enunciada.

3 A modo de aplicación

Todo equipo de una instalación minera tiene unas variables de estado que constituyen los indicadores de dicha instalación. Analizaremos en este ejemplo el funcionamiento de una rozadora actuando en el frente de explotación de una capa de carbón, tal y como se aprecia en la Figura #7.



Figura #7: Rozadora actuando contra capa de carbón

Por motivos de espacio, supongamos que la calidad recibida es $a = 1$ (asumimos que el carbón de la capa se encuentra en perfectas condiciones). Se puede determinar un modelo borroso donde la calidad transferida b, expresada por la variable lingüística V de conformidades venga dada en función de dos variables lingüísticas; velocidad angular de la rozadora y estado de la rozadora:

$$\text{Velocidad angular rozadora} = (\text{"Lenta"}, \text{"Normal"})$$

$$\text{Estado rozadora} = (\text{"Desgaste importante"}, \text{"Desgaste medio"}, \text{"Desgaste mínimo"}, \text{"Óptimo"})$$

Las funciones de pertenencia se observan en las Figuras #8 y #9:

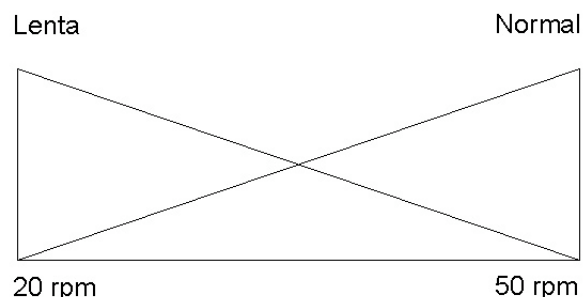


Figura #8: Variable lingüística que representa la velocidad angular de la rozadora en rpm

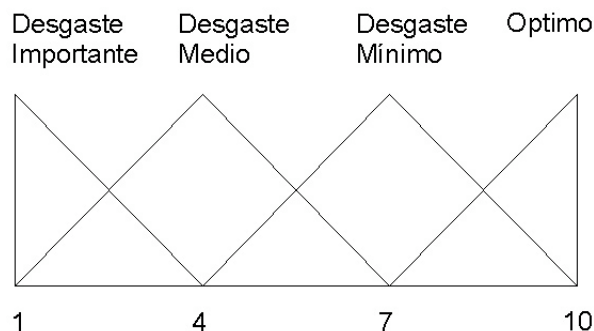


Figura #9: Variable lingüística que representa el estado de la rozadora

La velocidad angular se da en revoluciones por minuto, mientras que el estado de la rozadora se describe mediante un rango numérico del 1 al 10. Este valor se consigue mediante el conocimiento experto del operario del sistema. Las reglas del modelo se exponen en la Tabla #1:

Tabla #1: Reglas del modelo

Velocidad angular (rpm)		
Estado	Lenta	Normal
Desgaste importante	no-conf. grave	no-conf. grave
Desgaste medio	no-conf. grave	no-conf. leve
Desgaste mínimo	no-conf. leve	conforme
Optimo	no-conf. leve	conforme

Como consecuentes podemos utilizar singletons de Sugeno[3] con los valores “no-conformidad grave” = 0.0, “no-conformidad leve” = 0.5 y “conforme” = 1.0. Utilizando cualquier toolbox de construcción de modelos borrosos[4] y dando diversos valores ejemplo de entrada obtenemos los resultados de conformidades buscados (valor b de calidad transferida tras haberse efectuado los procesos de inferencia y desenfoque). Un ejemplo de los valores arrojados por el modelo se muestra en la Tabla #2:

Tabla #2: Entradas y salidas ejemplo del modelo

Estado rozadora	Velocidad (rpm)	Calidad transferida
2	40	0.11
4	25	0.08
8	32	0.70
10	48	0.97

4 Conclusiones

La aplicación de los conceptos de Calidad Total a la minería de carbón constituye un paradigma que incide notablemente en la seguridad de las labores mineras. El tradicional enfoque hacia la satisfacción al cliente es reemplazado por un sistema de organización interna enfocado a procesos debido a que la geología de un yacimiento es quien marca desde un primer momento la calidad obtenida del mineral.

Este enfoque mediante procesos permite una estructura jerarquizada de clientes y proveedores internos donde cada uno de ellos recibe y transfiere una cierta calidad. En este trabajo hemos representado tales calidades como números de variable compleja, habiéndose definido las conformidades y no-conformidades como variables lingüísticas sobre las partes reales (calidad recibida) e imaginarias (calidad transferida) de forma que las situaciones de conformidad se producen cuando un proceso proveedor transfiere calidad a un proceso cliente. El marco formal así establecido habilita el estudio de aquellos indicadores mineros para los que no existe una frontera nítida en cuanto a su conformidad o no-conformidad. Finalmente hemos presentado un caso real de la operación de una rozadora sobre frente de explotación de mineral donde se obtiene la calidad transferida mediante un modelo basado en técnicas Fuzzy-Logic.

Referencias

- [1] Ferrando, M., Granero, J., “Calidad Total: Modelo EFQM de Excelencia”, FC Editorial, 2008
- [2] Besterfield, D. et al, “Total Quality Management”, Prentice-Hall, 2003
- [3] Sugeno, M., Industrial applications of fuzzy control, Elsevier Science Pub. Co., 1985
- [4] Argüelles, L., “DiffuCalc: A simple computational tool for Fuzzy-Logic based modelling”, ESTYLF, 2008

Extracción de Conocimiento en el Desarrollo de Software mediante el Análisis de los Registros de Tiempos de Trabajo

Arturo Peralta¹ Francisco P. Romero² José A. Olivas² Macario Polo²

Departamento de Tecnologías y Sistemas de Información
Universidad de Castilla La Mancha

¹Arturo.Peralta@alu.uclm.es,

²{FranciscoP.Romero, JoseAngel.Olivas, Macario.Polo}@uclm.es

Resumen

Dentro de la disciplina del desarrollo de software así como en otras disciplinas existe un problema inherente en la organización de tareas para realizar un proyecto tanto en su definición como durante su elaboración. La incertidumbre en la definición de la duración de las tareas y el rendimiento de las personas que las realizan conforman un entorno en el que las técnicas de Soft-Computing pueden aportar nuevos enfoques con el fin de ayudar a los participantes del proyecto a administrar su tiempo, priorizar sus actividades y reajustar el trabajo para alcanzar de forma satisfactoria los objetivos marcados. Para ello, a partir del registros de tiempos de cada participante en el proyecto, se extraerá el conocimiento necesario con el que apoyar tanto la toma de decisiones para la distribución de tareas en el trabajo diario así como la elaboración de estimaciones para futuros proyectos.

Palabras Clave: Extracción de Conocimiento, Prototipos Borrosos, Soft-Computing.

1 INTRODUCCIÓN

En la actualidad, uno de los principales problemas al que se enfrentan las empresas es la estimación de la duración de sus proyectos. Habitualmente, en el momento de hacer la estimación, el conjunto de tareas a realizar y su definición no están suficientemente acotados para definirlos de forma completa. Además es habitual que en el transcurso del proyecto aparezcan nuevas tareas o proyectos que afecten al equipo, haciendo muy complicado cumplir su planificación.

Al problema de la definición de las tareas se le une el “factor humano” de la construcción del software, es decir, las personas que realizan el trabajo en sí [6]. El escenario actualmente existente en la consultoría tecnológica de este país está caracterizado por altos niveles de superposición, es decir, o bien muchas personas realizan la misma tarea a la vez o una persona realiza varias tareas a la vez. Además, los desarrolladores suelen estar multiasignados, realizando trabajos de diferentes proyectos en intervalos temporales cortos. A esto hay que añadir la flexibilidad de horarios, en cuanto a las interrupciones durante el trabajo, realización de horas extras e impuntualidad, y la definición difusa de los objetivos a cumplir. Es por ello que el manejo de la incertidumbre será una constante en la gestión del proyecto.

Dado que el desarrollo de productos de software frecuentemente requiere cumplir requisitos del cliente a un coste y planificación acordada, se han definido metodologías que ayuden a los profesionales a realizar su trabajo dentro de una serie de buenas prácticas en Ingeniería del Software. Una de las técnicas más destacadas es el Personal Software Process (PSP) [2]. Dentro del PSP se definen métodos que permiten realizar una mejor planificación y seguimiento del trabajo, estableciendo metas mensurables y ofreciendo la posibilidad de utilizar un proceso bien definido y medido de trabajo.

Actualmente, la estimación de la duración de proyectos software supone una línea de investigación de gran interés, contando con numerosos estudios centrados en el uso de diversas técnicas. Sin embargo, en la mayoría de los casos, el “factor humano”, o las distintas eventualidades hacen que las estimaciones planificadas en una primera aproximación, dejen de ser tan precisas como se pretendía. Ante esta problemática en este trabajo se propone la utilización de diversas técnicas enmarcadas dentro del Soft-Computing como la lógica borrosa, o algoritmos de clustering con el fin de modelar el comportamiento del desarrollador de software en cuanto a la distribución de su tiempo.

Como fuente de datos para realizar este trabajo de extracción de conocimiento se establecen las técnicas y herramientas definidas Personal Software Process (PSP), siempre teniendo en cuenta la dificultad que existe en la recolección de datos básicos reales de trabajo según esta metodología. De esta manera, se pretenden conseguir los siguientes dos objetivos:

1. Detectar patrones de comportamiento en base a la gestión del tiempo que realizan los desarrolladores.
2. Representar este conocimiento mediante mecanismos que sean flexibles y permitan manejar la incertidumbre de forma eficiente para su posterior uso en la planificación de los proyectos.

El resto del trabajo se organiza de la siguiente manera. En la siguiente sección se describen con detalle todos los pasos del proceso de extracción de conocimiento, desde la especificación de las fuentes de datos hasta la construcción de los prototipos borrosos. En la sección 3 se desarrolla el caso de estudio sobre el que se ha extraído el conocimiento para finalizar en la sección 4 con las conclusiones y líneas de trabajo que están actualmente abiertas.

2 EXTRACCIÓN DE CONOCIMIENTO

En este apartado se detalla la aplicación concreta de extracción de conocimiento prototípico (DCPB) sobre los datos brindados por el PSP. El DCPB[5] es una ampliación del proceso clásico de descubrimiento de conocimiento en bases de datos que presenta como novedades la incorporación de conocimiento en diferentes puntos mediante decisiones del usuario o del experto y un resultado preparado para generar unos prototipos conceptuales denominados Categorías Prototípicas Deformables Borrosas, basados en la idea de Prototipos Borrosos de Zadeh [7]. La utilización de la Lógica Borrosa permite conseguir estos resultados de forma más comprensible y útil para el posterior uso en la predicción; con ello, se permite evaluar situaciones nuevas a partir de dichos prototipos, establecer predicciones en cuanto a situaciones reales y también tomar decisiones a partir de dichas predicciones. Además se utilizan otras técnicas como el clustering borroso y las funciones de agregación, facilitando la generación de modelos estructurados, significativos y fácilmente actualizables.

2.1. FUENTES DE DATOS

Sin lugar a dudas, el éxito de cualquier proceso de extracción de conocimiento está condicionado al uso de una fuente fiable de datos.

Sin embargo, habitualmente los desarrolladores de software en pocas ocasiones tienen la costumbre completar con sus datos los registros de tiempos. No obstante, el presente estudio está centrado en el análisis de los datos proporcionados por el PSP de nivel 0 (Tabla 1), registrables por el profesional sin demasiado esfuerzo.

Tabla 1: Registro de Tiempos.

Fecha	Hora		Tiempo de Interrupción	Tiempo	Descripción Actividad
	Inicio	Fin			

En la primera etapa, o “fase de importación”, los datos son obtenidos y procesados para su posterior almacenamiento. Dentro de esta fase se incluye la interpretación del formato en el que esté la información así como la adaptación de la información al modelo a construir.

2.2. LIMPIEZA DE DATOS

Dentro de cualquier metodología de análisis de datos es inevitable realizar inicialmente una limpieza del material con el que se va a trabajar. En este estudio la realización de una limpieza de los datos es un punto especialmente importante dado que se trabaja con datos normalmente de baja calidad [1]. Esto se debe a que es muy habitual que los desarrolladores incluyan en sus datos multitud de errores, que podrían llegar a afectar a las medidas de calidad y productividad. Los diferentes procesos que deben realizarse para la limpieza de la información que será procesada son los siguientes:

- *Detección de sesiones de trabajo incorrectas.* En la primera fase son detectadas aquellas sesiones cuyos datos son incongruentes.
- *Corrección ortográfica.* Mediante el uso de diccionarios se identifica si cada una de las palabras de los descriptores de las tareas o los proyectos existen o no, tratando de corregirla en caso negativo.
- *Agrupación de tareas o proyectos según su nombre.* Analizando los registros de tiempos es habitual encontrar varias tareas o proyectos con nombres ligeramente distintos, que realmente se tratan del mismo. Por ejemplo, las tareas “Analizar-Revisar Software” y “Analizar/Revisar Software” muy probablemente sean la misma. En este punto son agrupadas todas aquellas tareas o proyectos con suficiente similitud en sus nombres. Para ello, es usada una función de similitud, basada en el cálculo de la distancia Euclídea. De este modo es analizada la distancia, a nivel de palabras y del conjunto total del texto, agrupando aquellas descripciones más similares.

2.3. OBTENCIÓN DE MÉTRICAS.

Mediante el análisis de los datos proporcionados y el uso de ciertas métricas, se pretende, como primer objetivo, determinar el comportamiento de un profesional dentro de una dimensión temporal y no de tareas. Para ello es necesario realizar un análisis preliminar con respecto a su tiempo dedicado al trabajo en su distribución diaria y semanal. Posteriormente, el conocimiento extraído de este análisis será utilizado para programar las labores de un profesional durante una o varias semanas teniendo en cuenta todos los factores que determinan su distribución habitual del tiempo (tiempo trabajado según el día, interrupciones, división de una tarea entre días consecutivos, etc.). Además, se pretende determinar ciertos atributos que puedan llegar a caracterizar a cada profesional de acuerdo a su capacidad de trabajo para cada tipo de tareas, su eficiencia o fiabilidad, etc.

Para llevar a cabo este análisis, es necesario estudiar el entorno como un sistema donde el comportamiento de cada individuo va a ser modelado a partir de variables lingüísticas. Estas variables van a cumplir las características de las variables lingüísticas especificadas en [4] para sistemas dinámicos, prescindiendo de la asunción de continuidad.

En los siguientes apartados son detalladas las medidas, su formalización, y las etiquetas lingüísticas seleccionadas para llevar a cabo la caracterización de los días, semanas, y la dedicación de los profesionales.

2.3.1. Métricas de caracterización de días y semanas.

A partir de los datos proporcionados por el libro de ingeniería de cada profesional es posible establecer un conjunto de métricas directas e indirectas, resultado de agregaciones de las anteriores. Para la caracterización de los días se establecen las siguientes métricas directas.

- *TE*: Tiempo en horas empleado por el profesional.
- *TI*: Tiempo en horas perdidas por interrupciones.
- *NS*: Número de sesiones anotadas por el profesional.
- *NSTT*: Número de sesiones en las que se finaliza la tarea que se comienza.

A partir de las anteriores métricas directas es posible el cálculo de las siguientes métricas indirectas:

- *TTT*: Tiempo Total Trabajado ($TTT = TE - TI$).
- *TMS*: Tiempo Medio por Sesión ($TMS = TTT / NS$).

Para definir una medida de similitud entre días o semanas y una medida su rendimiento, son usados los siguientes indicadores, calculables a partir de las anteriores métricas:

- *PROD*: Productividad del día en cuanto al trabajo diario ($PROD = TTT / TTT_{medio}$).

- *EFCC*: Eficacia en la finalización de tareas ($EFCC = NSTT / NS$).
- *INT*: Relación entre el número de horas perdidas debido a interrupciones y el tiempo total efectivo trabajado ($INT = TI / TTT$).

Estos tres indicadores son representados mediante técnicas lingüísticas procedentes de la lógica borrosa usando para ello variables y etiquetas lingüísticas (Fig. 1).

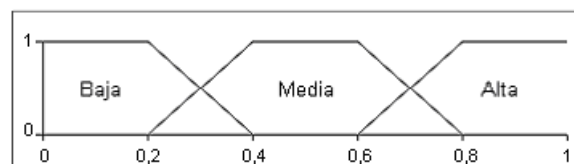


Figura 1: Etiquetas lingüísticas para indicadores.

También es posible establecer métricas de trabajo entre diferentes días o semanas analizando la media del tiempo trabajado, el volumen de interrupciones, y como estas, y los tiempos de las sesiones, afectan al número de tareas finalizadas.

2.3.2. Métricas para caracterización de profesionales.

Para la caracterización del trabajo de cada profesional, son usadas algunas de las anteriores métricas directas usadas para los días y semanas. No obstante, además de estas son definidas las siguientes:

- *NT*: Número de tareas realizadas.
- *TT_i*: Representa el tiempo empleado al desarrollo de tareas de un determinado tipo (análisis, diseño, desarrollo, etc.).
- *NR*: Representa el número de reparaciones, es decir, las sesiones que se marcaron como finalizadas y sin embargo, en una sesión posterior, fueron continuadas.

Para definir una medida de la dedicación a cada tipo de tareas, la fiabilidad o el rendimiento, son usados los siguientes indicadores, calculables a partir de las anteriores métricas.

- *RND*: Representa el rendimiento como la relación entre el número de tareas comenzadas y finalizadas en la misma sesión ($RND = (NSTT - NR) / NS$).
- *FBL*: Fiabilidad, en función del índice de reparaciones efectuadas ($FBL = 1 - NR / NS$).
- *DT_i*: Representa un conjunto de métricas capaces de cuantificar la dedicación a cada tipo de tareas desempeñadas (análisis, diseño, etc.) ($DT_i = TT_i / TT$).
- *DSP*: Indica el grado de dispersión existente en las tareas que realiza ($DPS = NT / NS$).

De modo similar a los indicadores para la caracterización de días y semanas, estos cuatro indicadores del trabajo profesional son representados mediante etiquetas lingüísticas.

2.4. CLUSTERING Y PROTOTIPAJE

Habitualmente la planificación del desarrollo de un proyecto es realizada en base a calendario laboral normal. En él los días laborables tienen un número fijo de horas de trabajo con una cierta distribución temporal. Sin embargo, esta planificación normalmente es insuficiente ya que no tiene en cuenta posibles imprevistos, y sobre todo porque la distribución fija del tiempo no suele ser una distribución realista de como se distribuye el tiempo semanalmente.

Los algoritmos de clustering nos permiten realizar agrupaciones de elementos similares con el fin de descubrir patrones regulares en nuestros datos de estudio. De esta manera, es posible obtener una agrupación de semanas con la cual podremos encontrar distintos tipos de comportamiento. Cada uno de estos comportamientos serán definidos a partir de las características presentes en ellos y ausentes en los demás (por ejemplo, semana con día festivo, o con entrega/deadline) obtenidas mediante un proceso de post-clasificación. De esta forma, se podrá identificar en qué semanas se trabaja mejor y cuales peor, incluso es posible estudiar la evolución de los tiempos de trabajo (y de otros factores) durante la propia semana.

Para llevar a cabo el proceso de DCPB es necesario dividir los atributos del conjunto de datos en tres colecciones no disjuntas:

- *Datos de Modelado*: Para esta colección deben ser elegidos los datos que puedan definir mejor a los prototipos, sin tener en cuenta si se dispone de ellos a priori o no.
- *Datos de Prototipaje*: Está formada por los datos conocidos a priori de una semana. Se usan para realizar la representación formal de los prototipos mediante números borrosos.
- *Datos de Predicción*: Corresponden a los datos, no son conocidos a priori, que pretenden ser averiguados de la semana entrante.

A partir de estos datos, es posible realizar la construcción de un modelo de predicción, con los siguientes objetivos:

- La búsqueda y extracción de forma automática de prototipos borrosos para caracterizar las semanas de trabajo según su productividad y rendimiento.
- La obtención de un modelo de predicción de la productividad y rendimiento de las semanas realizando la deformación de los prototipos borrosos.

A continuación son detallados los procesos llevados a cabo en esta etapa del DPCB para la construcción del modelo de predicción.

2.4.1. Obtención de los prototipos mediante clustering.

Con el objetivo de detectar las relaciones entre las semanas de trabajo que permita determinar su productividad como alta, baja o media, es realizado un proceso de clustering mediante el algoritmo EM (Expectation Maximization) [3], forzando la generación de tres grupos o tipos de semanas. Este algoritmo asigna a cada instancia una distribución de probabilidad de pertenencia a cada cluster, utilizando un modelo Gaussiano finito de mezclas, asumiendo que los atributos son variables aleatorias independientes

Para llevar a cabo el proceso de clustering, son usados los denominamos “*datos de modelado*”. Estos datos corresponden a los tiempos totales de trabajo de cada día y los indicadores del nivel de interrupciones sufridas en la semana, las sesiones de trabajo realizadas, y las tareas finalizadas. De este modo, cada semana es modelada como un vector de 8 parámetros normalizados (toman valores en el intervalo [0,1]).

2.4.2. Representación formal de los prototipos.

A partir de los “*datos de prototipaje*”, que serán las características propias de las semanas representadas mediante indicadores, los tres prototipos obtenidos tras el clustering son representados como “números borrosos”, permitiendo obtener un grado de pertenencia (entre 0 y 1) de una nueva semana con cada uno de ellos.

En el proceso de DCPB la definición formal de los prototipos en forma de números borrosos se realiza en dos pasos: normalización de los datos y agregación mediante la media de los datos de la colección de datos de prototipaje. Tras esto, cada tipo de semana puede ser representado mediante números borrosos triangulares. De esta manera va a ser posible conocer el tipo de una nueva semana a partir del grado de pertenencia obtenido mediante la normalización y agregación de sus “*datos de prototipaje*”. Posteriormente, una vez conocido el tipo será posible conocer cómo va a ser la semana siguiente a partir de la definición paramétrica de cada prototipo que será conformada por los “*datos de predicción*”.

2.4.3 Definición paramétrica de los prototipos.

Una vez obtenidas las agrupaciones de semanas se obtiene la definición paramétrica de cada uno de los prototipos. Para ello se obtiene la media, el mínimo y el máximo de cada uno de los indicadores existentes en los “*datos de predicción*” para cada uno de los grupos. De este modo, es posible establecer un marco de datos de predicción, mediante el cual se consigue conocer las características de una semana una vez conocido el tipo al que pertenece.

Este conocimiento será la base para realizar una planificación adaptativa de las tareas que cada uno de los componentes de un equipo de trabajo debe realizar durante una semana, pudiéndose conocer de antemano los efectos que un posible viaje inesperado o una enfermedad puedan tener sobre la distribución de tareas entre los componentes del equipo.

3 CASO DE ESTUDIO

El escenario para este caso de estudio es una consultora con un equipo de 5 personas responsables de todos los proyectos relacionados con una línea de trabajo. La principal tarea de los empleados es el mantenimiento de un proyecto, que además genera nuevos desarrollos que deben de ser correctamente estimados, planificados y distribuidos en el tiempo. Las entregas de las versiones de mantenimiento se concentran los jueves por la tarde lo cual hace que la distribución del tiempo se vea muy afectada. Para estudiar el comportamiento de este equipo de trabajo se ha partido del análisis de todos los datos expuestos en cada uno de los libros de ingeniería de cada empleado. Estos datos han sido recolectados de forma privada y confidencial. En primer lugar se ha comenzado por una limpieza de los datos recibidos, continuando por una fase de procesamiento y cálculo de métricas, y finalizando con un análisis de las mismas.

En la tabla 2 se muestra el número de sesiones totales realizadas e indicadas por cada empleado en su libro de ingeniería, así como el porcentaje con datos erróneos. Se muestra a su vez, el número de correcciones ortográficas realizadas y el número real de tareas tras su limpieza y agrupación a partir del análisis de sus descripciones.

Tabla 2: Limpieza de datos.

	Nº Total Sesiones	% Sesiones Inválidas	Correcciones ortográficas	Nº Tareas Limpieza
Emp.1	687	2%	24	245
Emp.2	495	1,8%	14	180
Emp.3	325	3,5%	27	215
Emp.4	780	5,7%	38	536
Emp.5	415	1,5%	7	268

Tras la limpieza de datos, se ha realizado un análisis, con el objetivo de calcular las métricas capaces de caracterizar cada empleado, obteniéndose los resultados en cuanto a etiquetas lingüísticas que se muestran en la tabla 3.

Posteriormente, se definen los “datos de modelado” con los cuales se realizará la agrupación de las diferentes semanas de trabajo. Éstos vendrán conformados por las métricas de caracterización de los días y las semanas explicadas en el apartado 2.3.1.

Tabla 3: Caracterización de profesionales.

	Emp.1	Emp.2	Emp.3	Emp.4	Emp.5
Fiabilidad (FBL)	Alta (0.89)	Media-Alta (0.69)	Alta (0.96)	Alta (0.91)	Alta (0.86)
Rendimiento (RND)	Media-Alta (0.65)	Media (0.57)	Media (0.4)	Media-Alta (0.89)	Media-Alta (0.69)
Dispersión (DSP)	Baja-Media (0.36)	Baja-Media (0.37)	Media-Alta (0.68)	Media-Alta (0.73)	Media-Alta (0.66)
Dedicación Tareas Análisis (DT_a)	Baja-Media (0.31)	Baja (0.07)	Media (0.40)	Baja (0.07)	Baja (0.20)
Dedicación Tareas Diseño (DT_d)	Baja (0.16)	Baja (0.06)	Baja (0.11)	Baja (0.06)	Baja-Media (0.31)
Dedicación Tareas Codificación (DT_c)	Baja (0.12)	Media-Alta (0.71)	Baja (0.07)	Media-Alta (0.70)	Baja (0.19)
Dedicación Tareas Mantenimiento (DT_m)	Baja (0.25)	Baja (0.07)	Baja-Media (0.31)	Baja (0.11)	Baja (0.20)

Tras realizar el agrupamiento se han detectado tres tipos distintos de semanas (Figura 2), las cuales se han podido caracterizar como:

- *Semanas con entrega* (64%), en la que se trabaja más, existiendo un pico de trabajo los días de la entrega.
- *Semanas sin entrega* (20%), en la que los tiempos de trabajo son más constantes y bajos.
- *Semanas de escaso trabajo*, incompletas o resultado de una semana anterior de estrés debido a festividades, especialmente en lunes y viernes (14%).

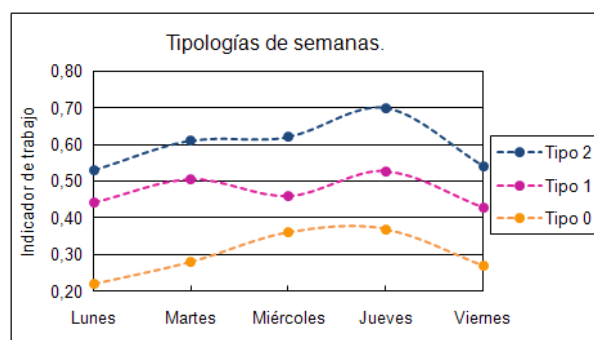


Figura 2: Tipos de semana según nivel de trabajo.

Con este conocimiento es posible inferir que las horas trabajadas al día siguiente de la entrega son inversamente proporcionales a las realizadas el día de la entrega.

Con el fin de representar los formalmente prototipos borrosos con datos conocidos a priori se definen los siguientes nuevos indicadores (Tabla 4) que complementan a las métricas explicadas con anterioridad para formar los “datos de prototipaje”.

Tabla 4: Indicadores para datos de prototipaje.

Dato	Descripción	Dominio
Compleitud	La semana es completa o tiene festivos	Completa (1), con festivo (0)
Docencia	Horas de docencia prevista	0-10
Desplazado	Nº de Días de trabajo desplazado	1-5
Fin de Semana	Previsión de Trabajo para el fin de semana	Nula (0), baja (0,33), media (0,66), alta (1)
Tres Semanas Anteriores	Nº medio de horas trabajadas en las tres semanas anteriores.	0-50

A partir de los datos de cada semana registrada, y el tipo al que pertenecen, se han obtenido los siguientes valores (tabla 5) mediante normalización y agregación.

Tabla 5: Datos de prototipaje de cada tipo de semana.

Dato	Tipo 0	Tipo 1	Tipo 2
Compleitud	0	0,93	0,42
Docencia	0,0	0,55	0,27
Desplazado	0,11	0,72	0,38
Fin de Semana	0,33	0,64	0,62
Semana Anterior	0,59	0,89	0,88

A partir de los mencionados “datos de prototipaje”, cuya agregación conforma el jefe de abscisas, los tres prototipos son representados como números borrosos triangulares (Figura 3).

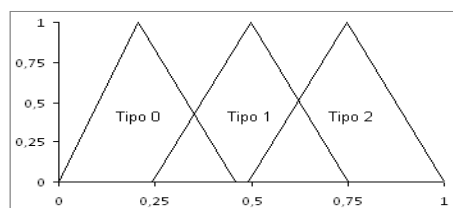


Figura 3: Tipologías representadas como números borrosos.

Haciendo uso de esta representación, es posible conocer el tipo de una nueva semana según el grado de pertenencia obtenido mediante la normalización y agregación de los indicadores anteriormente mencionados. Posteriormente, una vez conocido el tipo, será posible conocer las características de la nueva semana a partir de los “datos de predicción”.

3 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado una serie de técnicas enmarcadas dentro del descubrimiento de conocimiento que permiten el modelado del trabajo de los desarrolladores software en lo que se refiere a la distribución de su tiempo de trabajo. Con ello se han conseguido definir comportamientos en cuanto a tipologías de días o de semanas, o características propias del profesional. Con este fin se han utilizado técnicas como los algoritmos de clustering o las Categorías Prototípicas Deformables Borrosas, entre otras.

El próximo paso es la aplicación real del conocimiento extraído en la planificación y estimación de nuevos proyectos, sobre todo con el objetivo de que la planificación sea flexible y permita responder a crisis e imprevistos de forma eficiente. A su vez se pretende extraer conocimiento de datos procedentes de otros estudios, de simuladores y de experimentos controlados con el fin de poder definir de forma más completa los posibles comportamientos de los componentes de un equipo de desarrollo.

Agradecimientos

La investigación realizada está financiada por los proyectos TIN2007-67494 del Ministerio de Ciencia e Innovación y PEIC09-0196-3018 de la Junta de Comunidades de Castilla-La Mancha.

Referencias

- [1] M. Disney. Data Quality Problems in the Personal Software Process. Thesis of Master of Science in Information and Computer Sciences. University of Hawaii, 1998.
- [2] W. Humphrey. El Proceso de Software Personal. 16th Vol. 1. Prentice Hall, College Station, Texas, 2001.
- [3] G. McLachlan, T. Krishnan. The EM algorithm and extensions. Wiley series in probability and statistics. John Wiley & Sons. 1997
- [4] J. Moreno-García, L. Jiménez, X. Aguado. Modelado en el Tiempo de Sistemas Dinámicos. ESTYLF, 2002.
- [5] J. A. Olivas. Contribución al Estudio Experimental de la Predicción basada en Categorías Deformables Borrosas. Tesis Doctoral. Universidad de Castilla La Mancha, España, 2000.
- [6] Z. Yanyan, X. Renzuo, The Basic Research of Human Factor Analysis Based on Knowledge in Software Engineering. Vol. 5, pp.1302-1305, 2008 Int. Conf. on Computer Science and Software Engineering, 2008
- [7] L. A. Zadeh. A note on prototype set theory and fuzzy sets. Cognition 12, pp. 291- 297, 1982.

OBTENCIÓN DE UN NÚMERO DIFUSO UTILIZABLE EN APLICACIONES A PARTIR DE DATOS ORDENADOS

Juan Moreno-García	Luis Jiménez, Luis Rodríguez	Fuensanta Andrés
E.U.I. Técnica Industrial	Escuela Superior de Informática	E.U.I. Técnica Industrial
D. Tecnologías y Sist. de Información	Dep. Tecn. y Sist. de Información	Dep. de Matemáticas
Universidad de Castilla-La Mancha	Universidad de Castilla-La Mancha	Universidad de Castilla-La Mancha
Toledo	Ciudad Real	Toledo
Juan.Moreno@uclm.es	{Luis.Jimenez,Luis.Rodriguez}@uclm.es	Fuensanta.Andres@uclm.es

Resumen

Los datos imprecisos son tratados mediante números difusos en aplicaciones reales. El problema principal radica en la obtención de la función de pertenencia del número difuso a partir de un conjunto de datos. En este trabajo se trata el problema de la recapitulación de datos como número difuso, más concretamente, se presenta un método para la obtención de un número difuso trapezoidal a partir de un conjunto de datos ordenado. El método tiene un orden de complejidad $O(n^2)$.

Palabras Clave: números difusos, conjuntos de datos ordenados.

difusos mediante funciones trapezoidales facilitan su uso en aplicaciones.

Nuestro trabajo se centra en la obtención de la función de pertenencia del número difuso a partir de una distribución de frecuencias que determina la clase. Para simplificar la manipulación de dichos números generaremos funciones de pertenencia trapezoidales o triangulares. Por tanto se presenta un método para obtener la función de pertenencia del número difuso triangular o trapezoidal que representa al conjunto de datos ordenado.

Existen algunos trabajos previos, Grzegorzewski y Mrówka [3, 4] tratan el problema de obtener una aproximación trapezoidal a los números difusos. Para ello formulan una lista de requerimientos deseables para los operadores de aproximación, obteniendo un operador de aproximación satisfactorio. Delgado [1, 2] obtiene parámetros que capturan la información contenida en un número difuso discreto para simplificar su representación y tratamiento. Con éstos obtienen su representación canónica y posteriormente tratan con ellos en problemas de toma de decisión. Voxman [5] amplía los trabajos anteriores para números difusos no discretos.

En este trabajo se presenta un método para la obtención de un número difuso a partir de un conjunto de datos definido en un dominio ordenado. Está basado en el cálculo de las tres rectas que forman el trapecioide. Las rectas del lado izquierdo y derecho se calculan mediante rectas de regresión. Con la recta central, que representa los valores con pertenencia 1, se busca la mejor colocación de la misma en el eje Y. Las pruebas realizadas demuestran que el método obtiene buenos resultados y es computable.

En la Sección 2 se describe el método. Se muestra un ejemplo de su funcionamiento (Sección 3). En la Sección 4 se realizan las pruebas del método, y el trabajo finaliza exponiendo las conclusiones.

1. INTRODUCCIÓN

Una técnica para el aprendizaje de clasificadores es la estimación de la función $class(x)$, donde un dato $x \in X_1, X_2, \dots, X_n$ son los dominios de los n atributos de los datos, y $class(x) \in C = \{c_1, c_2, \dots, c_m\}$ es la clase asignada al elemento x siendo c_i cualquiera de las posibles clases. El uso de la distribución de frecuencias de los valores de los atributos de los datos de ejemplo es utilizado como aproximación a la función de clasificación $class(x)$. En este tipo de problemas es común representar una determinada clase a través de la frecuencia de aparición de los valores de determinados atributos en los ejemplos. Se establece de esta forma una correspondencia entre la distribución de frecuencias de los valores de dichos atributos con la asignación a una clase determinada. Considerando que los dominios sobre los que están definidos los datos son ordenados, una aproximación para la representación y manejo de estas distribuciones es el uso de conjuntos difusos, y más concretamente de los números difusos. La definición de la función de pertenencia de estos números

2. MÉTODO PROPUESTO

El método calcula el número difuso trapezoidal (NDT) que representa la función de distribución asignada a una determinada clase de los datos de un dominio ordenado. El caso del número difuso triangular se verá como un caso particular dentro del trapezoidal, el que sólo tiene un valor con pertenencia 1.

Algoritmo 1 Proceso de obtención de NDT

```

1. Suavizado de valores de  $Y$ 
   {Obtener lista ordenada de valores y otros índices}
2.  $ordenada \leftarrow \text{ordenar}(Y)$ 
3.  $min \leftarrow \text{ordenado}[0]$ 
4.  $max \leftarrow \text{ordenado}[\text{ordenado}]$ 
   {corte toma la posición del máximo valor en  $Y$ }
5.  $corte \leftarrow |\text{ordenado}|$ 
   {Cálculo de los valores de los extremos del trapezoide  $a$  y  $b$ }
6.  $a \leftarrow \text{Obtenido mediante Ecuación 1}$ 
7.  $d \leftarrow \text{Obtenido mediante Ecuación 2}$ 
while  $ordenada[corte] > media([y_a, y_d])$  do
   {Calcular posiciones derecha e izquierda de corte}
8.  $b \leftarrow \text{Obtenido mediante Ecuación 3}$ 
9.  $c \leftarrow \text{Obtenido mediante Ecuación 4}$ 
   {Calcular rectas izquierda, centro y derecha del trapezoide}
   {Recta de regresión para el intervalo  $[a, b]$ }
10. Calcular Recta Izquierda  $r_{a,b} \leftarrow a_{a,b} + (b_{a,b} * x)$ 
   {Recta de regresión para el intervalo  $[c, d]$ }
11. Calcular Recta Derecha  $r_{c,d} \leftarrow a_{c,d} + (b_{c,d} * x)$ 
   {Calcular Recta centro para intervalo  $[b, c]$ }
12. Calcular Recta Centro  $r_{b,c}$ 
   {Cálculo Bondad Trapecio Obtenido}
13.  $bd_{a,b} \leftarrow \text{Bondad}(r_{a,b})$  (Ecuación 5)
14.  $bd_{b,c} \leftarrow \text{Bondad}(r_{b,c})$  (Ecuación 5)
15.  $bd_{c,d} \leftarrow \text{Bondad}(r_{c,d})$  (Ecuación 5)
16. Cálculo bondad del trapezoide (Ecuación 6)
   {Seleccionar trapecio si obtenido es mejor}
if  $bd_{trap} < bd_{act}$  then
   17.  $bd_{act} \leftarrow bd_{trap}$ 
   18. Actualizar valores trapezoide con  $a, b, c, d$ .
end if
   {bajar línea de corte en ordenada}
19.  $corte \leftarrow corte - 1$ 
20. Recalcular  $a$  y  $d$  para que  $y_a = 0$  e  $y_d = 0$ 
end while

```

Sea una variable estadística unidimensional definida sobre el dominio X con una distribución de frecuencia $f(X) = Y$. Se representará de la siguiente forma:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

La salida es el número difuso trapezoidal (NDT) que representa a la aproximación de la distribución

de frecuencias dada. El número difuso se representará con cuatro componentes, $NDT = \{a, b, c, d\}$ donde a, b, c, d son los cuatro valores del trapezoide.

La idea del método consiste en un proceso iterativo que va descendiendo y cortando el eje Y desde los valores más altos a los más bajos. Se calculan las rectas izquierda (desde a hasta b) y derecha (desde c hasta d) del trapezoide mediante las rectas de regresión de Y sobre X , y la recta paralela al eje X del trapezoide (pertenencias a 1) como la recta que une ambas (desde b hasta c). Después se calcula la bondad del trapezoide obtenido, y si es mejor que el obtenido hasta ese instante se selecciona. El proceso termina cuando el valor de corte en el eje Y llega a la media de los valores desde y_a a y_d .

El Algoritmo 1 muestra en detalle el proceso seguido para obtener el NDT. En determinados problemas, la distribución de frecuencias obtenida presenta saltos abruptos en sus valores que provocan grandes picos en su gráfica, por ejemplo ver Entrada R de la Figura 1. Para eliminar dichos picos se propone una fase inicial de suavizado (Sentencia 1). Esta consiste en que cada valor y_i es sustituido por la media de los valores comprendidos entre y_{i-t} e y_{i+t} .

La Sentencia 2 crea una lista ordenada (llamada *ordenada*) a partir de los valores de Y . Esta lista es utilizada para calcular la ubicación de la recta con pertenencias a 1 en el eje Y en los trapezoides candidatos. En las Sentencias de la 3 a la 5 se obtienen las variables min , max y $corte$: min y max son los valores mínimos y máximos del soporte Y respectivamente, y $corte$ es la variable que almacena la posición donde se corta el eje Y para colocar la recta con las pertenencias a 1. Los cortes se van realizando por los valores de Y de mayor a menor, es decir, los valores de *ordenada* del último al primero (Sentencia 5).

En la fase experimental se han detectado problemas con los valores de y cercanos a 0 para determinar el soporte $[a, d]$ del NDT buscado. El trapezoide cubre un soporte muy amplio y el intervalo $[a, d]$ engloba un número de datos superior al estrictamente necesario (Figura 4). Al calcular las rectas de regresión se cubren las zonas con valores bajos provocando rectas que no se ajustan correctamente a las zonas con valores más altos y más fiables. Por ello se considera que un valor pertenece al soporte si supera un porcentaje del valor máximo de la distribución de frecuencias. En este trabajo le llamaremos umbral de soporte y lo representaremos como U_α , siendo α el porcentaje de corte. En las pruebas realizadas el α se ha fijado al 10 %.

A continuación se obtienen los valores que definen el valor mínimo y máximo del soporte del número difuso (a y d respectivamente). La posición de a , que se ob-

tiene mediante la sentencia 6 (Ecuación 1), se asigna al menor x_i desde 0 a x_n que verifica que $y_i > U_\alpha$.

$$\operatorname{argmin}_{x_i \in [0, x_n]} y_i > U_\alpha \quad (1)$$

La sentencia 7 fija la posición de d (Ecuación 2) al mayor x_i desde 0 a x_n que verifica que $y_i > U_\alpha$.

$$\operatorname{argmax}_{x_i \in [0, x_n]} y_i > U_\alpha \quad (2)$$

El bucle *while* calcula las tres rectas que forman el trapezoide, y posteriormente se obtiene la bondad del trapezoide obtenido. Primeramente se obtiene el valor de b y c del trapezoide (Sentencias 8 y 9 del bucle). b se asigna al menor x_i desde a a d que verifica que $y_i \geq \text{ordenada}[\text{corte}]$ (Ecuación 3).

$$\operatorname{argmin}_{x_i \in [a, d]} y_i \geq \text{ordenada}[\text{corte}] \quad (3)$$

De forma análoga, c toma el valor del mayor x_i que verifica que $y_i \geq \text{ordenada}[\text{corte}]$ (Ecuación 4).

$$\operatorname{argmax}_{x_i \in [a, d]} y_i \geq \text{ordenada}[\text{corte}] \quad (4)$$

Una vez obtenidos los puntos de trapezoide (a , b , c y d) se pueden obtener las rectas que forman el mismo. Para obtener las rectas $r_{a,b}$ y $r_{c,d}$ se hace uso de la regresión lineal. La Sentencia 10 calcula la recta de regresión de Y sobre el eje X comprendida en el intervalo $[a, b]$ ($r_{a,b}$). La Sentencia 11 calcula la recta de regresión de Y sobre el eje X en el intervalo $[c, d]$ ($r_{c,d}$). La Sentencia 12 calcula la recta central ($r_{b,c}$), la comprendida en el intervalo $[b, c]$. Es uno de los pasos críticos del método propuesto debido a que el valor y_b obtenido por $r_{a,b}$ no tiene porque ser cercano al valor y_c obtenido por $r_{c,d}$, provocando que haya escalones en el trapezoide (Figura 2). Para solucionar esto se debe recalcular el valor de b o de c mediante el Algoritmo 2 (Sección 2.1). Una vez obtenidos los valores definitivos de b y c , la recta paralela al eje Y comprendida entre b y c toma el valor de la Ecuación $r_{b,c} = a_{b,c}$.

A continuación se calcula la bondad del trapezoide obtenido, para ello se calcula la bondad de las rectas $r_{a,b}$, $r_{b,c}$ y $r_{c,d}$ (Sentencias 13, 14 y 15 respectivamente) mediante la Ecuación 5 obteniendo las bondades $bd_{a,b}$, $bd_{b,c}$ y $bd_{c,d}$ respectivamente.

$$bd_{j,k} = \frac{\sum_{i=j}^k (y_i - (a_{j,k} + (b_{j,k} * x_i)))^2}{|[j, k]|} \quad (5)$$

donde $i \in [j, k]$, $|[j, k]|$ es el número de elementos del intervalo $[j, k]$, $y_i \in Y$, y $a_{j,k}$ y $b_{j,k}$ definen la recta a calcular la bondad ($r_{j,k}$).

Como puede verse, la ecuación calcula el error cuadrático medio entre la recta obtenida y la distribución de frecuencias de entrada del eje Y del intervalo que cubre la recta.

La Sentencia 16 calcula la bondad del trapezoide (bd_{trap}) mediante la Ecuación 6 que realiza la media ponderada de las bondades de las tres rectas.

$$\frac{(bd_{a,b} * |[a, b]|) + (bd_{b,c} * |[b, c]|) + (bd_{c,d} * |[c, d]|)}{|[a, b]| + |[b, c]| + |[c, d]|} \quad (6)$$

Finalmente, si la bondad obtenida bd_{trap} es menor que la mejor hasta el momento (bd_{act}), ésta toma el valor de bd_{trap} (Sentencia 17), y se actualizan los valores que definen el trapezoide (Sentencia 18). La sentencia 19 decrementa *corte* para calcular el trapezoide del siguiente punto de *ordenada*.

De esta forma el *NDT* queda determinado por los puntos ($a, a_{a,b} + (b_{a,b} * a)$), ($b, a_{b,c}$), ($c, a_{b,c}$), ($d, a_{c,d} + (b_{c,d} * d)$). El soporte de Y para estos puntos está comprendido entre 0 y $a_{b,c}$, por lo que se debe obtener su equivalente para valores de salida en $y_i \in [0, 1]$.

b y c mantienen su valor y representarán el comienzo y fin del intervalo de x que ofrece como salida $y = 1$, es decir, $r_{c,d} = 1$. Sin embargo, si el *NDT* está en los límites del dominio de X , los valores de a y d puede que no verifiquen que $y_a = 0$ e $y_d = 0$ (Figura 3). Por ello se debe recalcular el valor de a y d para que $y_a = 0$ e $y_d = 0$ (Sentencia 20). Para calcular el punto de corte de a con 0 se debe verificar la ecuación $r_{a,b} = a_{a,b} + (b_{a,b} * a) = 0$, y despejando se obtiene que $a \leftarrow \frac{-a_{a,b}}{b_{a,b}}$. El valor de d se obtiene de igual forma que el caso anterior ($d \leftarrow \frac{-a_{c,d}}{b_{c,d}}$).

Una vez obtenido el *NDT* su función de pertenencia queda determinada por la Ecuación 7. El método propuesto tiene una complejidad de orden $O(n^2)$. Ésta viene determinada por la complejidad del bucle, que tiene sentencias de orden $O(n)$, y la propia iteración del bucle ($O(n)$), lo que provoca que la complejidad total del método sea $O(n^2)$.

$$(x \leq a) \rightarrow 0$$

$$(x > a) \text{ and } (x < b) \rightarrow \frac{(x-a)}{b-a}$$

$$(x \geq b) \text{ and } (x \leq c) \rightarrow 1 \quad (7)$$

$$(x > c) \text{ and } (x < d) \rightarrow \frac{(d-x)}{d-c}$$

$$(x \geq d) \rightarrow 0$$

2.1. SELECCIÓN DE LOS VALORES DEFINITIVOS DE b y c

El Algoritmo 2 recalcula el valor de b o de c para que el máximo valor de $r_{a,b}$ (y_b) sea lo más parecido posi-

ble al máximo valor de $r_{c,d}(y_c)$, evitando así que haya escalones en la gráfica. La Figura 2 muestra un ejemplo, como puede verse el máximo valor de la recta $r_{a,b}$ es mucho mayor que el máximo valor de la recta $r_{c,d}$, lo que provoca que aparezca un escalón al colocar la recta $r_{b,c}$ paralela al eje X .

Algoritmo 2 Cálculo de b y c definitivos

```

{Cálculo valores en posiciones  $b$  y  $c$  para el eje  $Y$ }
1.  $y_b \leftarrow a_{a,b} + (b_{a,b} * x_b)$ 
2.  $y_c \leftarrow a_{c,d} + (b_{c,d} * x_c)$ 
if  $a = b$  or  $c = d$  then
  { $b$  y  $c$  mantienen su valor en ambos casos}
  if  $a = b$  then
    {no existe  $r_{a,b}$ }
    {Pertenencias a 1 en el máximo valor de  $r_{c,d}$ }
    3.  $a_{b,c} \leftarrow y_c$ 
  else
    {no existe  $r_{c,d}$ }
    {Pertenencias a 1 en el máximo valor de  $r_{a,b}$ }
    4.  $a_{b,c} \leftarrow y_b$ 
  end if
else
  {Existen  $r_{a,b}$  y  $r_{c,d}$ }
  if  $y_c \leq y_b$  then
    {Se decrementa  $b$  hasta que  $y_c \leq y_b$ }
    while  $y_c < y_b$  do
      5.  $b \leftarrow b - 1$ 
      6.  $y_b \leftarrow a_{a,b} + (b_{a,b} * x_b)$ 
    end while
    {corte a  $y_c$  ya que  $y_c \geq y_b$ }
    7.  $a_{b,c} \leftarrow y_c$ 
    { $b$  toma valor que cumple  $r_{a,b} = a_{b,c}$ }
    8.  $b \leftarrow \frac{a_{b,c} - a_{a,b}}{b_{a,b}}$ 
  else
    {Se decrementa  $c$  hasta que  $y_b < y_c$ }
    while  $y_b < y_c$  do
      9.  $c \leftarrow c + 1$ 
      10.  $y_c \leftarrow a_{c,d} + (b_{c,d} * x_c)$ 
    end while
    {corte a  $y_b$  ya que  $y_b \geq y_c$ }
    11.  $a_{b,c} \leftarrow y_b$ 
    { $c$  toma valor que cumple  $r_{c,d} = a_{b,c}$ }
    12.  $c \leftarrow \frac{a_{b,c} - a_{c,d}}{b_{c,d}}$ 
  end if
end if

```

Las Sentencias 1 y 2 calculan el valor y_b e y_c que ofrecen las rectas $r_{a,b}$ y $r_{c,d}$ respectivamente. A continuación se comprueba si realmente se trabaja sobre un trapezoide o no, es decir, si el trapezoide está compuesto por tres rectas o no existe la recta izquierda o derecha del mismo ($r_{a,b}$ o $r_{c,d}$ respectivamente). La recta $r_{a,b}$ existe si $a \neq b$ (primer *if* dentro del primer *if*), y análogamente $r_{c,d}$ existe si $c \neq d$ (segundo *if*

dentro del primer *if*). En este caso, la recta de pertenencias a 1 ($r_{b,c}$) se fija en el máximo valor de la recta que existe: $a_{b,c} = y_c$ si no existe $r_{a,b}$ (Sentencia 3), o $a_{b,c} = y_b$ si no existe $r_{c,d}$ (Sentencia 4). El *else* del primer *if* trata el caso del trapezoide completo, es decir, existen $r_{a,b}$ y $r_{c,d}$. La idea consiste en bajar la ubicación de $r_{b,c}$ en el eje Y para que y_b e y_c estén lo más paralelos posible. Por ello, si y_b es mayor que y_c se va decrementando b (Sentencia 5) y calculando el nuevo valor de y_b (Sentencia 6) hasta que $y_c \geq y_b$. La línea de corte se coloca en y_c (Sentencia 7), y la Sentencia 8 asigna el valor de b al valor x que provoca el corte de las rectas $r_{a,b}$ y $r_{b,c}$, es decir, el que cumple que $y_c = r_{a,b}$, luego $b \leftarrow \frac{a_{b,c} - a_{a,b}}{b_{a,b}}$.

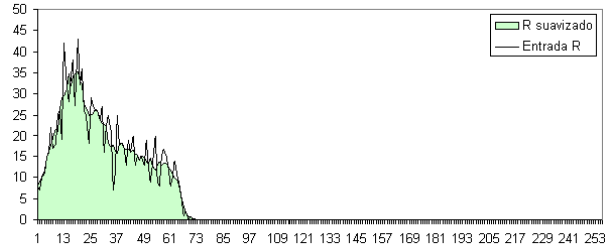


Figura 1: Proceso de suavizado de componente R

Si y_c es mayor que y_b se va incrementando c (Sentencia 9) y calculando el nuevo valor de y_c (Sentencia 10) hasta que $y_b \geq y_c$. La línea de corte se coloca en y_b (Sentencia 11), y análogamente al caso anterior, c toma el valor $b \leftarrow \frac{a_{b,c} - a_{c,d}}{b_{c,d}}$. La Figura 3 muestra cómo el algoritmo soluciona el problema del escalón mostrado en la Figura 2.

3. EJEMPLO DE FUNCIONAMIENTO

Para ver el funcionamiento del método propuesto se verá un ejemplo de su uso en una aplicación real. La aplicación procesa imágenes satélite en color en formato RGB para detectar olivos. Dado que la información de entrada es imprecisa, se ha decidido utilizar números difusos para representar los píxeles de los olivos. Por otro lado, se tiene una imagen satélite RGB de un campo de olivos en la que los píxeles están correctamente etiquetados como olivo, suelo u otro caso. A partir de esta imagen se puede calcular el NDT de cada una de sus componentes utilizando como distribución de frecuencias el histograma de cada una de las componentes RGB de los píxeles que representan a los olivos. La Figura 1, mediante la gráfica etiquetada como *Entrada R*, muestra el histograma de la componente R que será tomado como entrada al método en el ejemplo.

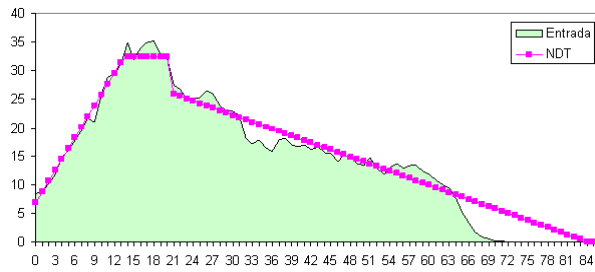


Figura 2: NDT con escalón

El primer paso es el suavizado con $t = 2$ que obtiene el resultado de la Figura 1 mediante la gráfica con la etiqueta *suavizado R*. A continuación se calculan las variables *ordenada*, *min*, *max* y *corte* (Sentencias de la 2 a la 5). *ordenada* es la lista de valores del histograma ordenados, *min* toma el valor 0, *max* es igual a 35,20 y el corte inicial en el eje Y es igual a *max* (*ordenada*[*corte*] = 35,20). Las Sentencias 6 y 7 calculan los valores de *a* y *d* con un $U_\alpha = 10\%$, *a* y *d* toman los valores 0 y 65 respectivamente.

Ya dentro del bucle *while*, las Sentencias 8 y 9 calculan los valores de *b* y *c* para la primera iteración, tomando ambos el valor 18. Con estos valores, el *NDT* inicial propuesto es $\{0, 18, 18, 65\}$, luego existen las rectas $r_{a,b}$ ($0 \neq 18$) y $r_{c,d}$ ($18 \neq 65$). La Sentencia 10 calcula la recta de regresión $r_{a,b}$, quedando definida por los valores $a_{a,b} = 7,9091$ y $b_{a,b} = 1,6705$. La recta $r_{c,d}$ toma el valor $36,3153 + (-0,4464 * x_i)$ (Sentencia 11). La Sentencia 12 calcula la recta central, al no estar correctamente alineadas las dos rectas de regresión (Figura 2) se recalcula el valor de *b* o *c* (Algoritmo 2), manteniendo *c* su valor y tomando *b* el valor 12,1949, el corte definitivo ($a_{b,c}$) pasa a valer 28,2809 y la recta $r_{b,c} = 28,2809$.

A continuación se calculan las bondades de cada una de las rectas mediante la Ecuación 5 obteniendo $bd_{a,b} = 1,3269$, $bd_{b,c} = 23,1222$ y $bd_{c,d} = 5,3426$ (Sentencias 13, 14 y 15). La Sentencia 16 calcula la bondad del trapecio obtenido aplicando la Ecuación 6, obteniendo que $bd_{trap} = 6,4338$. Las Sentencias 17 y 18 actualizan los valores de bd_{act} y del trapecio al ser la primera ejecución y finalmente se baja la línea de corte.

La siguiente ejecución establece el corte inicial en 34,80 y el *NDT* inicial a $\{0, 14, 18, 65\}$. Las rectas de los lados del trapecioide son $r_{a,b} = 6,7972 + (1,8874 * x_i)$ y $r_{c,d} = 36,3153 - (0,4464 * x_i)$. El cálculo de la recta $r_{b,c}$ provoca que *b* cambie de valor y se re-coloque $a_{b,c}$, quedando el corte definitivo en 28,2809 y el *NDT* definitivo como $\{0, 11,3828, 18, 65\}$, luego $r_{b,c} = 28,2809$. Las bondades de las rectas son $bd_{a,b} = 1,0970$, $bd_{b,c} = 19,8575$ y $bd_{c,d} = 5,3426$. La bondad

del trapecioide es $bd_{trap} = 6,2856$. Esta bondad mejora la bd_{act} con lo que se actualizan los valores de bd_{act} y del trapecio.

Para esta nueva ejecución del bucle el corte inicial toma el valor 34,00 y el *NDT* inicial es $\{0, 14, 18, 65\}$. Las rectas del trapecioide son $r_{a,b} = 6,7972 + (1,8874 * x_i)$ y $r_{c,d} = 36,3153 - (0,4464 * x_i)$. Se calcula la recta $r_{b,c}$ tomando ésta el valor 28,2809, y siendo el resto de valores los siguientes: $a_{b,c} = 28,2809$, $NDT = \{0, 11,3828, 18, 65\}$. Las bondades son $bd_{a,b} = 1,0970$, $bd_{b,c} = 19,8575$ y $bd_{c,d} = 5,3426$, lo que implica que la bondad del trapecioide es $bd_{trap} = 6,2856$. Al no ser mejor que bd_{act} no se modifica.

Para la siguiente ejecución el corte inicial se establece en 32,80 y el *NDT* inicial es $NDT = \{0, 14, 19, 65\}$. Las rectas del trapecioide son $r_{a,b} = 6,7972 + (1,8874 * x_i)$, $r_{b,c} = 23,2986$ y $r_{c,d} = 35,3616 - (0,4272 * x_i)$, quedando el corte definitivo en 27,2457 y el *NDT* definitivo en $\{0, 10,8343, 19, 65\}$. Las bondades de las rectas son $bd_{a,b} = 1,0566$, $bd_{b,c} = 29,6076$ y $bd_{c,d} = 4,3482$, y la del trapecioide es $bd_{trap} = 7,3157$. No se mejora bd_{act} .

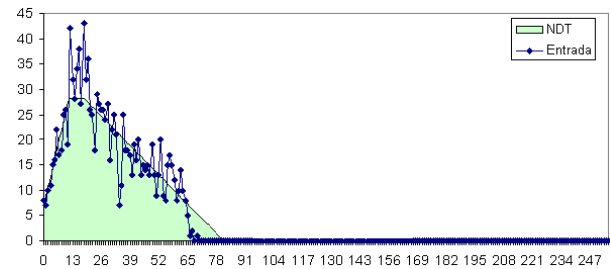


Figura 3: NDT obtenido para el ejemplo

En el resto de iteraciones del bucle la bondad del trapecioide se va incrementando iteración a iteración hasta llegar a un valor máximo de 33,8867.

De todos los trapecoides anteriores se ha seleccionado el de mejor bondad, es decir, el que obtiene la bondad $bd_{trap} = 6,2856$. Como puede verse, hay dos casos consecutivos, el algoritmo selecciona el primero de ellos. Luego el trapecioide final se muestra en la Figura 3. Se concluye que el error es pequeño y se observa un buen ajuste.

4. PRUEBA DEL MÉTODO

Para probar el método propuesto se ha utilizado función continua y otra discreta. Como función continua se ha seleccionado la función gaussiana y como función discreta se ha probado tomando como entrada el histograma de la componente B de los píxeles del olivo de la aplicación de la Sección 3.

Para poder realizar las pruebas de la función continua se ha definido un conjunto de datos con los valores que toma la función con un incremento fijo entre cada valor x_i a muestrear. Para la función gaussiana se ha utilizado la Ecuación $e^{-k*(x_i-m)^2}$ con $k = 3$, $m = 8$ e $\text{incremento} = 0,05$. La primera prueba ha consistido en obtener el NDT de la función gaussiana con $U_\alpha = 0\%$. La Figura 4 muestra el resultado obtenido. Se obtiene una bondad de 0,0473 y un resultado que no es del todo satisfactorio, ya que intenta cubrir todos los valores del dominio (incluidos los bajos). El NDT obtenido es igual a $\{43,00, 160,00, 178,54, 220,00\}$. La siguiente prueba se ha realizado con $U_\alpha = 10\%$, y se obtiene una bondad de 0,0006 aplicando la Ecuación 6, consiguiendo un ajuste perfecto y cubriendo perfectamente el dominio (Figura 5). El NDT obtenido es igual a $\{143,00, 158,00, 162,00, 177,00\}$, como puede verse el soporte cubierto por el NDT es bastante inferior al caso anterior y se ajusta mejor. Esto muestra que un $U_\alpha \neq 0\%$ es necesario en muchas ocasiones.

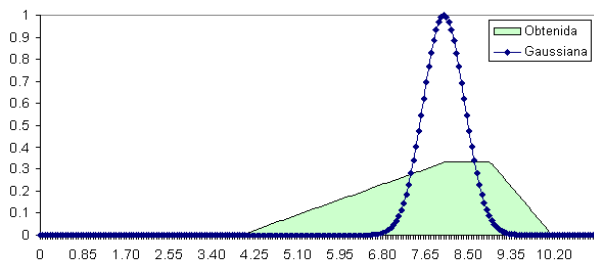


Figura 4: Resultado gaussiana con $U_\alpha = 0\%$.

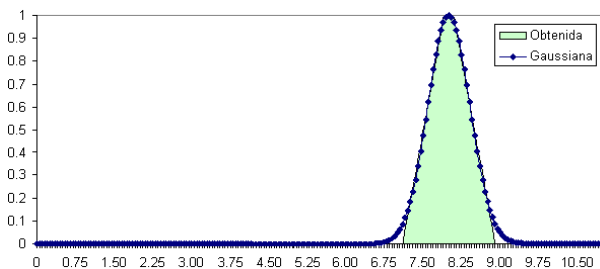


Figura 5: Resultado gaussiana con $U_\alpha = 10\%$.

La última prueba realizada ha tomado como entrada el histograma de la componente B de los píxeles del olivo de la aplicación de la Sección 3. La Figura 6 muestra tanto los datos de entrada como el resultado obtenido. Como puede verse se obtiene un buen ajuste en la recta $r_{c,d}$, y el error obtenido es de 375,3799. El NDT obtenido es igual a $\{0,00, 0,00, 1,00, 37,00\}$.

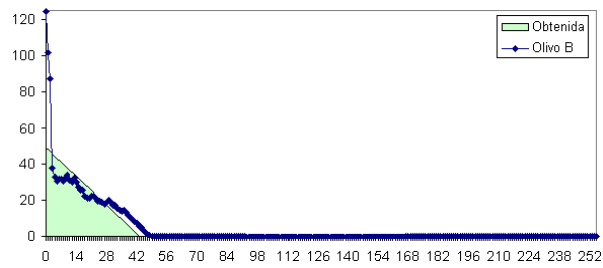


Figura 6: Resultado componente B de olivo.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Se ha presentado un método novedoso para la obtención del número difuso trapezoidal a partir de un conjunto de datos ordenado. El método se basa en el cálculo de las rectas de regresión de los dos lados del trapecioide. Es un método computable de orden $O(n^2)$ que puede ser utilizado en aplicaciones reales. Los resultados presentados pueden ser considerados como válidos, al obtener poco error y ajustarse el resultado a las gráficas de los datos de entrada. Basándonos en las pruebas podemos afirmar que los resultados son mejores si se aplica un $U_\alpha \neq 0\%$ a las frecuencias de distribución de entrada para eliminar los valores bajos.

Agradecimientos

Este trabajo ha sido financiado por la Consejería de Educación y Ciencia de Castilla-La Mancha a través de los proyectos PII2I09-0052-3440 y PII1C09-0137-6488, y por el Ministerio de Ciencia y Tecnología a través de los proyectos TIN2009-14538-C02-02.

Referencias

- [1] M. Delgado, M.A. Vila, W. Voxman. On a canonical representation of fuzzy numbers. *Fuzzy Sets and Systems*, Vol. 93, Pág. 125-135, 1998.
- [2] M. Delgado, M.A. Vila, W. Voxman. A fuzziness measure for fuzzy numbers: applications. *Fuzzy Sets and Systems*, Vol. 94, Pág. 205-216, 1998.
- [3] P. Grzegorzewski, E. Mrówka. Trapezoidal approximations of fuzzy numbers. *Fuzzy Sets and Systems*, Vol. 153, Pág. 115-135, 2005.
- [4] P. Grzegorzewski, E. Mrówka. Trapezoidal approximations of fuzzy numbers—revisited. *Fuzzy Sets and Systems*, Vol. 158, Pág. 757-768, 2007.
- [5] W. Voxman. Canonical representations of discrete fuzzy numbers. *Fuzzy Sets and Systems*, Vol. 118, Pág. 457-466, 2001.

SISTEMAS DE AFINACIÓN BORROSOS: UN ESTUDIO DEL TEMPERAMENTO IGUAL DE 12 NOTAS.

Teresa León Mendoza¹

Vicente Liern Carrión²

¹Dept. Estadística e I.O. Universitat de València. Dr. Moliner, 50, 46100 Burjassot, teresa.leon@uv.es

²Dept. Matemática para la Economía y la Empresa, Universitat de València. Av. Tarongers, s/n, 46022, València, vicente.liern@uv.es

Resumen

Un sistema de afinación es el conjunto de los sonidos que utiliza la Música. Los sonidos admitidos por el sistema de afinación se denominan sonidos afinados o notas musicales. A lo largo de la Historia han aparecido centenares de afinaciones de las que sólo se siguen utilizando alrededor de media docena, las cuatro que conviven en la orquesta clásica actual son la afinación Pitagórica, la Justa entonación, el temperamento igual de 12 notas y el temperamento de Hölder. En la práctica, para conseguir que la orquesta “suene bien”, los músicos modifican “un poco” las notas teóricas. De acuerdo con la lógica booleana podríamos decir que los músicos desafinan, sin embargo la lógica borrosa nos proporciona un modelo flexible que explica satisfactoriamente lo que ocurre en realidad. Consideramos las notas musicales como números borrosos y por tanto los sistemas de afinación estarán formados por notas borrosas. El concepto de similitud entre notas borrosas nos permite justificar teóricamente la práctica diaria de los músicos. Prestamos especial atención al temperamento igual de 12 notas ya que es el sistema más empleado por sus ventajas teóricas y prácticas. A cada una de sus notas (borrosas) le podemos asociar un conjunto de notas que son similares a ella y que serían “intercambiables” por ella.

Palabras Clave: Sistemas de afinación, Temperamento igual de 12 notas, Números borrosos trapezoidales, Similitud.

1 INTRODUCCIÓN

Hay dos acepciones fundamentales de afinación en música: el acto de afinar un instrumento (o una voz) y

también un conjunto de sonidos utilizados para hacer música y su base teórica. Además del indudable interés teórico del estudio de los sistemas de afinación, las necesidades de músicos y musicólogos proporcionan un motivo mucho más convincente que justifica la necesidad de su estudio. Estos han establecido dos campos de actuación: la búsqueda de nuevas afinaciones que aumenten las posibilidades en la creación musical y la recuperación de la fidelidad a partituras antiguas.

En 1948, N. A. Garbuzov publicó un trabajo, *The zonal nature of the human aural perception*, en el que se analizan doce compases del aria de la suite en Re de J. S. Bach interpretados por tres afamados violinistas. En este artículo se demostraba que, con una precisión de cinco cents, la gran mayoría de las notas interpretadas por estos violinistas no pertenecían al sistema de afinación en el que los músicos pensaban que estaban afinados, el temperamento igual de doce notas. Sin embargo, cuando se escuchaba el pasaje, la sensación no sólo era agradable sino que, incluso personas dotadas de un oído muy sensible, calificaron su interpretación como bien afinada (aunque el trabajo original se publicó en ruso, puede encontrarse una descripción de esta publicación en [2].)

Entonces, ¿a qué nos referimos cuando aceptamos que una nota o pasaje están bien afinados? La respuesta a esta pregunta ha sido y continúa siendo motivo de discusión de muchos estudiosos de la Acústica Musical. Además, en la orquesta clásica conviven instrumentos con características físicas diferentes y los sistemas en los que afinan no son los mismos. De hecho, podemos suponer que en una orquesta clásica con coros aparecen, al menos, cuatro formas diferentes de afinar: la afinación pitagórica, la Justa entonación, el temperamento igual de 12 notas y el temperamento de Hölder. En estas circunstancias, ¿es posible afinar una orquesta sinfónica? (Para ampliar información acerca de estos sistemas puede consultarse por ejemplo [1]).

En este trabajo consideramos las notas musicales como números borrosos trapezoidales y por tanto los sistemas de afinación estarán formados por notas borrosas. El concepto de similitud entre notas borrosas nos permite justificar teóricamente la práctica diaria de los músicos.

2. CONCEPTOS PREVIOS

Normalmente se identifica cada nota musical con la frecuencia de su armónico fundamental (la frecuencia que miden los afinadores cromáticos). La forma habitual, en música, de relacionar dos frecuencias es a través de su cociente y a este número se le llama *intervalo*.

En todos los sistemas de afinación aparece el concepto de *octava*. Un sonido de frecuencia f_1 se dice que es una octava más grave que otro f_2 si $f_2 = 2 \cdot f_1$.

Esta idea se usa de forma natural aunque no se tenga formación musical. Piénsese, por ejemplo, cuando cantan juntos un hombre y una mujer. El hombre suele cantar una octava más grave y sin embargo, cualquiera reconoce que están interpretando las mismas notas.

A partir del concepto de octava, se parte el intervalo de frecuencias audibles por octavas:

$$[f, 2f], [2f, 4f], [4f, 8f], \dots$$

y se identifican las notas que están a diferente octava. Es decir, hablaremos de un Do sin importarnos la octava en la que se encuentra. Por tanto, es mucho más cómodo suponer que las notas están en el intervalo $[1, 2[$. De acuerdo con esto, "afinar es elegir una cantidad finita de puntos del intervalo $[1, 2[$ ". La siguiente relación más utilizada es la basada en el cociente $3/2$ que es el intervalo de la quinta perfecta.

En el temperamento igual de 12 notas, el que utiliza la mayoría de la música que escuchamos en la actualidad, la octava se divide en 12 partes igualmente espaciadas, sin embargo otros sistemas "eligen" sus notas de acuerdo con otros criterios. Por ejemplo, todas las notas de la afinación pitagórica se obtienen subiendo o bajando quintas, es decir, dada una frecuencia f multiplicamos o dividimos por $3/2$ cualquier número de veces.

Los sistemas de afinación (como el Pitagórico) basados en un único intervalo admiten una construcción matemática directa, sin embargo aquellos sistemas generados por más de un intervalo necesitan que se especifique cuándo y cuántas veces aparece cada intervalo. A continuación damos una definición general de *sistema de afinación* ([3]).

Definición 1. Dada una familia de escalares $\Lambda = \{\lambda_i\}_{i=1}^k \subset [0, 1[$ y $F = \{h_i : Z \rightarrow Z\}_{i=1}^k$ una familia de funciones, llamamos sistema de afinación generado por los intervalos $\{\lambda_i\}_{i=1}^k$ y F al conjunto

$$S_\Lambda^F = \left\{ 2^{c_n} : c_n = \sum_{i=1}^k \lambda_i h_i(n) - \left\lfloor \sum_{i=1}^k \lambda_i h_i(n) \right\rfloor, n \in Z \right\} \quad (1)$$

siendo $\lfloor x \rfloor$ la parte entera por defecto de x .

Si algún elemento de la afinación es un número irracional, al sistema se le llama *temperamento*, mientras que si todos los elementos son números racionales se llama simplemente *afinación*.

La ventaja de expresar las notas afinadas como potencias de 2, es decir cantidades de la forma 2^{c_n} , es que si nuestra nota de referencia es 2^0 , el exponente c_n mide la "sensación" sonora que provoca esta frecuencia. El motivo es que en la zona media del campo de audición humana la sensación sonora cambia aproximadamente de acuerdo con el logaritmo de la frecuencia. Según esta propiedad la distancia entre dos sonidos cuyas frecuencias son f_1 y f_2 puede estimarse de acuerdo con la expresión:

$$d(f_1, f_2) = 1200 \times \left| \log_2 \left(\frac{f_1}{f_2} \right) \right|. \quad (2)$$

El logaritmo en base 2 y la constante 1200 se utilizan para expresar d en cents. Desde finales del siglo XIX, cuando el uso del sistema temperado de 12 notas ya se había impuesto, A. J. Ellis introdujo la menor unidad con la que habitualmente se miden los intervalos musicales: el cent. Equivale a una centésima de semitono temperado, por tanto en una octava hay 1200 cents. La tabla 1 muestra algunos ejemplos de sistemas de afinación.

Tabla 1: Ejemplos de generadores de algunos sistemas de afinación.

S	Λ	F
T_1	$\lambda_1 = \log_2(3/2)$	$h_1(n) = n$
T_2	$\lambda_1 = 7/12$	$h_1(n) = n$
T_3	$\lambda_1 = \log_2(3/2)$ $\lambda_2 = \log_2(5/4)$	$h_1(n) = n - 4h_2(n)$ $h_2(n) = \left\lfloor \frac{n+1}{7} \right\rfloor + \left\lfloor \frac{n+4}{7} \right\rfloor$
T_4	$\lambda_1 = \log_2(3/2)$ $\lambda_2 = 1/6 \cdot \log_2(2^{12}/3^6)$ $\lambda_3 = 7/12$	$a_n = n - 12 \lfloor n/12 \rfloor$ $h_1(n) = \left\lfloor \frac{a_n+2}{12} \right\rfloor + \left\lfloor \frac{n+3}{12} \right\rfloor + \left\lfloor \frac{a_n+10}{12} \right\rfloor + \left\lfloor \frac{n+11}{12} \right\rfloor$ $h_2(n) = \left\lfloor \frac{a_n+6}{12} \right\rfloor + \left\lfloor \frac{n+7}{12} \right\rfloor + \left\lfloor \frac{a_n+8}{12} \right\rfloor + \left\lfloor \frac{n+9}{12} \right\rfloor$ $h_3(n) = \left\lfloor \frac{a_n+1}{12} \right\rfloor + \left\lfloor \frac{n+4}{12} \right\rfloor + \left\lfloor \frac{a_n+5}{12} \right\rfloor$

T_1 , T_2 , T_3 , y T_4 representan respectivamente los sistemas pitagórico, temperamento igual de 12 notas, la Justa entonación (Zarlino) y el temperamento de Neidhart ($1/2$ & $1/6$ coma).

3. NOTAS MUSICALES BORROSAS Y SISTEMAS DE AFINACIÓN BORROSOS.

Vamos a modelar las notas musicales como números borrosos trapezoidales. $\tilde{A} = (a^L, a^R, \alpha^L, \alpha^R)$. El intervalo

$[a^L, a^R]$ es el núcleo y $[a^L - \alpha^L, a^R + \alpha^R]$ el soporte del número borroso.

Es bien conocido que el oído humano percibe como iguales dos notas cuyas frecuencias están muy próximas. Según los experimentos de N. A. Garbuzov si dos notas no se diferencian en más de 12 cents, se perciben como la misma (unísono). Otros autores reducen a 6 cents esta distancia entre frecuencias. En general podemos definir esta banda del unísono como:

$$]f2^{-\varepsilon}, f2^{\varepsilon}[\equiv f \quad (3)$$

siendo $\varepsilon > 0$, entonces 1200ε expresa en cents, la precisión del oído humano a la percepción del unísono.

La cantidad que llamamos ε determinará la amplitud del núcleo en cambio para la amplitud del soporte debemos tener en cuenta la tolerancia que estamos dispuestos a aceptar para cada nota.

Si la cantidad de notas por octava es q , podemos dividir la octava en q intervalos de $1200/q$ cents. Así que podemos expresar la banda de la nota f como:

$$]f2^{-\delta}, f2^{\delta}[\quad (4)$$

donde $\delta = 1/(2q)$. La cantidad $\Delta = 1200\delta$ expresa, en cents, la tolerancia que admitimos para cada nota. De hecho, esto es lo que hacen los afinadores cromáticos, asignan 12 divisiones por octava $\delta = 1/(2 \times 12)$ y la tolerancia correspondiente a cada nota es $\Delta = 1200 \frac{1}{2 \times 12} = 50$ cents.

La Figura 1 ilustra el concepto de nota musical borrosa.

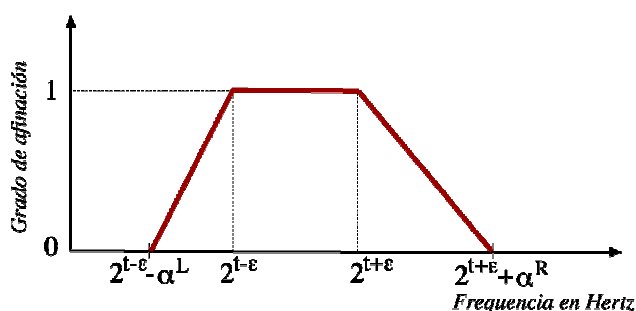


Figura 1: Nota musical borrosa.

Como ya se ha comentado, las notas musicales se suelen expresar como potencias de 2, así pues, parece lógico y más práctico expresar también las notas musicales borrosas de la misma manera.

Vamos a dar una medida de similitud entre dos notas borrosas. Esencialmente es la definición de Zadeh [4] particularizada a nuestro caso.

Definición 2. Sean $2^{\tilde{t}}$ y $2^{\tilde{s}}$ dos notas musicales, donde $\tilde{t} = (t - \varepsilon, t + \varepsilon, \delta - \varepsilon, \delta - \varepsilon)$ y $\tilde{s} = (s - \varepsilon, s + \varepsilon, \delta - \varepsilon, \delta - \varepsilon)$.

Definimos el grado de similitud entre $2^{\tilde{t}}$ y $2^{\tilde{s}}$ como:

$$\text{Sim}[2^{\tilde{t}}, 2^{\tilde{s}}] = \max_x \mu_{\tilde{t} \cap \tilde{s}}(x), \quad (5)$$

y decimos que $2^{\tilde{t}}$ y $2^{\tilde{s}}$ son p -similares, $p \in [0, 1]$, si

$$\text{Sim}[2^{\tilde{t}}, 2^{\tilde{s}}] \geq p.$$

Nótese que esta relación de similitud entre notas borrosas es reflexiva y simétrica pero no transitiva.

La siguiente proposición se demuestra directamente a partir de la expresión (5).

Proposición 1. Consideremos dos notas $2^{\tilde{t}}$ y $2^{\tilde{s}}$, siendo $\tilde{t} = (t - \varepsilon, t + \varepsilon, \delta - \varepsilon, \delta - \varepsilon)$ y

$\tilde{s} = (s - \varepsilon, s + \varepsilon, \delta - \varepsilon, \delta - \varepsilon)$, entonces

a) La similitud entre $2^{\tilde{t}}$ y $2^{\tilde{s}}$ se puede obtener de acuerdo con la fórmula siguiente:

$$\text{sim}[2^{\tilde{t}}, 2^{\tilde{s}}] = \begin{cases} 1 & \text{si } |t-s| < 2\varepsilon \\ 1 - \frac{|t-s| - 2\varepsilon}{2(\delta - \varepsilon)} & \text{si } 2\varepsilon \leq |t-s| \leq 2\delta \\ 0 & \text{si } |t-s| > 2\delta \end{cases}$$

b) Si $p \in [0, 1]$,

$$\text{Sim}[2^{\tilde{t}}, 2^{\tilde{s}}] \geq p \text{ si y sólo si } |t-s| \leq 2\delta - 2p(\delta - \varepsilon).$$

El concepto de similitud entre notas se puede extender a los sistemas de afinación, para esto hemos de definir previamente el concepto de *sistema de afinación borroso*.

Definición 3. Sean $\delta \in [0, 1]$, $\Lambda = \{\lambda_i\}_{i=1}^k \subset [0, 1]$ y

$F = \{h_i : Z \rightarrow Z\}_{i=1}^k$ una familia de funciones. Llamamos sistema de afinación borroso generado por los intervalos $\{\lambda_i\}_{i=1}^k$ y F al conjunto

$$\tilde{S}_\Lambda^F(\delta) = \{2^{\tilde{c}_n} : \tilde{c}_n = (t_n - \varepsilon, t_n + \varepsilon, \delta - \varepsilon, \delta - \varepsilon), n \in Z\}$$

$$\text{donde } t_n = \sum_{i=1}^k \lambda_i h_i(n) - \left\lfloor \sum_{i=1}^k \lambda_i h_i(n) \right\rfloor, n \in Z.$$

Definición 4. Dados $\tilde{S}_q(\delta)$ y $\tilde{T}_q(\delta)$ dos sistemas de afinación con q notas. Decimos que $\tilde{S}_q(\delta)$ y $\tilde{T}_q(\delta)$ son p -similares, si para cada nota $2^{\tilde{s}_i} \in \tilde{S}_q(\delta)$ existe una única $2^{\tilde{t}_i} \in \tilde{T}_q(\delta)$ de manera que

$$\text{Sim}[2^{\tilde{s}_i}, 2^{\tilde{t}_j}] \geq p \quad (6)$$

La cantidad p representa el grado de “intercambiabilidad” entre $\tilde{S}_q(\delta)$ y $\tilde{T}_q(\delta)$ y la unicidad que se exige en la definición garantiza que los dos sistemas tienen una distribución general en el círculo de quintas.

4. EL TEMPERAMENTO IGUAL DE 12 NOTAS BORROSO. CONJUNTOS DE SIMILITUD.

Recordemos que para construir el temperamento igual de 12 notas, la octava se divide en 12 semitonos iguales, por tanto el conjunto de exponentes de 2 que lo generan es:

$$\Lambda = \{n/12 : 1 \leq n \leq 11\}.$$

A pesar de que este temperamento se conocía, al menos, desde el renacimiento, tardó mucho tiempo en imponerse. Fue consagrado por J. S. Bach (1685 - 1750) en su obra *El Clave Bien Temperado* donde realiza 48 Preludios y fugas (en dos libros) en todas las tonalidades. A pesar de su pobreza, debida a que elimina algunas notas naturales que vienen dadas por la escala de armónicos, el temperamento igual de 12 notas es el sistema más empleado por sus ventajas teóricas y prácticas.

El problema real con el que se encuentra el músico es que a pesar que la Teoría de la Música le proporciona las frecuencias exactas de cada nota, él sabe que si un pasaje es rápido y los cambios en las posiciones de los dedos resultan muy complejos debe optar, cuando pueda, por posiciones alternativas que, a pesar de estar menos afinadas consiguen un efecto más estético en el que lo escucha. Veámoslo en un ejemplo.

Ejemplo 1. Consideremos que un saxofón alto en Mib debe interpretar el siguiente fragmento.



notas de $[\tilde{t}]_p$ que tienen menor similitud y se verifica

$$\text{Sim}[2^{\tilde{t}_1}, 2^{\tilde{t}_2}] = 2p - (1 - \varepsilon / \delta)^{-1}.$$

Por comodidad en la notación, llamaremos

$$\kappa = \frac{1}{1 - \varepsilon / \delta}, \quad (8)$$

que es una constante que depende de la tolerancia que estemos dispuestos a admitir y de la sensibilidad del oído humano.

Proposición 3. Sean $2^{\tilde{t}_1}$ y $2^{\tilde{t}_2}$ dos notas de $\tilde{T}_2(\delta)$ de modo que $|t_1 - t_2| = 1/12$, se verifica

$$\text{i) } \text{Sim}[2^{\tilde{t}_1}, 2^{\tilde{t}_2}] \leq \max\{0, \kappa - p\}, \quad \forall 2^{\tilde{t}_1} \in [\tilde{t}_1]_p.$$

$$\text{ii) } \text{Sim}[2^{\tilde{t}_1}, 2^{\tilde{t}_2}] \leq 2(\kappa - p), \quad \forall 2^{\tilde{t}_1} \in [\tilde{t}_1]_p, 2^{\tilde{t}_2} \in [\tilde{t}_2]_p$$

Si $2^{\tilde{t}_1}$ y $2^{\tilde{t}_2}$ son dos notas del sistema temperado de 12 notas borroso tales que $|t_1 - t_2| = n/12$ con $n \geq 2$ entonces la similitud entre la nota de $[\tilde{t}_1]_p$ más similar a $2^{\tilde{t}_2}$ y la nota borrosa $2^{\tilde{t}_2}$ es igual a 0.

Ejemplo 2. Si tomamos $\delta = 1/24$ (la tolerancia del afinador cromático) y $\varepsilon = 3/1200 = 1/400$ (suponemos que el oído humano percibe como iguales las notas que se diferencian en 6 cents) entonces $\kappa = 1.064$.

Si fijamos $p = 0.8$, y para cada nota $2^{\tilde{t}}$ del temperamento igual de 12 notas borroso construimos el conjunto de notas similares a nivel 0.8, $[\tilde{t}]_{0.8}$, por la Proposición 2, sabemos que la similitud entre la nota más grave y la más aguda de cada conjunto es $2p - \kappa = 0.536$. Además, por la Proposición 3 sabemos que si consideramos dos notas que difieren en un semitono, es decir $2^{\tilde{t}_1}$ y $2^{\tilde{t}_2}$ tales que $|t_1 - t_2| = 1/12$, entonces, la similitud más alta que puede darse entre una nota de $[\tilde{t}_1]_{0.8}$ y la nota $2^{\tilde{t}_2}$ es de $\kappa - p = 0.264$. Y la similitud más alta que puede darse entre una nota de $[\tilde{t}_1]_{0.8}$ y otra de $[\tilde{t}_2]_{0.8}$ es de $2(\kappa - p) = 0.528$.

5. CONCLUSIONES

Gracias a la lógica borrosa podemos dar una explicación teórica de la práctica diaria de los músicos. Según el modelo presentado, las notas que suenan bien ya no tienen por qué considerarse como desafinadas aunque la frecuencia que detectaría un afinador cromático no sea la teórica.

El estudio que hemos realizado sobre el temperamento igual de 12 notas sugiere que todas las notas de un

conjunto de similitud $[\tilde{t}]_p$ a un nivel $p > 0.5$ suficientemente alto serían candidatas a sustituir a la nota $2^{\tilde{t}}$. Incluso podría justificar el hecho de que a veces en los pasajes rápidos los músicos toquen alguna nota con una posición alternativa (aunque suene menos afinada) si esto les resulta más cómodo, sin que se tenga la sensación de que desafinan.

Como en muchas otras actividades humanas lo que hacen los músicos es aplicar reglas de decisión borrosas.

Agradecimientos

Los autores agradecen la ayuda prestada por Jorge Sanz Liern (estudiante de saxofón) y la financiación proporcionada por los proyectos de investigación TIN2008-06872-C04-02 y TIN2006-10134 del Ministerio de Ciencia e Innovación del gobierno español.

Referencias

- [1] A. del Corral, T. León, V. Liern, Compatibility of the Different Tuning Systems in an Orchestra, Mathematics and Computation in Music, Springer Verlag, Pág. 93-103, Berlin, 2009.
- [2] J. J. Goldáraz Gaínza, Afinación y temperamento en la música occidental, Alianza Editorial, Madrid, 1992.
- [3] J. Haluska, Equal temperament and pythagorean tuning: a geometrical interpretation in the plane, Fuzzy Sets and Systems 114, Pag 261-269, 2000.
- [4] J. Haluska, The Mathematical Theory of Tone Systems, Marcel Dekker, Inc., Bratislava, 2005.
- [5] V. Liern. Fuzzy tuning systems: the mathematics of the musicians, Fuzzy Sets and Systems 150. Pág. 35-52, 2005.
- [6] L. A. Zadeh. Similarity relations and fuzzy orderings, Inform. Sci. 3. Pág. 177-200, 1971.

UNA PRIMERA APROXIMACIÓN AL PROBLEMA DE ASIGNACIÓN DE ATRAQUES CON TIEMPOS DE LLEGADA DIFUSOS

Belén Melián-Batista¹ J. Marcos Moreno-Vega¹ José Luis Verdegay Galdeano²

¹ Dpto. de Estadística, IO y Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de La Laguna, {mbmelian, jmmoreno}@ull.es

² Dpto. de Ciencias de la Computación e Inteligencia Artificial
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada, verdegay@decsai.ugr.es

Resumen

En un problema de asignación de atraques, dado un conjunto de barcos, se desea determinar cómo asignar éstos a localizaciones en el muelle (atraques) de forma que se optimice alguna función de coste. La función de coste del problema considerado es la suma de los tiempos de espera y de trabajo de cada barco. En las aplicaciones reales existe un cierto grado de incertidumbre con respecto al tiempo de llegada de los barcos al puerto. Esta incertidumbre puede modelizarse adecuadamente considerando que estos tiempos son números difusos. En este trabajo proponemos un modelo difuso para la asignación de atraques y diseñamos un método de solución apropiado para el mismo.

Palabras Clave: Asignación de atraques, modelos difusos

un puerto como base de operaciones son la ubicación geográfica, la estabilidad política y social y los costes de las operaciones.

Los puertos españoles cuentan con una envidiable situación geográfica para el tráfico marítimo de contenedores Asia-Occidente, que utiliza el Estrecho como base de operaciones, y para el tráfico América-Europa-África, que emplea los puertos canarios. Sin embargo, algunos puertos africanos, como el de Tánger, llevan a cabo una fuerte competencia por estos mercados. La respuesta debe ser una mejora en la gestión logística de las terminales portuarias de contenedores por su directa incidencia en los precios y en la calidad del servicio.

Para la gestión eficiente de una terminal portuaria de contenedores pueden adoptarse decisiones a tres niveles: estratégico, táctico y operacional. Al nivel estratégico pertenecen decisiones sobre qué equipamiento adquirir y cómo operar; al nivel táctico, qué estrategia seguir para almacenar los contenedores; y al nivel operacional, dónde colocar un contenedor para su posterior carga en el transporte que lo repartirá fuera del puerto. El horizonte temporal de las decisiones varía de un nivel a otro. Así, las decisiones estratégicas cubren de uno a varios años, las decisiones tácticas de unos días a varios meses y las decisiones operacionales se adoptan diariamente. Evidentemente, los tres niveles se ven influenciados unos por otros: la decisión estratégica sobre qué infraestructura adquirir condiciona la forma en que pueden almacenarse los contenedores, y esto último influye en los tiempos de carga en los medios de transporte terrestres. Del mismo modo, la solución dada a estos últimos problemas repercute en el comportamiento global de la terminal y debe tenerse en cuenta, por tanto, a la hora de adoptar decisiones estratégicas.

Para enfrentarse a la compleja tarea de gestionar a nivel estratégico y táctico una terminal de contenedores, los gestores requieren herramientas con las que

1 Introducción

Según datos de la Eno Transportation Foundation, alrededor del 60% de las mercancías que viajan por mar lo hacen en contenedores. Este porcentaje se acerca al 100% en algunas rutas que conectan países estables y económicamente fuertes. La carga anual transportada asciende a unos 100 millones de TEUs (unidad de medida equivalente a las dimensiones de un contenedor estándar). Los puertos compiten por ser puntos de intercambio (hubs) o puntos origen-destino de las rutas empleadas para el transporte. Los gobiernos regionales y nacionales consideran un objetivo estratégico contar con puertos en los que ubicar terminales de contenedores, por ser focos de crecimiento económico. La competencia en este campo es feroz. Los principales criterios usados por las grandes operadoras para escoger

simular el comportamiento de su terminal y adoptar proactivamente las decisiones adecuadas. Existe, por tanto, la necesidad de disponer de Sistemas de Ayuda a la Decisión que, incorporando información sobre las características de la terminal y del equipamiento disponible y conocimiento experto sobre la forma en que se resuelven los problemas, asistan a los gestores. Por otra parte, se necesitan también procedimientos que encuentren soluciones apropiadas para los problemas operacionales que deben resolverse diariamente (descarga de barcos, asignación de espacio de almacenamiento para los contenedores, determinación de rutas para los vehículos que transportan los contenedores dentro y fuera del puerto ...). En definitiva, se requiere de un sistema inteligente para la gestión integral de la terminal de contenedores.

A grandes rasgos, el proceso de carga y descarga en una terminal de contenedores consta de varias etapas (ver, por ejemplo, [8], [3], [9]). Al llegar el barco al puerto, hay que asignarle un muelle de descarga, descargar los contenedores y, usando vehículos adecuados, transportarlos hasta el lugar en que se almacenan apilados durante un cierto tiempo. Finalmente, son transferidos, mediante vehículos apropiados, a camiones, trenes o barcos para su reparto fuera del puerto.

En cada una de las anteriores etapas surgen complejos problemas de planificación y gestión de recursos que, si se resuelven adecuadamente, suponen un mayor nivel de satisfacción y un ahorro económico importante para las operadoras del transporte marítimo de contenedores y para las autoridades portuarias. Uno de los recursos que mayor impacto tiene en los costes es el atraque.

Aunque es evidente que existe cierta incertidumbre asociada a muchos de los datos de entrada que definen un problema (tiempo de llegada del barco, tiempo de carga o descarga, distancias recorridas ...), no hemos encontrado en la literatura especializada modelos para la gestión de una terminal de contenedores que contemplen esta incertidumbre. En este trabajo proponemos un modelo difuso para la asignación de atraques y diseñamos un método de solución apropiado para el mismo.

2 Asignación de atraques

En un Problema de Asignación de Atraques (*Berth Allocation Problem*), dado un conjunto de barcos, se desea determinar cómo asignar éstos a localizaciones en el muelle (atraques) de forma que se optimice alguna función de costo. Las características de los barcos, del puerto o de los recursos del mismo determinan qué restricciones deben cumplirse para obtener

una solución factible. Asimismo, pueden considerarse diferentes funciones de costo para medir la eficiencia de la terminal.

En muchas terminales multi-usuarios (*Multi-User Terminals*, (MUT)) el muelle se particiona en un conjunto finito de atraques a los que se pueden asignar los barcos para las operaciones de estiba y desestiba. Se denomina problema discreto de asignación de atraque (*Discrete Berth Allocation Problem*, (BAPD)) al problema asociado a este esquema de asignación.

En la versión estática del problema (*Static Discrete Berth Allocation Problem*, (SBAPD)) todos los barcos se encuentran en la terminal antes de que se inicie la planificación, mientras que la versión dinámica (*Dynamic Discrete Berth Allocation Problem*, (DBAPD)) tiene en cuenta también a aquellos barcos que llegarán en algún momento del horizonte de planificación. El problema estático de asignación de atraques fue propuesto por Imai et al. [5] quienes probaron que se reduce a un problema de asignación bidimensional que puede resolverse fácilmente por el método Hungaro. El problema dinámico ha sido estudiado en [4], [6] y [7]. Imai et al. [6] propusieron una heurística basada en la relajación Lagrangiana del problema original y desarrollaron una amplia experiencia computacional que muestra que el procedimiento propuesto es útil en aplicaciones reales. En [7] se estudia el problema continuo en el que se considera que un barco puede atracar en cualquier lugar del puerto. Los autores proponen un Algoritmo Genético para este problema y muestran experimentalmente la validez del mismo. Recientemente, Hansen et al. [4] han propuesto un VNS para una variante de este problema en el que consideran que, asociado a cada barco, existe un premio y una penalización por acabar antes o después, respectivamente, del tiempo previamente convenido.

2.1 El problema discreto estático de asignación de atraques

Sean

$V = \{1, \dots, T\}$,	conjunto de barcos
$B = \{1, \dots, I\}$,	conjunto de atraques
$O = \{1, \dots, T\}$,	conjunto de órdenes de servicio
S_i ,	tiempo en el que está disponible el atraque i
A_j ,	tiempo estimado de llegada del barco j
C_{ij} ,	tiempo de trabajo en el barco j en el atraque i

y

$$x_{ijk} = \begin{cases} 1 & \text{si el barco } j \text{ es servido como el} \\ & k\text{-ésimo barco en el atraque } i \\ 0 & \text{en otro caso} \end{cases}$$

Los conjuntos V y O tienen el mismo número de elementos T porque todos los barcos podrían ser atendidos en el mismo atraque.

El problema se formula como

$$\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in O} \{(T - k + 1)C_{ij} + S_i - A_j\}x_{ijk}, \quad (1)$$

sujeto a

$$\sum_{i \in B} \sum_{k \in O} x_{ijk} = 1 \quad \forall j \in V, \quad (2)$$

$$\sum_{j \in V} x_{ijk} \leq 1 \quad \forall i \in B, k \in O, \quad (3)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in B, j \in V, k \in O \quad (4)$$

La función objetivo (1) es la suma de los tiempos de espera y de trabajo de cada barco. Nótese que, dado que se trata de la versión estática del problema, $S_i \geq A_j, \forall i \in B, j \in V$. La restricción (2) asegura que cada barco es atendido en algún atraque en algún orden, y la restricción (3) que en cada atraque y en cada orden como máximo puede ser atendido un barco. En la función objetivo (1), el tiempo de trabajo C_{ij} se pondera con el término $(T - k + 1)$ ya que el tiempo de espera de un barco incluye el tiempo de trabajo en los barcos que han sido atendidos antes que él en el mismo atraque.

2.2 El problema discreto dinámico de asignación de atraques

A diferencia del modelo estático, el modelo dinámico tiene en cuenta a los barcos que llegarán durante el horizonte de planificación.

El problema se formula como

$$\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in O} \{(T - k + 1)C_{ij} + S_i - A_j\}x_{ijk} + \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in O} \{(T - k + 1)y_{ijk}, \quad (5)$$

sujeto a

$$\sum_{i \in B} \sum_{k \in O} x_{ijk} = 1, \quad \forall j \in V, \quad (6)$$

$$\sum_{j \in V} x_{ijk} \leq 1, \quad \forall i \in B, k \in O, \quad (7)$$

$$\sum_{l \in V} \sum_{m \in P_k} (C_{il}x_{ilm} + y_{ilm}) + y_{ijk}$$

$$-(A_j - S_i)x_{ijk} \geq 0, \quad \forall i \in B, j \in W_i, k \in O, \quad (8)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i \in B, j \in V, k \in O, \quad (9)$$

$$y_{ijk} \geq 0, \quad \forall i \in B, j \in V, k \in O, \quad (10)$$

donde

$$P_k = \{p \in O/p < k\}$$

$$W_i = \{j \in V/S_i \leq A_j\}$$

y

y_{ijk} = tiempo ocioso del atraque i cuando el barco j es atendido como el k -ésimo barco en ese atraque

La función objetivo (5) es el tiempo total de espera y atraque de todos los barcos, incluidos aquellos que llegarán durante el horizonte de planificación. Las restricciones (8) aseguran que los barcos son atendidos después de su llegada al puerto.

3 Asignación de atraques con tiempos de llegada difusos

En las aplicaciones reales existe un cierto grado de incertidumbre con respecto al tiempo de llegada de los barcos al puerto. Esta incertidumbre puede modelizarse adecuadamente considerando que estos tiempos son números difusos. Se obtiene así el siguiente modelo difuso

$$\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in O} \{(T - k + 1)C_{ij} + S_i - \tilde{A}_j\}x_{ijk} + \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in O} \{(T - k + 1)y_{ijk},$$

sujeto a

$$(6), (7), (9), (10),$$

$$\sum_{l \in V} \sum_{m \in P_k} (C_{il}x_{ilm} + y_{ilm}) + y_{ijk}$$

$$-(\tilde{A}_j - S_i)x_{ijk} \geq 0, \quad \forall i \in B, j \in W_i, k \in O, \quad (11)$$

donde

$$P_k = \{p \in O/p < k\}$$

$$W_i = \{j \in V/S_i \leq \tilde{A}_j\}$$

Se trata de un problema de programación lineal entera mixta en el que tanto los costes en la función objetivo como los parámetros de la matriz tecnológica de las restricciones son difusos. A diferencia de otros problemas que pueden encontrarse en la literatura, los mismos parámetros que son difusos en la función objetivo, lo son difusos en las restricciones. Además, el número de restricciones (11) también depende de los valores difusos $\tilde{A}_j$. Por otro lado, aún sabiendo que

no son comparables números difusos y reales, sólo por propósitos expositivos, aquí hemos escrito estas comparaciones mezclando ambos dominios. Más adelante, para resolver el modelo que se propone, se matizará esa comparación y se modelizará de manera rigurosa.

3.1 Modelo difuso

De forma general, el modelo anterior se establece como

$$\begin{aligned} \min \tilde{c}x \\ \text{s.a. :} \\ \tilde{A}x \leq \tilde{b}, \\ x \geq 0, \end{aligned}$$

donde $\tilde{A}$ es una matriz $n \times m$ de números difusos y $\tilde{c}$ y $\tilde{b}$ son vectores de números difusos de dimensiones respectivas n y m .

Para la resolución del anterior modelo analizaremos métodos de solución para el problema con función objetivo con coeficientes difusos y para el problema con coeficientes difusos en las restricciones.

3.1.1 Coeficientes difusos en la función objetivo

Cuando se considera un modelo de programación lineal entera mixta con costes imprecisos como en el caso real presentado en este trabajo, la función objetivo resultante es una función difusa. El modelo con costes difusos se establece entonces de la siguiente manera.

$$\begin{aligned} \min \tilde{c}x \\ \text{s.a. :} \\ Ax \leq b, \\ x \geq 0, \end{aligned} \quad (12)$$

donde A es una matriz $n \times m$ de números reales, $b \in \mathbb{R}^n$ y $\tilde{c}$ es un vector n -dimensional con números difusos caracterizados por funciones de pertenencia expresadas de la siguiente manera:

$$\mu_j(x) = \begin{cases} 0, & x \geq c_j^U, x \leq c_j^L \\ h_j(x), & c_j^L \leq x \leq c_j \\ g_j(x), & c_j \leq x \leq c_j^U \end{cases}$$

donde $h_j(\cdot)$ y $g_j(\cdot)$ son funciones continuas estrictamente crecientes y decrecientes, respectivamente, y $h_j(c_j) = g_j(c_j) = 1, j \in \mathbb{N}$. En la literatura han sido propuestos diferentes enfoques para resolver este problema con costes difusos [2].

En este trabajo consideramos, por simplicidad y sin pérdida de generalidad, funciones de pertenencia triangulares para los costes. Si consideramos los $(1 - \alpha)$ -cortes de los costes, $\alpha \in [0, 1]$, entonces

$$\forall x \in \mathbb{R}, \mu_j(x) \geq 1 - \alpha \Leftrightarrow h_j^{-1}(1 - \alpha) \leq x \leq g_j^{-1}(1 - \alpha),$$

y, si denotamos,

$$\phi_j(\cdot) = h_j^{-1}(\cdot), \psi_j(\cdot) = g_j^{-1}(\cdot), j \in \mathbb{N},$$

obtenemos que

$$\phi_j(1 - \alpha) \leq x \leq \psi_j(1 - \alpha), \forall x \in \mathbb{R}, j \in \mathbb{N}.$$

Según Bitran [1], es posible obtener una solución difusa al modelo (12) a partir de la solución paramétrica del siguiente modelo de programación lineal paramétrico multiobjetivo.

$$\begin{aligned} \min [c^1 x, c^2 x, \dots, c^{2^n} x] \\ \text{s.a. :} \\ Ax \leq b, \\ x \geq 0 \\ c^k \in \mathbb{IE}(1 - \alpha), \alpha \in [0, 1], k = 1, 2, \dots, 2^n \end{aligned} \quad (13)$$

donde, para cada $\alpha \in [0, 1]$, $\mathbb{IE}(1 - \alpha)$ es el conjunto de vectores en $\mathbb{R}^n$ tal que cada uno de sus componentes está en la cota superior, $\psi_j(1 - \alpha)$, o en la cota inferior, $\phi_j(1 - \alpha)$, del $(1 - \alpha)$ -corte correspondiente. Por tanto, $\forall k = 1, 2, \dots, 2^n$,

$$c^k = (c_1^k, c_2^k, \dots, c_n^k) \in \mathbb{IE}(1 - \alpha).$$

Con el objetivo de obtener una solución paramétrica al modelo (13), asignamos pesos a los objetivos. Por simplicidad y sin pérdida de generalidad, consideramos los pesos iguales para todos los objetivos, esto es $w_k = 1/2^n, k = 1, 2, \dots, 2^n$. Entonces, una solución al problema con costes difusos se obtiene de la solución del siguiente modelo matemático.

$$\begin{aligned} \min w_1 c^1 x + w_2 c^2 x + \dots + w_{2^n} c^{2^n} x \\ \text{s.a. :} \\ Ax \leq b, \\ x \geq 0 \\ c^k \in \mathbb{IE}(1 - \alpha), \alpha \in [0, 1], k = 1, 2, \dots, 2^n \\ \sum w_k = 1, w_k \geq 0 \end{aligned}$$

3.1.2 Coeficientes difusos en las restricciones

En el problema considerado de asignación de atraques a barcos, los tiempos de llegada de los barcos aparecen también como coeficientes de las restricciones. La modelización de la imprecisión en los valores de los

parámetros puede enfocarse considerando que éstos son números difusos. En este sentido, independientemente de la amplia gama de modelos diferentes que pueden ser considerados, el problema mencionado podría abordarse mediante el siguiente modelo:

$$\begin{aligned} & \min \tilde{c}x \\ & \text{s.a. :} \\ & \quad \tilde{A}x \leq_I \tilde{b}, \\ & \quad x \geq 0, \end{aligned}$$

donde $\tilde{A}$ y $\tilde{b}$ hacen referencia al hecho de que estamos teniendo en cuenta números difusos en los coeficientes que definen las restricciones, y el símbolo $\leq_I$ significa que la manera de comparar los dos miembros de la desigualdad, debido a la coherencia formal, debe hacerse mediante una relación de ordenación de números difusos. Esta relación de comparación $\leq_I$ puede ser cualquiera de la extensa lista disponible [10], que a su vez permitiría al decisor tener un mayor grado de libertad a la hora de establecer preferencias. En términos más específicos, a fin de disponer de un modelo teórico con una forma de funcionamiento, vamos a referirnos brevemente a los distintos índices para comparar los números difusos que se han descrito en la literatura [10]. Entre los diferentes enfoques descritos para realizar la comparación, por simplicidad, en este artículo sólo consideramos el que se deriva del uso de índices de comparación. Por lo tanto, denotando como $F(R)$ al conjunto de números difusos, si

$$I : F(R) \rightarrow [0, 1]$$

es un índice de comparación para esta clase de números, entonces

$$\forall \tilde{P}, \tilde{Q} \in F(R), \tilde{P} \leq_I \tilde{Q} \Leftrightarrow I(\tilde{P}) \leq I(\tilde{Q})$$

Según el índice usado I , pueden obtenerse diferentes modelos auxiliares para resolver eficazmente el problema descrito anteriormente desde el punto de vista práctico. Por lo tanto, en general, el modelo auxiliar que se utiliza para resolver el problema descrito desde el punto de vista práctico, sería de la siguiente manera:

$$\begin{aligned} & \min \tilde{c}x \\ & \text{s.a. :} \\ & \quad I(\tilde{A}x) \leq I(\tilde{b}), \\ & \quad x \geq 0, \end{aligned}$$

Usar un índice I u otro depende de cada decisor, y por tanto qué índice I elegir para ser usado no es estudio de este trabajo. Con el objetivo de ilustrar el

enfoque propuesto, y como un caso trivial, consideremos dos números difusos $\tilde{P}, \tilde{Q} \in F(R)$, denotados como $\tilde{P} = (P, P_i, P_d)$ y $\tilde{Q} = (Q, Q_i, Q_d)$, y como forma de comparación, el dado por el primer índice de Yager [10],

$$\tilde{P} \leq_I \tilde{Q} \Leftrightarrow (1/3)(P + P_i + P_d) \leq (1/3)(Q + Q_i + Q_d)$$

Entonces, el modelo anterior adquiere la siguiente forma,

$$\begin{aligned} & \min \tilde{c}x \\ & \text{s.a. :} \\ & \quad (A + A_i + A_d)x \leq (b + b_i + b_d), \\ & \quad x \geq 0, \end{aligned}$$

De este modelo obtenemos una solución al problema considerado.

Dado que en el modelo descrito para la resolución del problema de asignación de barcos a ataques, los tiempos de llegada de los barcos no pueden ser estimados exactamente y aparecen en los coeficientes de las restricciones, la metodología explicada en esta sección puede ser usada para proporcionar soluciones al problema. Cuando se aplica esta metodología con el primer índice de Yager como un ejemplo simple, las restricciones difusas originales se convierten en restricciones de un modelo lineal entero mixto.

3.1.3 Aplicación al problema de asignación de ataques

Supongamos que cada número difuso $\tilde{A}_j$, que representa el tiempo de llegada del barco j y que aparece tanto en la función objetivo como en la matriz tecnológica de las restricciones, viene dado por el siguiente número triangular.

$$\mu(A_j) = \begin{cases} 0, & A_j \geq u_j, A_j \leq l_j \\ \frac{1}{m_j - l_j}(A_j - l_j), & l_j \leq A_j \leq m_j \\ \frac{1}{u_j - m_j}(-A_j + u_j), & m_j \leq A_j \leq u_j \end{cases}$$

Dado $(1 - \alpha)$, $\alpha \in [0, 1]$, obtenemos las siguientes cotas inferiores y superiores para cada uno de los costes difusos de la función objetivo, $A_j, j \in [1, \dots, T]$.

$$\phi_{\tilde{A}_j}(1 - \alpha) = m_j - (m_j - l_j)\alpha.$$

$$\psi_{\tilde{A}_j}(1 - \alpha) = m_j + (u_j - m_j)\alpha.$$

Teniendo en cuenta lo explicado en las secciones (3.1.1) y (3.1.2) anteriores, dado un valor $\alpha \in [0, 1]$, una

solución al problema de atraque de barcos con tiempos de llegada difusos se obtiene del siguiente modelo de programación lineal entera mixta:

$$\begin{aligned} \min & \left(\sum_{i \in B} \sum_{j \in V} \sum_{k \in O} \{(T - k - 1)C_{ij} + S_i\}x_{ijk} + \right. \\ & \left. \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in O} (T - k + 1)y_{ijk} + \right. \\ & \left. \sum_{i \in B} \sum_{j \in V} \sum_{k \in O} \left[\frac{1}{2}\phi_{\tilde{A}_j}(1 - \alpha) + \frac{1}{2}\psi_{\tilde{A}_j}(1 - \alpha) \right]x_{ijk} \right) \end{aligned}$$

sujeto a

$$\begin{aligned} \sum_{i \in B} \sum_{k \in O} x_{ijk} &= 1, \quad \forall j \in V, \\ \sum_{j \in V} x_{ijk} &\leq 1, \quad \forall i \in B, k \in O, \\ \sum_{l \in V} \sum_{m \in P_k} (C_{il}x_{ilm} + y_{ilm}) + y_{ijk} \\ &- \left(\frac{1}{3}(l_j + m_j + u_j) - S_i \right) x_{ijk} \geq 0, \\ &\quad \forall i \in B, j \in W_i, k \in O, \\ x_{ijk} &\in \{0, 1\}, \quad \forall i \in B, j \in V, k \in O, \\ y_{ijk} &\geq 0, \quad \forall i \in B, j \in V, k \in O, \end{aligned}$$

donde

$$\begin{aligned} P_k &= \{p \in O/p < k\} \\ W_i &= \{j \in V/S_i \leq \frac{1}{3}(l_j + m_j + u_j)\}. \end{aligned}$$

4 Conclusiones y trabajos futuros

Para gestionar adecuadamente una terminal portuaria de contenedores, los gestores necesitan herramientas con las que simular el comportamiento de su terminal para adoptar proactivamente las decisiones adecuadas. Los atraques son unos de los recursos que mayor impacto tienen en los costes de gestión de la terminal. En el presente trabajo hemos propuesto un modelo difuso para la asignación de atraques cuando los tiempos de llegada de los barcos vienen dados por números difusos. Además, hemos diseñado un método de solución operativo apropiado para tratar este modelo, que, si bien proporciona soluciones de forma satisfactoria, no hemos presentado aquí por cuestiones de espacio. Pero será motivo de una inmediata próxima publicación. Desde esa perspectiva, actualmente trabajamos en el diseño e implementación de un Sistema de Ayuda a la Decisión Inteligente (SADI) que, incorporando el modelo de programación lineal entera mixta anterior, asista a los gestores en sus tareas.

Agradecimientos

El presente trabajo ha sido parcialmente financiado por el Fondo Europeo de Desarrollo Regional (FEDER), el Ministerio de Ciencia y Tecnología (proyectos TIN2008-06872-C04-01 y TIN2008-06872-C04-04), el Gobierno de Canarias (proyecto PI2007/019) y la Junta de Andalucía (TIC-02970).

Referencias

- [1] G.R. Bitran. Linear multiple objective problems with interval coefficients. *Management Science*, 26(7):694–706, 1980.
- [2] M. Delgado, J.L. Verdegay, and M.A. Vila. Relating different approaches to solve linear programming problems with imprecise costs. *Fuzzy Sets and Systems*, (37):33–42, 1990.
- [3] Robert Stahlbock Dirk Steenken, Stefab Voß. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26(1):3–49, 2004.
- [4] Pierre Hansen, Ceyda Oğuz, and Nenad Mladenović. Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, 191(3):636–649, 2008.
- [5] Akio Imai, Ken Nagaiwa, and Weng Tat Chan. Efficient planning of berth allocation for container terminals in asia. *Journal of Advanced Transportation*, 31(1):75–94, 1997.
- [6] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4):401–407, 2001.
- [7] Etsuko Nishimura, Akio Imai, and Stratos Papadimitriou. Berth allocation planning in the public berth system by a genetic algorithms. *European Journal of Operational Research*, 131(2):282–292, 2001.
- [8] Robert Stahlbock and Stefan Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, 2008.
- [9] Iris F. A. Vis and Ren de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1):1–16, 2003.
- [10] X. Wang and E. Kerre. On the classification and the dependencies of the ordering methods. In D. Ruan, editor, *Fuzzy Logic Foundation and Industrial Applications. International Series in Intelligent Technologies*, pages 73–90. Kluwer, 1996.

ALGORITMO GENÉTICO CON EVALUACIÓN BORROSA MULTICRITERIO: APLICACIÓN A LA PLANIFICACIÓN DE MOVIMIENTOS EN MANIPULADORES PARALELOS

D. A. López García.¹ F. Gómez-Bravo.² J. M. Martín Ramos.³ I. López Quintero.⁴ G. Carbone⁵

^{1,2} D. Ing. Electrónica, Sist. Informáticos y Autm., U. Huelva, {diego.lopez, fernando.gomez} @diesia.uhu.es,

³ D. Tecnologías de la Información, jmmartin@dti.uhu.es

⁴ Escuela Politécnica Superior, Universidad de Huelva, ismael.lopez@alu.uhu.es

⁵ Laboratorio de Robótica y Mecatrónica, Universidad de Cassino, carbone@unicas.it

Resumen

La planificación de movimiento en robots manipuladores presenta cierta dificultad cuando existen obstáculos. Esta dificultad es mayor si se trata de manipuladores paralelos. En este artículo se presenta un método de planificación basado en la utilización de un algoritmo genético que aplica una técnicas de selección Multicriterio Borrosa. Su utilidad se ha verificado al aplicarla al manipulador paralelo desarrollado en el Laboratorio de Robótica y Mecatrónica de la Universidad de Cassino.

Palabras Clave: Robótica, Manipuladores, Planificación de Trayectorias, Algoritmos Genéticos, Decisión Multicriterio Borrosa.

1 INTRODUCCIÓN

La planificación de movimientos en sistemas robóticos es un problema frecuentemente abordado en el ámbito científico [2, 4, 5, 7, 9], que cobra particular relevancia en el caso de los robots manipuladores. En estos sistemas, la posición queda descrita por un conjunto de parámetros de configuración determinados por los ángulos de sus articulaciones [9]. En consecuencia, es habitual que la manipulación de objetos se base en la planificación de trayectorias en el espacio articular [2]. Dichas trayectorias suelen expresarse como secuencias de configuraciones muy próximas entre sí. Si además, en el espacio cartesiano existen obstáculos, será necesario que dichas trayectorias los eviten, de forma que para cada configuración ninguno de los elementos del robot colisione con los mismos. Un algoritmo capaz de obtener este tipo de trayectorias es el conocido como Rapidly-exploring Random Trees (RRT) [4,5,7]. Su inconveniente radica en la escasa calidad de sus resultados. Otros métodos alternativos, se basan en la aplicación de algoritmos genéticos sobre poblaciones generadas al azar [2,3]. El problema en este caso reside en el elevado número de generaciones necesarias para obtener

trayectorias libres de colisión. De cualquier modo, existe cierta dificultad para establecer un procedimiento que pondere la bondad de unas soluciones sobre otras. Con este fin, en este artículo se ha utilizado un método de clasificación de trayectorias inspirado en los Métodos de Decisión Multicriterio [1], que permite incorporar criterios de preferencias humanas mediante la definición de un conjunto pesos y umbrales de preferencias borrosos.

Este trabajo presenta una técnica que utiliza el RRT como generador de una población inicial, a partir de la cual un algoritmo genético proporciona trayectorias óptimas en tiempos admisibles. Este algoritmo aplica el método de Decisión Multicriterio Borroso FPromethee II [10], para realizar la selección de individuos.

Se ha probado dicha técnica en manipuladores de 2, 3 y 4 parámetros de configuración, entre los que se encuentra el manipulador paralelo CaPaMan (Cassino Parallel Manipulator) desarrollado en el Laboratorio de Robótica y Mecatrónica de la Universidad de Cassino [2].

Una estructura paralela se define como aquella en la cual el efector final esta unido a la base por más de una cadena cinemática cerrada independiente [2,9]. Estos manipuladores pueden trabajar con regímenes de velocidad alta, presentando gran robustez y precisión en el posicionamiento. Sin embargo, su reducido espacio de trabajo hace especialmente difícil la maniobrabilidad de los mismos. CaPaMan, es un manipulador paralelo con tres grados de libertad, compuesto por una base fija y una plataforma móvil que están conectadas por tres brazos, como se puede ver en las imágenes de la Fig. 1 a) y b). Cada mecanismo de la plataforma consta de un paralelogramo articulado que lleva una barra acopladora que se une a una varilla vertical mediante un par prismático. La varilla vertical se conecta a la plataforma móvil a través de una junta esférica. La representación geométrica de CaPaMan utilizada en este trabajo para la detección de colisiones puede observarse en la Fig. 1 (b). En ella se ha añadido sobre la plataforma un cilindro y un prisma representando una herramienta. Los tres parámetros de configuración son los tres ángulos α_1 ,

α_2 y α_3 que forman cada paralelogramo articulado con el plano horizontal (ver Fig. 1 b). Cada terna de valores determina la orientación de la plataforma móvil y la posición del extremo de la herramienta

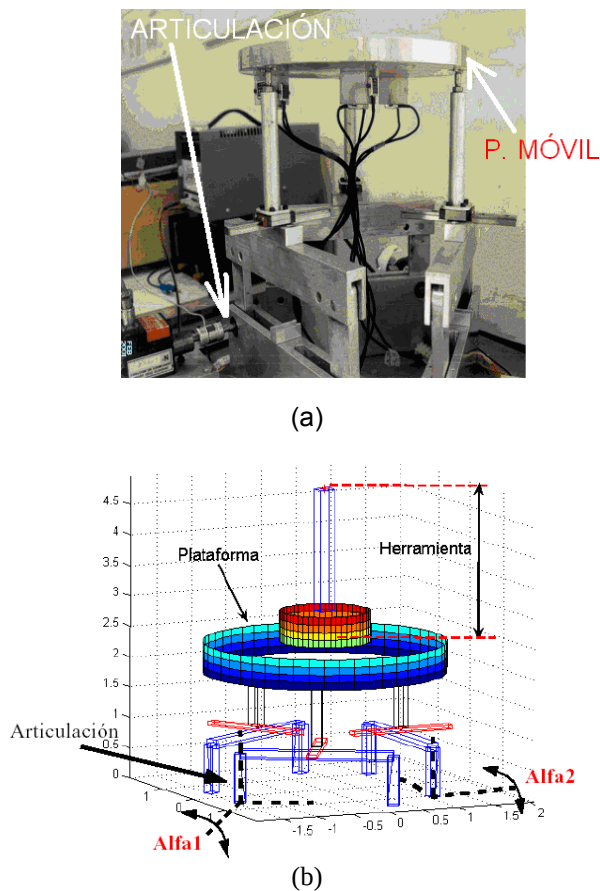


Figura 1: Manipulador Paraleo CAPAMAN

2 POBLACIÓN INICIAL

Durante el movimiento de un brazo robótico, la existencia de obstáculos puede provocar colisiones con diferentes partes del mismo. Por tanto, además de considerar el efector final es necesaria la computación de las trayectorias de todos los puntos del robot para asegurar la viabilidad de la solución. Un procedimiento que simplifica este contratiempo consiste en la definición de un espacio articular biunívocamente relacionado con el cartesiano. Así, para cada combinación de parámetros de configuración, las distintas partes del robot ocupan un lugar en el espacio cartesiano. La construcción del espacio de configuración libre de colisión, se obtiene discretizando el espacio articular y comprobando para cada punto de este si existe colisión. La detección de colisiones se realiza aplicando las ecuaciones de la cinemática directa del manipulador [6, 9]. Así, se genera un mapa en el espacio articular con lugares permitidos y prohibidos, donde se planteará hallar las trayectorias (ver

Fig. 2, 3 y 4). De este modo, se aumenta hasta 30 veces la velocidad de cómputo respecto de la comprobación iterativa de colisiones en el espacio cartesiano. Los Autores se encuentran trabajando en una técnica que, a partir de la detección de obstáculos en el espacio cartesiano, permite una rápida actualización del mapa articular, lo que habilita al método para su utilización en aplicaciones con escenarios dinámicos.

La planificación de trayectorias utilizando un algoritmo genético y una población inicial generada al azar, tiene el inconveniente de elevar el coste computacional (son necesarias poblaciones iniciales superiores a los 500 individuos para encontrar una solución válida). La baja calidad de la población inicial, demasiado alejada de la solución buscada, aumenta el tiempo de convergencia.

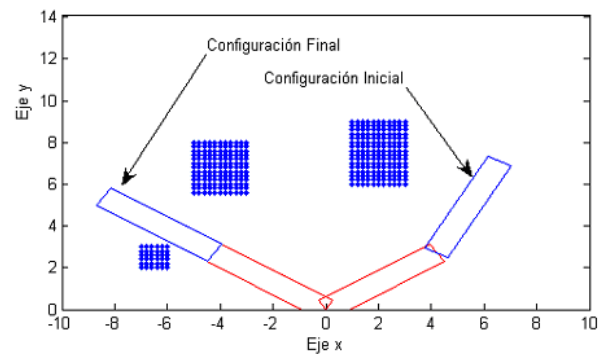


Figura 2: Brazo robótico en dos posiciones

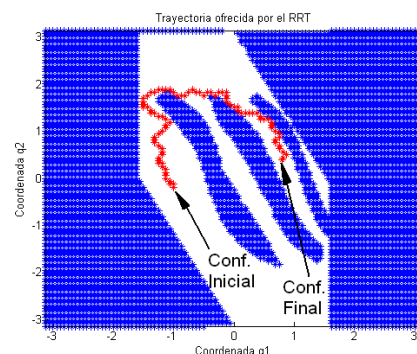


Figura 3: Espacio articular. Solución del RRT.

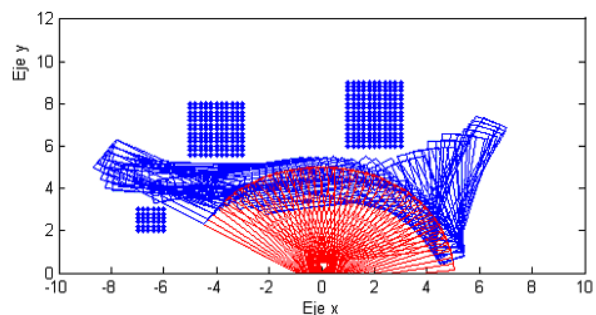


Figura 4: Solución en el espacio cartesiano.

Por el contrario, en este artículo, para la generación de la población inicial se utiliza el algoritmo RRT. Se trata de

una técnica que hace crecer, utilizando métodos aleatorios, dos estructuras en forma de árbol, de tal forma que ambas evolucionan ocupando el espacio libre de colisiones hasta que los dos árboles se encuentran [7]. En la Fig. 3 se muestra un resultado para un manipulador plano de dos Grados de Libertad. El inconveniente de este método estriba en la complejidad de la solución obtenida, compuesta habitualmente de segmentos diminutos que cambian continuamente de dirección y que se aproximan excesivamente a los obstáculos. Por ello esta técnica precisa un tratamiento posterior de sus soluciones para que sean practicables. Generando con RRT una población inicial de tan sólo 20 individuos y aplicando sobre ellos un algoritmo genético se pueden obtener soluciones muy eficientes como los que se muestran en la Fig. 5 a).

3 ALGORITMO GENÉTICO

El algoritmo utilizado para optimizar las trayectorias es un Algoritmo Genético Básico [3]. A partir de una población inicial se realizan un conjunto de operaciones clásicas: cruce, mutación, evaluación y selección. Cada posible solución (individuo) estará compuesta por una secuencia de configuraciones (genes) que permiten conectar la configuración inicial con la final. El número de genes no está limitado, de forma que se permite obtener individuos de distinta longitud.

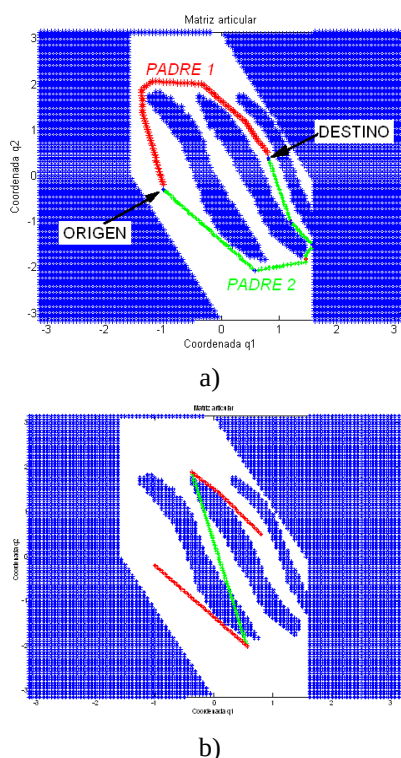


Figura 5: Operación de cruce

En la operación de cruce los individuos de la población son emparejados al azar. De cada “progenitor” (ver Fig. 5a) se extrae un fragmento de trayectoria definido por un

punto aleatorio y un extremo (la configuración inicial o la final), que se unen utilizando segmentos rectos para lograr continuidad (ver Fig. 5b). Éstos agregan puntos interpolados que incrementan la exploración del espacio de soluciones. Consecuentemente, por cada pareja se generan dos hijos, cada uno con fragmentos de ambos padres.

En la operación de mutación se elimina un fragmento de un individuo al azar (como el definido por los puntos A y B de la Fig. 6), se genera un punto aleatorio (punto P de la Fig. 6) y luego se trata de unir dicho punto a los restos del individuo con segmentos rectos (AP y PB).

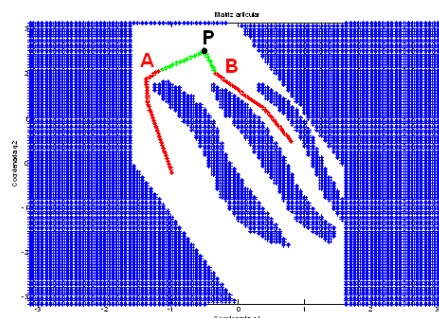


Figura 6: Operación de mutación

Finalmente se realiza la selección a partir de la población alcanzada y con la nueva se vuelve a iterar. Cuando se realiza un número determinado de iteraciones o se cumple un criterio de calidad, el algoritmo devuelve el mejor individuo como solución (ver individuo “Padre 1” de la Fig. 5 a).

4. MÉTODO BORROSO DE SELECCIÓN

Para la selección de los mejores individuos de una población, lo usual es utilizar una función de coste [2, 3]. Sin embargo es frecuente que haya preferencias entre trayectorias que son difíciles de expresar de modo analítico. Para estos casos resultan idóneas las técnicas de Decisión Multicriterio Borrosas. Estos métodos permiten establecer una clasificación entre posibles alternativas que se aproxima a la que obtendría una persona en base a los criterios establecidos. En el caso que se presenta, la incorporación del conocimiento experto y la heurística se consigue mediante la definición de un conjunto de pesos difusos que determinan la importancia relativa que tiene cada criterio. Asimismo, se introducen funciones de preferencia cuya forma se establece mediante umbrales también definidos como variables borrosas. El método también permite la incorporación de criterios borrosos que modelen la incertidumbre asociada a los actuadores y articulaciones del manipulador. Esto último no ha sido considerado en este trabajo quedando pendiente para futuros desarrollos. Por tanto, en este artículo, la aproximación borrosa se fundamenta en la necesidad de modelar el conocimiento experto en base a términos lingüísticos.

En el problema abordado, se han seleccionado tres criterios que determinan la mayor o menor bondad de cada trayectoria. El primero es la longitud (L) de la trayectoria, entendida como la suma de las distancias entre cada par ($p_i, p_{(i-1)}$) de configuraciones consecutivas:

$$L = \sum_{i=2}^n \sqrt{(p_{ix} - p_{(i-1)x})^2 + (p_{iy} - p_{(i-1)y})^2} \quad (1)$$

El segundo criterio es la vecindad de obstáculos, dado que es preferible una trayectoria que discurra por zonas de paso amplias. Un índice apropiado para este criterio es:

$$M = \sum_{i=1}^n M_i | M_i \in \{0,1\} \quad (2)$$

Para calcular este valor se explora el entorno de cada configuración en un radio constante. Si en el mismo existe algún obstáculo entonces $M_i=1$, incrementándose el valor de M . Si no existe ninguno, entonces $M_i=0$.

El último criterio es la existencia de colisión, cuyo índice (V) se calcula de forma parecida al anterior.

$$V = \sum_{i=1}^n C_i | C_i \in \{0,1\} \quad (3)$$

En este caso, se analiza cada configuración de la trayectoria y si existe colisión se hace $C_i=1$, incrementándose el valor de V . Si por el contrario no existe colisión entonces $C_i=0$. El resultado de calcular estos criterios a los individuos de las figuras 3 y 5 se muestra en la Tabla-1. Además de estos criterios, podrían incluirse otros tales como el tiempo de ejecución de la trayectoria, consumo energético, etc. para los que habría que considerar características dinámicas del manipulador. Este aspecto será abordado en próximos trabajos.

Tabla-1: Criterios evaluados en diferentes trayectorias

Trayectoria	Longitud (L)	Vecindad (V)	Colisión (C)
RRT (fig. 3)	149,47	28	0
PADRE1 (fig. 5a)	139,91	0	0
PADRE2 (fig. 5a)	145,09	15	0
CRUCE (fig. 5b)	184,79	60	21

Estos tres aspectos son considerados en el método de decisión multicriterio FPromethee. Este método de decisión [10], es la extensión difusa del método Promethee (Preference Ranking Organization Method for Enrichment Evaluation) [1] para poder utilizar, dentro de su metodología, tanto información cualitativa como cuantitativa o crisp. Para ello se ha utilizado la teoría de conjuntos difusos en su desarrollo.

El método Promethee es un método heurístico de la Escuela Europea de Toma de Decisión (TD), ampliamente utilizado en todo tipo de problemas de TD [8]. Una de sus mejores características es que sus resultados coinciden con los que se obtendrían si fuese un experto quien realiza la misma TD, ya que la preferencia del experto incorpora el conocimiento que éste tiene para discriminar trayectorias en función de los criterios establecidos.

El método general parte de la denominada matriz de decisión difusa formada por filas de un conjunto de alternativas $\tilde{A} = \{\tilde{A}_0, K, \tilde{A}_n\}$ a ser elegidas (en este problema de TD el conjunto está formado por las trayectorias candidatas), y por columnas de un conjunto de criterios de decisión difusos $\tilde{C} = \{\tilde{C}_0, K, \tilde{C}_m\}$. Cada criterio necesita definir una meta: minimizar o maximizar. Además el método define un conjunto de pesos difusos $\tilde{W} = \{\tilde{w}_0, K, \tilde{w}_m\}$ que determinan la importancia relativa que cada criterio tiene a la hora de ordenar la trayectorias.

Para integrar las preferencias humanas, el método utiliza por cada criterio una función de preferencia $F(\tilde{A}_i, \tilde{A}_k) \in [0,1]$ que refleja el cambio de preferencia que tendría un ser humano ante la comparación de dos trayectorias conforme a su diferencia (Fig.7). Algunas funciones de preferencia necesitan uno o dos umbrales para especificar un límite del cambio de preferencia. Así, por ejemplo, se tiene para la función de la Fig. 7:

- 1.- Umbral de Indiferencia ($\tilde{q}$). Indica el umbral a partir del cual se experimenta un cambio de preferencia de una trayectoria por otra.
- 2.- Umbral de Preferencia ($\tilde{p}$). Indica el valor a partir del cual, siempre se prefiere estrictamente a una trayectoria sobre otra independientemente de la diferencia entre ambas.

Estos umbrales se definen mediante etiquetas atendiendo a la variable lingüística mostrada en la Fig. 8.

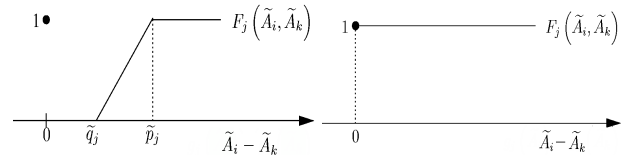


Figura 7. Funciones de preferencia lineal y estricta.

A partir de la matriz de decisión, el método FPromethee define la relación de sobreclasificación mediante la expresión:

$$p(\tilde{A}_i, \tilde{A}_k) = p_{i,k} = \frac{\sum_{j=1}^n \tilde{w}_j \cdot F_j(\tilde{A}_i, \tilde{A}_k)}{\sum_{j=1}^n \tilde{w}_j} \quad (4)$$

Esta relación representa la credibilidad de que la alternativa $\tilde{A}_i$ sobreclasifica a la alternativa $\tilde{A}_k$. Obsérvese que (4) involucra sumatorios de variables borrosas y por tanto obedecerá a las reglas de la aritmética difusa [10]. Finalmente $p_{i,k}$ está definida como el cociente de dos variables borrosas, por lo que habrá que establecer un método de desfuzzificación que permita establecer esta relación como el cociente de dos números reales. En el caso estudiado se ha utilizado el método de

desfuzzificación de Yagger. A partir de (4) se obtiene una matriz cuadrada de sobreclasificación, mediante la comparación de las N trayectorias entre sí, teniendo en cuenta para todos los criterios las funciones de preferencia, pesos y umbrales de preferencia en caso de que sean necesarios. A partir de la matriz de sobreclasificación, el método FPromethee define dos funciones denominadas flujos f que determinan, para cada alternativa $\tilde{A}_i$, el número de trayectorias $\tilde{A}_k$ que la superan o el número de trayectorias $\tilde{A}_k$ que son superadas por $\tilde{A}_i$. Ésos flujos son:

Flujo de entrada $f^- : \tilde{A} \rightarrow R$ para una alternativa $\tilde{A}_i$ determina aproximadamente el n° de alternativas $\tilde{A}_k$ que superan a la alternativa $\tilde{A}_i$. Este flujo se define como:

$$f^-(\tilde{A}_i) = \sum_{k=1, k \neq i}^m p(\tilde{A}_k, \tilde{A}_i) \quad (5)$$

Flujo de salida $f^+ : \tilde{A} \rightarrow R$ para una alternativa $\tilde{A}_i$ determina aproximadamente el n° de alternativas $\tilde{A}_k$ que son superadas por la alternativa $\tilde{A}_i$:

$$f^+(\tilde{A}_i) = \sum_{k=1, k \neq i}^m p(\tilde{A}_i, \tilde{A}_k) \quad (6)$$

Finalmente el método FPromethee clasifica el conjunto de trayectorias mediante una relación de superación S . Gracias a esta relación, el método determina para cada dos trayectorias cuál es superior. La relación es:

$$S(\tilde{A}_i, \tilde{A}_k) \Leftrightarrow f^+(\tilde{A}_i) \geq f^+(\tilde{A}_k) \wedge f^-(\tilde{A}_i) \leq f^-(\tilde{A}_k) \quad (7)$$

Existe dentro de la familia de métodos FPromethee el método FPromethee II que comparte la metodología de su antecesor pero obtiene a cambio una ordenación total de todas las trayectorias. El método FPromethee II define un **Flujo de neto** $f : \tilde{A} \rightarrow R$ como la diferencia del flujo de salida f^+ y el flujo de entrada f^- . Este flujo determina para una alternativa $\tilde{A}_i$ el nivel en el que supera al resto de alternativas o por el contrario el nivel en el que es superada por el resto. A partir del flujo neto, el método FPromethee II clasifica el conjunto de trayectorias mediante la relación de superación S definida en (7):

$$S(\tilde{A}_i, \tilde{A}_k) \Leftrightarrow f(\tilde{A}_i) \geq f(\tilde{A}_k) \quad (8)$$

Para el problema real (robot CAPAMAN), la selección de los mejores individuos se realiza mediante el método FPromethee II. Las alternativas de selección son los individuos del algoritmo genético que codifican la información de las distintas trayectorias. Los criterios de

selección son la longitud L de la trayectoria (1), la vecindad M de obstáculos (2) y por último la colisión con los obstáculos V (3). Una vez ordenadas las trayectorias, se escogen las mejor clasificadas y se desechan las situadas en los últimos lugares.

La información necesaria para la aplicación del método en cada época del algoritmo genético se puede ver en la Tabla-2. En ella se observan las etiquetas lingüísticas que han sido escogidas para los pesos de cada criterio y los umbrales de las distintas funciones de preferencia. Las etiquetas seleccionadas, están definidas en la variable lingüística presentada en la Fig. 8.

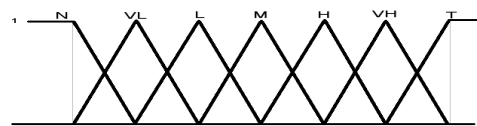


Figura 8. Variable lingüística utilizada.

Con N= nulo, VL= muy bajo, L= bajo, M= medio, H= alto, VH= muy alto y T= total.

Tabla-2: Criterios de selección de trayectorias

Criterio	Longitud (L)	Vecindad (V)	Colisión (C)
Meta y objetivo	Minimizar	Minimizar	Minimizar
Pesos $\tilde{W}$	Alto	Medio	Total
Función Preferencia (figura 7)	Lineal entre Umbrales	Lineal entre Umbrales	Estricta Preferencia
Umbral de Indiferencia $\tilde{q}$	Muy Bajo	Bajo	--
Umbral de Preferencia $\tilde{p}$	Medio	Alto	--

La aplicación del método promethee II a los resultados de la tabla-1 nos dan un flujo neto de Padre1= 0,72, Padre2= 0.62, RRT= 0.49 y Cruce= -1,83 lo que demuestra que la mejor trayectoria generada es la etiquetada como Padre1 que puede observarse en la figura 5a).

6. APLICACIÓN A CaPaMaN

A continuación se muestra la aplicación del método a la planificación de trayectorias para el manipulador paralelo CAPAMAN [2]. El planteamiento del problema consiste en hallar una trayectoria entre dos configuraciones libres de colisión. La ilustración 9 (a) muestra la descripción geométrica de la configuración inicial y la configuración final que se desea alcanzar. Se representa también un obstáculo que impide el movimiento directo de la plataforma desde una configuración a otra. En la Fig. 9 (b) se presenta en un color más oscuro la trayectoria del punto final de la herramienta que proporciona el método propuesto. Para simplificar el dibujo sólo se han dibujado los prismas de las herramientas y el obstáculo. Esta solución puede compararse con una de las trayectorias proporcionadas por el RRT que se ha representado en un color más claro. En la Fig. 9 (c) se ilustra la trayectoria articular de la solución obtenida. Obsérvese que el

espacio articular es de dimensión tres, formado por los ejes que representan los valores de los ángulos de las articulaciones (Alfa1 , Alfa2 y Alfa3). En color oscuro se representan las configuraciones con colisión y que en color claro se presenta la trayectoria generada. En la Fig. 9 (e) se muestra la evolución de las variables articulares desde su valor inicial a su valor final.

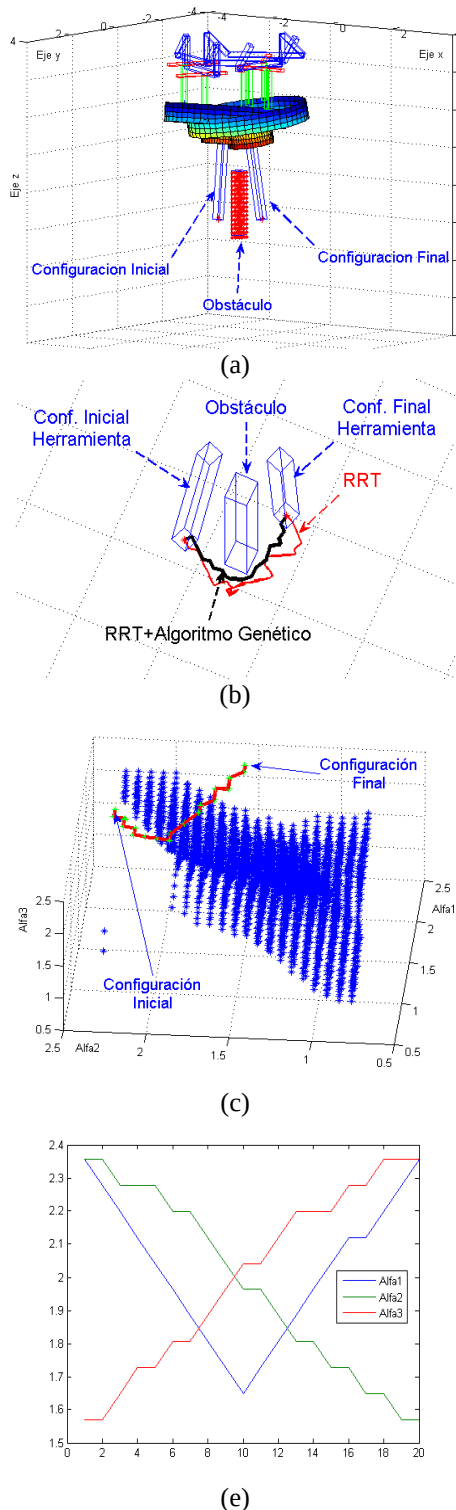


Figura 9: Aplicación del método al robot CAPAMAN

Los resultados mostrados han sido obtenidos mediante simulación. El tiempo medio necesario para conseguir una población inicial sin colisión está entorno a 0,25 s. y para la generación de una solución óptima entorno a los 34 s.

7 CONCLUSIONES

En este artículo se presenta un nuevo método para generar trayectorias para robots manipuladores. Las soluciones obtenidas permiten conectar, sin colisión, una configuración inicial y otra final. El algoritmo propuesto combina técnicas de búsqueda aleatoria (RRT), Algoritmos Genéticos y Métodos de Decisión Multicriterio Borrosos. Así, se consiguen trayectorias óptimas y libres de colisión. El método se ha aplicado al manipulador paralelo CaPaMan desarrollado en la Universidad de Cassino. En futuros trabajos se abordará la inclusión de más criterios, tanto crisp como borrosos.

Referencias

1. Brans, J. y Vincke, P. (1985). A preference ranking organization method, the PROMETHEE method. *Management Science*, 31, 647-656.
2. Carbone G., Ceccarelli M., Oliveira P.J., Saramago S.F.P., Carvalho J.C.M., "An Optimum Path Planning of CaPaMan (Cassino Parallel Manipulator) by Using Inverse Dynamics", *Robotica: An International Journal*, Vol.26, N.2, 2008, pp.229-239, 2008.
3. D. E. Goldberg, "Genetic Algorithms in Search Optimisation, and Machine Learning" (Addison-Wesley, Reading, MA, 1989).
4. D. López, F. Gómez-Bravo, F. Cuesta, A. Ollero (2006). "Aplicación del algoritmo RRT a la planificación de movimientos en robots no holónomos". *Revista Iberoamericana de Automática e Informática Industrial (RIAI)*. Vol. 3, pp. 56-67.
5. Gómez-Bravo, F., Ollero, A., Cuesta, F., and Lopez, D. (2008). A New Approach for Car Like Robots Manoeuvring Based on Rrt. *Robotica*. N. 71. pp. 10-14.
6. J. P. Merlet, "Parallel Robots", *Kluwer Academic Publishers* Dordrecht, 2006.
7. LaValle, S. M. and Kuffner, J. J. (1999). Randomized kinodynamic planning. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 473-479.
8. Martin J.M., Blanco A., Requena I., Fajardo W. (2003). Constructing Linguistic Versions For The Multicriteria Decision Support Systems Preference Ranking Organization Method For Enrichment Evaluation I And II. *International Journal Of Intelligent Systems*, 18, 711-731.
9. Ollero, A. (2001). Manipuladores y Robots Moviles. Marcombo Boixareau.
10. Martin J.M., Blanco A., Delgado M., Polo M.P. (2001). *Fuzzy methods of multicriteria decision. The methods fpromethee(2T)*. World Mult. On Systemics, Cybernetics And Informatics, Orlando, USA.

Aprendizaje de la sintaxis para la descripción de escenas compuestas por figuras geométricas*

David Pérez¹ Sergio Guadarrama¹

¹ European Centre for Soft Computing
Mieres (Asturias), España.
{david.perez, sergio.guadarrama}@softcomputing.es

Resumen

La forma en que las personas describimos los objetos de una escena tiene características particulares concretas. Nuestro trabajo será tratar de aprender estas características. Para ello, primeramente necesitamos conocer la sintaxis de estas descripciones, motivo principal de este paper. El objetivo final será describir nuevas escenas de forma automática.

1 Introducción

El trabajo aquí presentado está englobado dentro del proyecto Smart-Bees, en el que se tratan de realizar descripciones automáticas de objetos presentes en una escena. Las escenas están formadas por varios objetos, por lo que las descripciones han de discriminar al elemento seleccionado de forma inequívoca. Estas escenas constarán de distintos objetos geométricos coloreados, dispuestos en un lienzo negro, como se puede ver en la figura 1. Uno de estos objetos, el que se debe describir, tiene el borde punteado para diferenciarlo del resto.

Para resolver este problema se han de cubrir distintas fases: obtención de descripciones de objetos hechas por los usuarios; aprendizaje del léxico y de la sintaxis de estas frases; aprendizaje de la semántica de las palabras utilizadas teniendo en cuenta las características del objeto descrito; generación de frases siguiendo la sintaxis y semántica previamente adquiridas; validación de las frases generadas.

La parte del problema al que nos enfrentamos en este documento es el correspondiente a las primeras fases del aprendizaje; la obtención de frases, el aprendizaje

*Este trabajo ha sido financiado por la Fundación para el progreso del Soft Computing, el Ministerio de Ciencia e Innovación Español (MICINN) y el Fondo Social Europeo, bajo el programa Juan de la Cierva JCI-2008-3531.

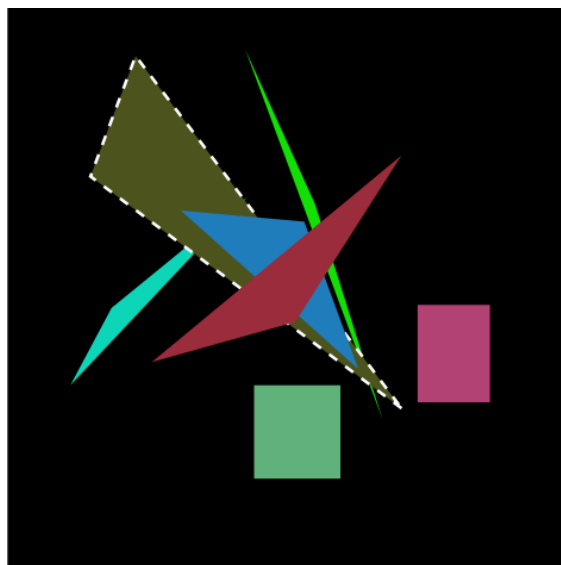


Figura 1: Ejemplo de imagen para describir

del léxico y de la sintaxis; las otras fases se explican en el trabajo [1] (también presentado en esta conferencia).

Las frases recopiladas tienen distintas estructuras, y en las mismas los usuarios tratan de describir el objeto de forma directa, haciendo referencia sólo al objeto descrito, o de forma indirecta, haciendo referencia a algún otro objeto que sirva para aclarar posibles ambigüedades.

Para llevar a cabo esta tarea, inicialmente, y tras la recolección de las frases, realizamos un clustering por tipos de palabras, considerando únicamente las palabras que tengan una frecuencia significativa. Una vez obtenidos estos grupos se creará una gramática que reconozca las frases, con la cual se generarán frases que podrán ser valoradas. Hemos probado distintos tipos de clusterings y de aprendizaje de gramáticas para comprobar que combinación nos da mejores resultados.

Evaluar los resultados de las opciones contempladas para cada caso será una parte importante del documento, destacando, en base a estos resultados, la mejor opción en cada caso.

El problema de generar frases de forma automática para describir un objeto dentro de una escena, nuestro objetivo global, ha sido estudiado por varios autores [2, 3]. En [2] se describen de forma detallada las distintas fases del problema al que nos enfrentamos.

En cuanto a las distancias a emplear en el clustering, el método basado en proximidad de palabras, utilizado por los niños en su aprendizaje, pueden verse descripciones del mismo en [6] y usos similares al que se le da aquí en [3, 4, 7]. Por otra parte, las distancias basadas en coocurrencias puede verse en [9].

Finalmente, los algoritmos de aprendizaje de gramáticas utilizados pueden verse en los documentos [3, 5].

Este documento comienza con un resumen de los puntos claves del trabajo y de las soluciones adoptadas en cada uno de estos casos. Pasamos, posteriormente, a mostrar y valorar los resultados, dejando, finalmente, un apartado de conclusiones y trabajo a realizar en el futuro.

2 Propuesta

2.1 Recolección de datos

La obtención de descripciones de objetos de una escena se realiza a través del portal

<http://www3.softcomputing.es/smart-bees>

En éste se presenta un apartado, “Describe images”, en el que los usuarios pueden realizar las descripciones en inglés de un objeto seleccionado dentro de una escena, que ha sido generada de forma aleatoria. Hay 350 escenas que contienen 7 objetos con formas, colores y posiciones distintas.

Se han recogido 360 frases, con las que se realizará el entrenamiento, de 34 usuarios distintos.

Tras estudiar el problema del análisis sintáctico vamos a mostrar los pasos relevantes que hemos seguido en pseudo-código:

```
carga(frases, palabras);
frases=filtrado(frases);
palabras=filtrado(palabras);
[train test]=separacion(frases);
clases=clustering(palabras,train);
gramatica=train_gramatica(clases,train);
frasesGeneradas=generar_frases(gramatica,train);
```

```
evaluacion(frasesGeneradas,test);
```

A su vez, describiremos a continuación las diferentes pasos, así como las alternativas barajadas en cada caso, de la siguiente forma:

Filtrado de frases: Se han filtrado las frases teniendo en cuenta únicamente las que tenían un número de palabras igual o menor que 10. Las frases de un tamaño mayor a 10 se consideran demasiado complejas como para ayudar al aprendizaje. Con este filtrado se descartan un 5% de las frases.

Filtrado de palabras: Se consideran únicamente las palabras que tenían una frecuencia mayor o igual a 10, para evitar palabras poco representativas o con errores.

Generación de conjuntos de entrenamiento y test:

Para ello realizamos dos tipos de división. Una en la que se utilizarán todas las frases para entrenamiento y validación; y otra en la que dividiremos los datos disponibles 80% para entrenamiento y el 20% restante para evaluar lo aprendido.

Formación de clases de palabras: Se han comparado dos tipos de formación de clases dependiendo de como se construyen las distancias entre las palabras: según las coocurrencias de palabras en la misma frase y otra en función de la proximidad de las mismas. Esta cuestión será abordada más a fondo en el apartado 3.

Aprendizaje de gramáticas: Se han probado cuatro algoritmos que aprenden gramáticas: entrenando a una cadena de Markov con las secuencias de palabras presentes en los datos; entrenando a una cadena de Markov con las secuencias de clases previamente creadas; escogiendo una estructura al azar de las presentes en los datos de entrenamiento; finalmente generando el bigrama de probabilidades de que después de una clase venga otra. Se explicará con más detalle este punto en la sección 4.

Generación de frases: Una vez generada la gramática correspondiente en cada caso pasaremos a generar las frases haciendo uso de esta gramática. Para ello tenemos dos tipos, utilizar directamente la salida de la secuencia de palabras que nos de la gramática, para el caso de cadena de Markov de secuencias de palabras, o convertir la secuencia de clases a palabras utilizando el método de Monte Carlo [10] de acuerdo con la frecuencia de las palabras de cada clase, utilizada para el resto de aprendizajes de gramáticas. Se verá más a fondo en el punto 5.

3 Formación de clases de palabras

Agruparemos una serie de datos, en nuestro caso concreto palabras, de acuerdo con un criterio de cercanía y utilizando un clustering jerárquico, en el que se van juntando las palabras más próximas de forma iterativa. Esta cercanía se define en términos de una determinada función de distancia.

Pasamos ahora a describir las distancias utilizadas para realizar el clustering jerárquico que agrupa las palabras en clases. Como anticipo podemos decir que utilizamos medidas de distancia que únicamente se centran en el orden y aparición de dichas palabras en las frases dadas. Tendremos una distancia basada en coocurrencias y otra basada en la proximidad de las palabras dentro de la frase.

3.1 Distancia basada en coocurrencias

Palabras que desempeñan el mismo papel en una frase tienden a ocurrir de modo excluyente, es decir, dos palabras que pertenecen a la misma clase no suelen aparecer en la misma frase. Por ejemplo, palabras que hagan referencia a la forma no será común encontrarlas a la vez en una frase, al menos en las descripciones directas. Se trata de asignar una distancia baja, por tanto facilidad para que se encuentren dentro de la misma clase, a aquellas palabras que aparezcan pocas veces en la misma frase.

Denotaremos el conjunto de M descripciones como $U = u_1, u_2, \dots, u_M$. Las palabras de todo el vocabulario serán representadas por $W = w_1, w_2, \dots, w_N$ con N el número de palabras del vocabulario manejadas. Definimos también $S(w_i)$ como el número de apariciones de la palabra w_i en todas las descripciones dadas en U . Como se define en [2], un indicador de coocurrencia de palabras es:

$$\pi(u, w_i, w_j) = \begin{cases} 1 & \text{si } w_i \text{ y } w_j \text{ aparecen en } u \\ 0 & \text{sino} \end{cases}$$

Basándonos en π obtenemos la matriz R de coocurrencias de la siguiente forma:

$$R(w_i, w_j) = \sum_{u \in U} \pi(u, w_i, w_j)$$

Una vez calculada esta matriz, pasamos ahora a calcular una matriz de distancias entre palabras, D , que será la base de cálculo de los distintos clusters en los que se dividirán las palabras. La misma será calculada como: (diferiendo con la dada en [2] en este caso)

$$D(w_i, w_j) = \frac{R(w_i, w_j)}{\min(S(w_i), S(w_j)) + 1}$$

Finalmente esta matriz se acota entre 0 y 1 dividiendo todos los valores por el máximo valor de la misma.

Una vez hecho esto, la matriz de distancias será pasada a una función de clustering jerárquico que irá agrupando las palabras más cercanas primero. Se han probado, no obstante, varios parámetros de entrada a esta función de clustering para comprobar cual nos da el mejor resultado.

Obteniendo las siguientes clases al aplicar el clustering con esta distancia:

Clase 1 = { TOP, OVAL, ELLIPSE, BIG }

Clase 2 = { BOTTOM, YELLOW }

Clase 3 = { LEFT, SQUARE }

Clase 4 = { ON, IN, AT, BEHIND }

Clase 5 = { CIRCLE, TRIANGLE, RECTANGLE }

Clase 6 = { THE }

Clase 7 = { LIGHT, BACKGROUND, A, RIGHT, DARK, FRONT, PURPLE }

Clase 8 = { PINK, BLUE, GREEN, ORANGE, RED, VIOLET, BROWN }

3.2 Distancia basada en la proximidad de las palabras

Esta forma de cálculo de distancias va más allá de la coocurrencia de las palabras en la misma frase y también considera que las palabras se encuentren próximas en las frases en las que aparecen, haciendo que estas palabras sean clasificadas en distintos grupos.

Los cálculos realizados son similares a los del caso de la distancia basada en coocurrencias, salvo que para la función R y π se calculan de la siguiente forma:

$$R_l(w_i, w_j) = \sum_{u \in U} \pi(u, w_i, w_j, l)$$

$$\pi(u, w_i, w_j, l) = \begin{cases} 1 & \text{si } w_i \text{ y } w_j \text{ ocurren en } u \\ & \text{y sus posiciones distan } l \\ 0 & \text{sino} \end{cases}$$

Como concluyeron los autores de [4], las palabras próximas más relevantes son las dos precedentes y las dos siguientes ($l = \{-2, -1, 1, 2\}$) y por tanto así lo hemos hecho en este trabajo.

Por ejemplo, para $u = \{\text{"the blue rectangle"}\}$ la llamada $\pi(u, \text{"the"}, \text{"rectangle"}, 2) = 1$ ya que sus posiciones distan 2, $\pi(u, \text{"the"}, \text{"blue"}, 1) = 1$, ...

La distancia utilizada en este caso es la del coseno

$$D(w_i, w_j) = \frac{\sum_{w_k \in W} \min(R_l(w_i, w_k), R_l(w_j, w_k))}{\sqrt{\sum_{w_k \in W} R_l(w_i, w_k)} + \sqrt{\sum_{w_k \in W} R_l(w_j, w_k)}}$$

Obtenemos las siguientes clases al aplicar este método a los datos manejados:

Clase 1 = { THE, A }

Clase 2 = { BACKGROUND, FRONT }

Clase 3 = { CIRCLE, OVAL, TRIANGLE, RECTANGLE, ELLIPSE, SQUARE }

Clase 4 = { ON, IN, AT, BEHIND }

Clase 5 = { LIGHT, BIG, DARK }

Clase 6 = { TOP, BOTTOM, RIGHT, LEFT }

Clase 7 = { PINK, BLUE, GREEN, ORANGE, RED, YELLOW, PURPLE, VIOLET, BROWN }

4 Aprendizaje de gramáticas

A la hora de aprender las gramáticas que describan la sintaxis se han contemplado cuatro opciones, una que proporciona una gramática implícita y otras tres que proporcionan una gramática explícita. Se pondrá a prueba a las mismas para decidir cual nos proporciona la mejor solución.

4.1 Aprendizaje cadena de Markov con secuencias de palabras

Utilizando todas las secuencias de palabras que nos proporcionan los datos de entrenamiento aprendemos una cadena de Markov que generará frases como secuencias de palabras.

Al generar frases de este modo, el coste computacional será muy alto, e incluso el algoritmo en muchos casos no llega a converger.

4.2 Aprendizaje cadena de Markov con secuencias de clases

Al igual que en la anterior solución aprendemos una cadena de Markov, aunque en este caso con los datos de las clases de las frases de entrenamiento, en vez de las palabras. Al producirse más frecuentemente una secuencia concreta de clases que de palabras esta opción nos proporcionará, a priori, mejores resultados que la previa.

El coste computacional de esta opción es alto y las frases tendrán un tamaño no prefijado.

4.3 Patrones de las frases de entrenamiento

De todas las frases de entrenamiento se cogerá de forma aleatoria una de las estructuras de clases allí presentes.

El coste computacional de este método es muy bajo.

4.4 Utilización del bigrama de probabilidades de clases

En este caso se construye un bigrama con las probabilidades de transición de ir de una clase a otra, de forma similar a lo expuesto en [2]. Se muestra un ejemplo de este tipo de bigramas en la figura 2. En éste los nodos S y E son los correspondientes a principio y fin de sentencia respectivamente, mientras que los números representan las clases de palabras, en este caso son las mostradas en el apartado 3.2, generadas con la distancia basada en la proximidad.

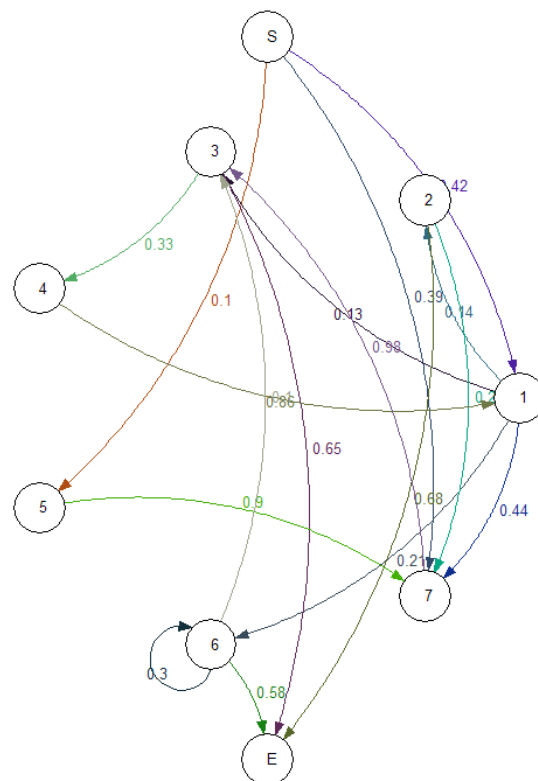


Figura 2: Ejemplo de bigrama de probabilidades de clases

Una vez construido este bigrama utilizaremos la matriz asociada al mismo como matriz de transición de una cadena de Markov que nos dará secuencias de clases.

La complejidad computacional de este método es media.

5 Generación de frases

Para el caso del aprendizaje de una cadena de Markov con secuencias de palabras esta generación es directa utilizando la secuencia de palabras que genere la cadena de Markov.

En el resto de casos se utilizará el método de Monte Carlo [10] para determinar que palabra utilizar en cada clase, en función de la frecuencia de aparición de las palabras con respecto a las demás palabras en su clase.

Algunos ejemplos de frecuencias de palabras dentro de su clase, para las clases de la sección 3.2, son:

Clase 1 = { THE [88%], A [12%] }

Clase 2 = { BACKGROUND [49%], FRONT [51%] }

Clase 3 = { CIRCLE [21%], OVAL [6%], ... } ...

6 Resultados

En esta sección se presentarán brevemente los resultados obtenidos con los algoritmos antes expuestos y comparando las distintas alternativas barajadas. Para estas pruebas se han evaluado las frases generadas en cada caso en función de precisión y recall de los mismos:

Precisión : Es una medida de exactitud o fidelidad.

En nuestro caso será el porcentaje de frases del conjunto de las generadas que son reproducciones exactas de, al menos, una frase de entrenamiento. Una precisión del 1.0 significaría que todas las frases generadas están presentes en los datos de entrenamiento.

Recall : Es una medida de completitud. Para el caso que nos centramos, consistirá en el porcentaje de frases de entrenamiento que el sistema es capaz de generar. Un recall de 1.0 se daría en el caso de que todos los datos de entrenamiento son generados por el sistema al menos una vez.

Como una primera prueba, se han realizado 100 repeticiones de 1000 frases generadas, utilizando un 80% de las frases disponibles para entrenar y el 20% restante para validar las generadas, utilizando los 2 tipos de distancias para el clustering y las 4 formas de aprendizaje de gramáticas mencionadas. Estas frases han sido evaluadas de forma automática obteniendo resultados de precisión y recall.

En la tabla 1 presentamos la media y la desviación típica de las 100 repeticiones al generar 1000 frases con los cuatro algoritmos de generación de frases anteriormente introducidos para la distancia basada en coocurrencias. Y en la tabla 2 mostramos los mismos

para el caso de las distancias basadas en la proximidad de las palabras.

Algoritmos	Precisión	Recall
Markov palabras	0.076±0.026	0.037±0.009
Markov clases	0.028±0.017	0.024±0.008
Patrones	0.032±0.015	0.027±0.007
Bigrama	0.034±0.019	0.023±0.007
Media	0.043±0.028	0.028±0.009

Tabla 1: Comparativa de algoritmos de generación de frases con clases basadas en coocurrencia.

Algoritmos	Precisión	Recall
Markov palabras	0.089±0.026	0.040±0.009
Markov clases	0.065±0.027	0.032±0.008
Patrones	0.088±0.019	0.043±0.007
Bigrama	0.114±0.025	0.042±0.007
Media	0.089±0.030	0.039±0.009

Tabla 2: Comparativa de algoritmos de generación de frases con clases basadas en proximidad de palabras.

Seguidamente, para comprobar la convergencia de los resultados obtenidos, utilizando todos los datos para entrenar y validar, se han calculado los recall y precisión, al generar 100, 1.000, 10.000, 100.000, 1.000.000 frases, con las distancias basadas en proximidad. Se pueden ver las evoluciones de la precisión en la figura 3 y del recall en la figura 4

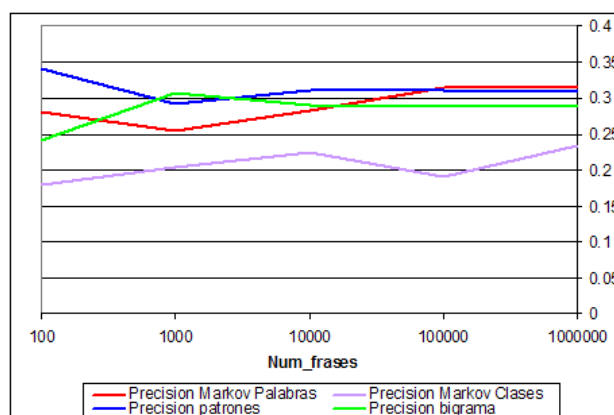


Figura 3: Evolución de la precisión.

Finalmente, han sido generadas 100 frases para evaluar su corrección sintáctica manualmente por dos personas con los 3 mejores métodos de aprendizajes de gramáticas y utilizando para realizar el clustering la distancia basada en la proximidad de las palabras, que

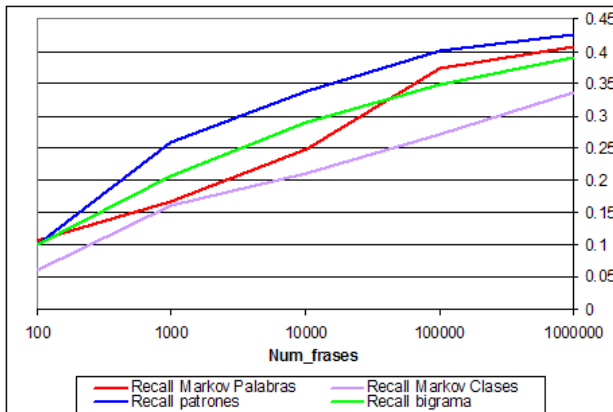


Figura 4: Evolución del recall.

obtiene mejores resultados que la basada en coocurrencias. Estos resultados se pueden ver resumidos en la tabla 3.

Algoritmos	Porcentaje de frases correctas
Markov palabras	0.68
Patrones	0.88
Bigrama	0.79

Tabla 3: Resumen de resultados obtenidos manualmente.

7 Análisis de resultados

Como se puede apreciar en las tablas 1 y 2 existe una mejora al utilizar la distancia basada en la proximidad de palabras para todos los tipos de aprendizaje de gramáticas.

Al comparar las frases generadas por los cuatro métodos de aprendizaje de gramáticas con los datos de test, puede verse en las figuras 3 y 4 que el método de aprender una cadena de Markov con las clases obtiene los peores resultados. Y que los otros 3 métodos obtienen unos resultados muy similares para todos ellos.

Al comparar las frases generadas por los otros 3 métodos de forma manual, tabla 3, los mejores resultados se obtienen con el método basado en utilizar patrones del conjunto de entrenamiento.

8 Conclusión y trabajo futuro

Al analizar todas las pruebas realizadas con las distintas distancias de clustering y los distintos métodos de aprendizaje de gramáticas concluimos que la mejor

solución para construir frases sintácticamente correctas (88%) es utilizar la distancia basada en la proximidad de palabras para crear las clases de palabras y, utilizar patrones extraídos de los datos de entrenamiento. La segunda mejor combinación es la de utilizar el bigrama de probabilidades, junto con la distancia basada en proximidad, ya que se obtienen unos resultados cercanos a la mejor opción estudiada (79%).

Como se indica al inicio del documento, el trabajo aquí realizado pertenece a la búsqueda de un objetivo más ambicioso de describir objetos dentro de una imagen de forma adecuada, por lo que generar frases sintácticamente correctas no es suficiente, sino que deben ser semánticamente correctas.

La generación de frases compuestas, donde se hace referencia a más de un objeto, queda como trabajo futuro. Ya que existirán casos donde los objetos que acompañan al que se debe describir tendrán características similares al seleccionado por lo que la frase ha de ser más compleja.

Referencias

- [1] Sergio Guadarrama, David Pérez. Learning Semantic Models from Descriptions of Shapes made by Web-Users. 2009.
- [2] Deb K. Roy. Learning Visually-Grounded Words and Syntax for a Scene Description Task. 2002.
- [3] Peter Gorniak, Deb Roy. Grounded Semantic Composition for Visual Scenes. 2004.
- [4] Martin Redington, Nick Chater, Steven Finch. Distributional Information: A Powerful Cue for Acquiring Syntactic Categories. 1998.
- [5] Zach Solan, David Horn, Eytan Ruppín. Unsupervised Learning of natural languages. 2005.
- [6] Steven Finch, Nick Chater. Distributional Bootstrapping: From Word Class to Proto-Sentence. 1994.
- [7] Fernando Pereira, Naftali Tishby, Lillian Lee. Distributional Clustering of English Words. 1993.
- [8] Percy Liang, Michael I. Jordan, Dan Klein. Learning Semantic Correspondences with Less Supervision. 2009.
- [9] Thomas Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. 2001.
- [10] Nicholas Metropolis, Stanislaw Ulman. The Monte Carlo Method. 1949.

A PROTOTYPE OF A MACHINE SPEAKING WITH EMOTIONS

Albert van der Heide¹ Gracian Trivino¹

¹ European Centre for Soft Computing (ECSC), Mieres (Asturias)
{albert.vdheide,gracian.trivino}@softcomputing.es

Abstract

Many computational applications use natural language to communicate with their users. Nevertheless, this communication usually is unnatural due to the absence of emotional content.

This paper presents a prototype of a computational system that, thanks to a simulated model of emotions, is capable of including emotional content in its spoken utterances. The system's emotional content is based on the conditions of light and temperature, and implemented using a Fuzzy Finite State Machine.

Keywords: model of emotions, fuzzy finite state machine, emotional speech.

1 INTRODUCTION

Currently there are quite a number of systems that attend to humans automatically. A number of these systems use voice to transmit content, for example, in the domain of telephony many automated attendants exist, various web applications use speech, and there are many more examples.

However, the interaction with these machines is currently still too artificial. Usually these machines are insensitive to the emotional content being expressed by the communication partner, as well as incapable of modifying the expressed voice. This sensitivity is of importance in human interaction and actually transmits a lot of valuable information.

In order to increase the effectiveness and acceptance of these automated systems these machines would need to respond in a natural way. One way that this can be

improved is by making these machines sensitive to the emotional content expressed by a human communication partner, as well as being able to express themselves emotionally as humans do. An emotional aware machine would be able to adjust its behavior in response to the emotional content transmitted by the communication partner.

In this article, we elaborate upon a prototype that was created to simulate an emotional state of a machine. The objective of the prototype is to simulate and express emotional state.

The prototype consists of a virtual being that appears as if it has an emotional state that depends on the weather, such that when the weather is 'bad' the emotional state deteriorates and when the weather is good the emotional state improves. The emotional state is influenced by two parameters: luminosity and temperature during the day. The emotional state is expressed by emotionally modified speech. A depiction of the basic scheme is given in figure 1.

The emotional state of the system is fuzzy, i.e., at any point in time the system can be in various emotional states to certain degree.

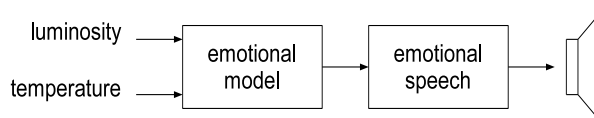


Figure 1: The general scheme of the system.

2 A SIMPLE MODEL TO SIMULATE THE EMOTIONAL STATE

Currently there is no consensus on how emotions should be modeled, nor on what representation to choose. It is clear however that emotions are an

integral part of human functioning and are a key part in human cognition [4, 11]. There are general models about the universality of emotions [3, 8] and the expression of emotions in faces [6]. Several researchers have begun to model emotions for conversational agents, for example [13, 5, 20].

We believe that fuzzy logic provides an expressive language that enables our model to represent complex emotional states and behaviors. Fuzzy logic makes it possible to deal with multiple and conflicting emotions and makes it possible to output a mixture of emotions. Fuzzy logic has previously been used in emotion simulation, for example [7].

In previous studies we have learned that Fuzzy Finite State Machines (FFSM) are suitable tools for modeling signals that follow an approximately repetitive pattern [1, 16, 17]. Here, we use a FFSM to model how luminosity and temperature patterns during the day can influence the simulated emotional state of a virtual being.

As part of a first prototype we have developed a FFSM that uses two input variables, namely the average temperature and luminosity during the last 24 hours. These variables are fuzzified, subsequently fuzzy rules determine the new state activation based on input and current activation of states. In short, the prototype explores how ambient parameters can influence emotional states of the system.

A FFSM is a tuple $\{S, U, Y, f, g, S_0\}$. We describe each one of its components in the following sections.

2.1 Linguistic Fuzzy States

In the first steps of modeling a system, the designer must define a set $\{S_i\}$ of fuzzy sets that summarize the relevant states of the system. The *linguistic fuzzy state* of the system (S) is a linguistic variable that take its values in the set of linguistic labels $\{S_1, S_2, \dots, S_n\}$.

We represent the fuzzy state of the system with a *state activation vector* $s(t) = (s_1(t), s_2(t), \dots, s_n(t))$ where $s_i \in [0, 1]$ and $\sum_1^n s_i = 1$.

We will see in section 2.2 how to obtain $s(t)$ using a set of fuzzy rules.

S_0 is the initial state or *state activation vector* at $(t = 0)$, e.g., $(1, 0, 0, 0)$.

2.1.1 Input variables

U is the input vector $(u_1, u_2, \dots, u_{nu})$. Typically, U is a set of linguistic variables obtained after fuzzification of numerical measures obtained from sensors.

The emotion-FFSM is sensitive to two linguistic vari-

ables [21]; *luminosity* and *temperature*, both obtained from sensors attached to a window in our laboratory. The sensors measure the light intensity coming from the outside of the building as well as the temperature near the window in one minute intervals. In a preprocessing state this signal is smoothed by taking the running average over 10 samples. The basic input signal to the system is the average luminosity and the average temperature during the last 24 hours. Linguistic variables take values 'high', 'medium', and 'low' to represent the current values of luminosity and temperature. The membership functions are depicted in figure 2.

These membership functions have been defined automatically based on the range in input values. (Approximately 35% of the samples are defined as 'high', 30% as 'medium', and 35% 'low').

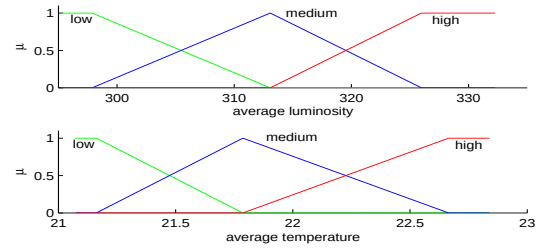


Figure 2: The linguistic variables for luminosity and temperature.

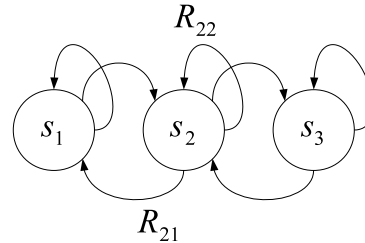


Figure 3: A depiction of the Fuzzy Finite State Machine and the possible transitions within the model.

2.2 Transition function f

We obtain the new *activation vector* using the *transition function* $S[t + 1] = f(U[t], S[t])$.

Once the designer has identified the relevant states in the model, he/she must define the rules that govern the temporal evolution among the states of the system. In our approach these rules are fuzzy rules. In Figure 3 depicts a graph that shows the allowed paths between the states. Here, state s_1 represents a bad or

angry mood, state s_2 represents a neutral mood, and state s_3 represents a joyful/happy mood. As personal characteristics of the virtual being we had envisioned that it should be moody when the days are cold and dark, and happy when the days are warm and bright.

The being that we are designing is envisioned as having an pleasant personality, and therefore will be more likely to be in a happy mood than in a depressed mood. Such a personality can be created by modifying the state transition rules, as well as by making use of *TimeToStay* and *TimeToMove* linguistic variables which will be explained in the following.

We distinguish between rules to remain in a state (R_{ii}) and rules to change a state (R_{ij}).

2.2.1 Rules to remain in a state

The designer uses these rules to express the conditions of the system to remain in a specific state. We distinguish between amplitude conditions (C_A) and temporal conditions (C_T).

Using this terminology, the generic expression of these rules is formulated as follows:

R_{ii} : IF ($S(t)$ is S_i) AND (C_A) AND (C_T) THEN ($S(t+1)$ is S_i)

Amplitude conditions. The designer uses the vector of input variables U to write the fuzzy rules that describe the constraints on the input variables to remain in the state S_i . For example, $C_A = ((u_1 \text{ is High}) \text{ AND } (u_2 \text{ is High}))$.

Temporal conditions. These rules express constraints on the duration of states. The duration of a state (D_i) is calculated as:

$$\text{IF } (S_i > \delta) \text{ THEN } D_i(t+1) = D_i(t) + 1 \\ \text{ELSE } D_i(t+1) = 0;$$

where δ is a threshold that is application dependent.

Typically the temporal constraints are modeled using the linguistic label *TimeToRemain* (TR_{ii}) (see figure 4). The conditions (D_i is TR_{ii}) allows the designer to constrain the maximum time the system is in the state S_i .

In this application, we designed the virtual being as optimistic, e.g. it is not moody indefinitely, although the weather conditions might give it sufficient reason to be so. These personality characteristics have been modeled using temporal conditions.

An example rule to remain in a state is the following:

$$R_{11} : \text{IF } (S(t) \text{ is } S_1) \text{ AND } (u_1 \text{ is Low}) \text{ AND } \\ (u_2 \text{ is Low}) \text{ AND } (D_1 \text{ is } TR_{11})$$

THEN ($S(t+1)$ is S_1)

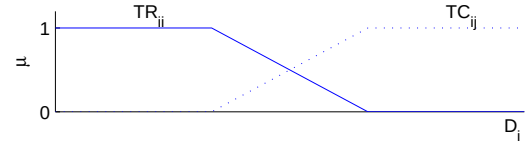


Figure 4: Linguistic labels used for defining temporal constraints, namely, *TimeToRemain* and *TimeToChange*.

2.2.2 Rules to change of state

The designer uses these rules to express the conditions that provoke the system changing from state S_i to state S_j . Also here, we distinguish between amplitude conditions (C_A) and temporal conditions (C_T).

The generic expression of R_{ij} is formulated as follows:

R_{ij} : IF ($S(t)$ is S_i) AND (C_A) AND (C_T) THEN ($S(t+1)$ is S_j)

Amplitude conditions. In a first approach these conditions coincide with the amplitude conditions to remain in the destination state of the transition. Nevertheless, some tuning could be needed to express a softer condition to change.

Temporal conditions. These conditions are based on the linguistic label *TimeToChange* (TC_{ij}) (Figure 4). The condition (D_i is TC_{ij}) allows the designer to constrain the minimum time the system must be in S_i before to move to S_j .

What follows is an example of rules used for state transitions:

$$R_{12} : \text{IF } (S(t) \text{ is } S_1) \text{ AND } (D_1 \text{ is } TC_{12}) \\ ((u_1 \text{ is Med}) \text{ OR } (u_2 \text{ is Med})) \\ \text{THEN } (S(t+1) \text{ is } S_2)$$

2.3 State activation

We calculate the degree of activation of a state using the Takagi-Sugeno-Kang paradigm [14].

In the rule-base a subset of all rules k contribute to the same destination state i . To calculate the activity of state i at time $t+1$ we take into account the rules contributing to the state i , ω_i , weighted by the degree of firing of all rules, ω_k :

$$S_i[t+1] = \begin{cases} \frac{\sum \omega_i}{\sum \omega_k} & \text{if } \sum \omega_k \neq 0 \\ S_i[t] & \text{if } \sum \omega_k = 0 \end{cases}$$

where the degree of firing for each rule ω is calculated using the minimum for the AND operator.

2.4 Output variables

Y is the output vector $(y_1, y_2, \dots, y_{no})$. Y is a summary of the characteristics of the system evolution that are relevant for the application. In our case this is directly the emotional state represented by the FFMSM. For example, in this application $Y = (0, 0.7, 0.3)$ represents an emotional state between *neutral* and *happy*.

The output vector is calculated by the output function g .

2.5 Output function g

The output function $g(U[t], S[t])$, calculates the values for the output variables $Y[t]$. As a first possible implementation of Y we have chosen the function $Y = s(t) = (s_1(t), s_2(t), \dots, s_n(t))$. In this case the model output is the current fuzzy state of the system represented as a *state activation vector*.

In the prototype application we are directly interested in the emotional state represented by the model. However, this does always not need to be the case and other output functions could be defined. For example, one could apply the average and the standard deviation to the values of the input variables while the system remained in the considered state. Furthermore, each output variable y_i could represent any parameter which is related to an internal emotional state, and usually more complex functions g are needed.

3 EMOTIONAL STATE EXPRESSION IN XML

The FFMSM provides as output directly the state current activation S . We express this emotional state through a voice which states a simple phrase.

For the voice synthesis we use Verbio Text-To-Speech (TTS) [15]. The speech can be annotated using a standard markup language based on and very similar to Speech Synthesis Markup Language (SSML) [19].

In the literature there are various articles describing how emotional speech is different from normal speech and what the parameters are that are modified [12, 18]. It should be noted however that there is not much literature specifically centered on Spanish language [10, 2, 9]. We have used the references specified here and applied these to our speech synthesis method.

For example, when a person is experiencing *joy* a number of parameters are modified. Among these parameters are the fundamental frequency, which is higher

than normal, the speech rate, which is faster than normal, etc. Additionally, according to the literature, the fundamental frequency decreases during the phrase expression. We have annotated speech to include these parameters. Table 2 gives an overview of the used parameters. The annotated speech for *joy* is as follows:

```
<prosody pitch="+40%" range="+20%" rate="-30%">Estoy </prosody>
<prosody pitch="+20%" range="+20%" rate="-30%">muy </prosody>
<prosody pitch="+20%" range="+20%" rate="-30%">feliz.</prosody>

<prosody pitch="+40%" range="+20%" rate="-30%">Todo </prosody>
<prosody pitch="+26%" range="+20%" rate="-30%">ha </prosody>
<prosody pitch="+13%" range="+20%" rate="-30%">salido</prosody>
<prosody pitch="+20%" range="+20%" rate="-30%">bien. </prosody>
```

There are quite a number of parameters that can be modified. Nevertheless, not all parameters we find in the literature can be expressed using Verbio TTS or other TTSs, for example variance in pitch, or mean / range modification of intensity (only one general parameter exists for intensity).

Another limitation of the current prototype is that it does not directly accept a mixture of emotional states as input. Therefore, for the moment, the prototypical annotations have been constructed for five types of emotional states. These annotations are related to states $\{(1, 0, 0), (0.5, 0.5, 0), (0, 1, 0), (0, 0.5, 0.5), (0, 1, 1)\}$. For the generation of an output sentence at a certain point in time the most similar state's annotation is selected.

Table 1: Emotional states and acoustic features.

	sadness	neutral	joy
pitch mean	<		>
pitch range	<		>
pitch contour	\		/
intensity	<		
speech rate	>		<

4 EXPERIMENTATION

To test our prototypical system we have applied temperature and luminosity data to it. The data was collected during three weeks. The average values during the last 24 hours has been calculated over the input signal. The membership functions for defining when a temperature or luminosity is 'high', 'medium' or 'low' has been defined automatically based on the range in input values (approximately 35% of the samples are defined as 'high', 30% as 'medium', and 35% 'low').

The activation of the states is evolving in function of the current inputs, as well as on the basis of the current simulated emotional state. An example showing the evolution of the different states is given in figure 5.

The system started in a depressed state due to the initial state activations settings, (1,0,0). However, the input to the system did not provide any reason for being in this state, and quickly the system moves from a mainly neutral state towards a joyful state. This state continues for two days, after which the time restrictions for state 3 start to take effect, lowering the state activation. At the same time the temperature and luminosity start to fall, and during day 2-3 the conditions are sufficiently bad to give rise to a depressed state. But later in the week conditions start to improve again, and a neutral state is recovered, followed by a happy state, due to the improved weather conditions.

The speech output at a particular point in time was annotated based on the nearest prototypical state's annotation, as explained at the end of section 3, and displayed in table 2.

Currently the prototype is visualized by an icon on a computer screen with a button next to it. When the button is pushed a voice is heard speaking 'hello world' with different emotional intonations.

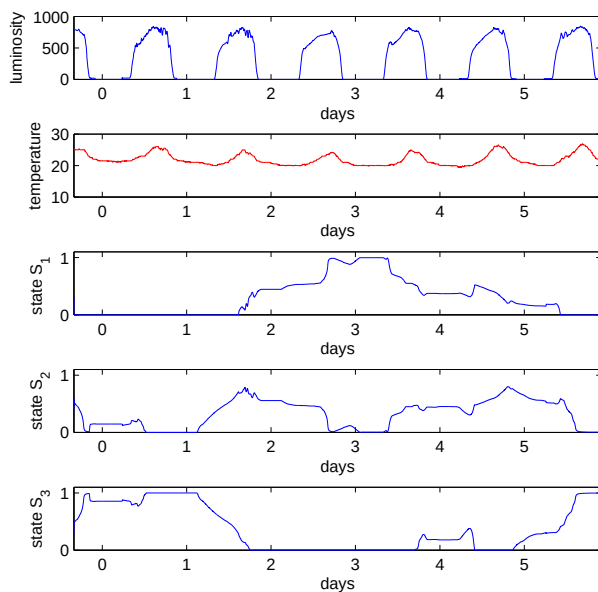


Figure 5: The activity of three states S_1, S_2 and S_3 , as well as the input variables *luminosity* and *temperature*.

5 CONCLUSIONS

We have presented a simple prototype as demonstration of how to create a virtual being that seems to have emotional status and that behaves in agreement with these emotions.

Table 2: Different annotations of the words in a sentence based on the emotional state.

	"and	have	a	good	day"
depressed					
pitch mean and contour		-10%	-20%	-30%	-40%
pitch range	-20%	-20%	-20%	-20%	-20%
intensity	-30%	-30%	-30%	-30%	-30%
speech rate	+20%	+20%	+20%	+20%	+20%
depressed / neutral					
pitch mean and contour		-5%	-10%	-15%	-20%
pitch range	-10%	-10%	-10%	-10%	-10%
intensity	-15%	-15%	-15%	-15%	-15%
speech rate	+10%	+10%	+10%	+10%	+10%
joyful / neutral					
pitch mean and contour		+5%	+10%	+15%	+20%
pitch range	+10%	+10%	+10%	+10%	+10%
intensity	-15%	-15%	-15%	-15%	-15%
speech rate	-15%	-15%	-15%	-15%	-15%
joyful					
pitch mean and contour		+10%	+20%	+30%	+40%
pitch range	+20%	+20%	+20%	+20%	+20%
intensity					
speech rate	-30%	-30%	-30%	-30%	-30%

In this paper we have contributed to the development of new techniques of human - computer communication. A FFSM has been used to create a model of different simulated emotional states. The computer produces the same spoken utterances with varying emotional content, adding an additional connotation to it's content. The emotional content can be expressed using SSML-like XML annotations.

This prototype is an additional step in the project of building a complex virtual being. This virtual being will be the personal assistant of the user. As humans do, the computerized assistant will be able to communicate information about it's context and emotional state. Of course, to take advantage of this possibility, the emotional states must be influenced by the variables in the context in such a way that the human user will realize the content is modified. For example, the tone of the voice could emphasize a message of warning or the communication of a positive result.

Acknowledgements

This work has been partially funded by the Spanish government (CICYT), under grant: TIN2008-06890-

C02-01, and by the Foundation for the Advancement of Soft Computing.

References

- [1] A. Alvarez and G. Trivino. Comprehensible model of a quasi-periodic signal. In *Proc. of the 9th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2009.
- [2] P. Boula de Mareüil, P. Celerier, and J. Toen. Generation of emotions by a morphing technique in english, french and spanish. In *Proceedings on Speech Prosody 2002*, pages 187 – 190, 2002.
- [3] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, and W. Fellenz. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 1:32 – 80, 2001.
- [4] A.R. Damasio. *Descartes' Error: Emotion, Reason, and the Human Brain*. G.P. Putnam, New York, 1994.
- [5] E. M. Eisman, V. López, and J. L. Castro. Controlling the emotional state of an embodied conversational agent with a dynamic probabilistic fuzzy rules based system. *Expert Systems with Applications*, 36:9698 – 9708, 2009.
- [6] P. Ekman. An argument for basic emotions. *Cognition and Emotion*, 6:169 – 200, 1992.
- [7] M.S. El-Nasr, J. Yen, and Ioerger T.R. Flame fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, 3(3):219 – 257, 2000.
- [8] N. Fragopanagos and J. G. Taylor. Emotion recognition in human-computer interaction. *Neural Networks*, 18(4):389 – 405, 2005.
- [9] I. Iriondo, R. Guaus, A. Rodriguez, P. Lázaro, N. Montoya, J.M. Blanco, D. Bernadas, J.M. Oliver, D. Tena, and L. Longth. Validation of an acoustical modelling of emotional expression in spanish using speech synthesis techniques. In *Proceedings ISCA 2000*, pages 161 – 166, 2000.
- [10] J.M. Montero, J. Gutierrez-Arriola, J. Colas, E. Enriquez, and J.M. Pardo. Analysis and modelling of emotional speech in spanish. In *Proceedings of the 14th International Conference on Phonetic*, pages 957 – 960, 1999.
- [11] A. Ortony, G.L. Clore, and A. Collins. *The cognitive structure of emotion*. Cambridge University Press, Cambridge, UK, 1998.
- [12] M. Schröder. Emotional speech synthesis - a review. In *Proceedings Eurospeech 2001*, volume 1, pages 561 – 564, Aalborg, 2001.
- [13] M. Schröder. *Speech and Emotion Research: An overview of research frameworks and a dimensional approach to emotional speech synthesis*. PhD thesis, Saarland University, 2004.
- [14] M. Sugeno. *Industrial Applications of Fuzzy Control*. Elsevier Science Inc., New York, NY, USA, 1985.
- [15] Verbio Speech Technologies. Read aloud. <http://www.verbio.com/>.
- [16] G. Trivino and A. van der Heide. An experiment on the description of sequences of fuzzy perceptions. *Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS2008), September 10-12th, Barcelona, Spain*, 2008.
- [17] G. Trivino and A. van der Heide. Linguistic summarization of the human activity using skin conductivity and accelerometers. *Proceedings of the Conference Information Processing and Management of Uncertainty in Knowledge Based Systems. (IPMU2008), June 22-27, Malaga, Spain*, 2008.
- [18] D. Ververidisa and C. Kotropoulos. Emotional speech recognition: Resources, features, and methods speech communication. 48(9):1162 – 1181, 2006.
- [19] World Wide Web Consortium (W3C). Speech synthesis markup language 1.0. <http://www.w3.org/TR/speech-synthesis/>.
- [20] I. Wilson. The artificial emotion engine (tm), driving emotional behavior. *AAAI Technical Report*, pages 76 – 80, 2000.
- [21] Lotfi A. Zadeh. The concept of linguistic variable and its application to approximate reasoning. *Information sciences*, 1975.

DETECCIÓN DE IDIOMA DE SITIOS WEB MEDIANTE REDES NEURONALES

Piedad Garrido¹ Francisco J. Martínez¹ Francisco J. Vela² Jesus Tramullas¹ Inmaculada Plaza¹

¹ Universidad de Zaragoza, {piedad, f.martinez, tramullas, iplaza}@unizar.es

² Universidad de Zaragoza, fjvela@gmail.com

Resumen

La detección del idioma de un documento puede tener especial importancia, sobretodo en entornos donde se trabaja con grandes volúmenes de documentos escritos en diferentes idiomas y que se desean clasificar. Normalmente esta detección se realizaba o de forma manual, o usando métodos estadísticos con computadores. En este artículo se presenta un proyecto que hemos realizado que permite identificar de forma automática el idioma de las páginas web, usando una nueva metodología basada en redes neuronales. Ha sido necesario desarrollar tres aplicaciones: (i) la primera ayuda a la creación, entrenamiento, proyección y visualización de redes neuronales, (ii) la segunda recoge y ajusta los datos, y (iii) la tercera sirve para comprobar si la red neuronal está bien entrenada, hasta alcanzar una tasa de fallos que pueda ser asumida. Los resultados demuestran que el uso de esta metodología da muy buenos resultados con páginas web de diferentes idiomas.

Palabras Clave: Internet, detección de idioma, Redes Neuronales.

1 INTRODUCCIÓN

La identificación del idioma es el proceso por el cual se determina cuál es el idioma en el que está escrito un texto dado. Tradicionalmente, la identificación del idioma se ha llevado a cabo en textos escritos, de forma manual y en entornos de bibliotecas. Se realizaba identificando palabras que aparecían con frecuencia, o identificando letras y signos de puntuación que son característicos en ciertas lenguas.

Desde hace un tiempo a esta parte, se han incorporado los llamados métodos computacionales (en los que se hace uso de los ordenadores) para resolver el problema, usando métodos de procesamiento de lenguaje natural basados en métodos estadísticos. De esta forma, se pueden clasificar multitud de documentos en función del idioma de una forma más rápida, barata y eficiente. Además, con la aparición de los documentos electrónicos, la detección automática del idioma ha pasado a tener una mayor importancia, al no estar sólo enmarcado en el ámbito bibliotecario, sino también a cualquier organización que tenga que trabajar con grandes volúmenes de documentos escritos en diferentes idiomas.

Desde hace tiempo, los científicos han estudiado el funcionamiento del cerebro, intentando desarrollar algunos modelos matemáticos que tratan de simular su comportamiento, ya que es capaz de procesar a gran velocidad grandes volúmenes de información, combinarla o compararla con información almacenada y dar respuestas adecuadas. Estos modelos se han basado en los estudios de las características esenciales de las neuronas y sus conexiones.

Las redes de neuronas artificiales (denominadas habitualmente como RNA o ANN, en inglés) [5] son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Este paradigma nos permite afrontar problemas de difícil solución con la programación tradicional. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Esta salida viene dada por tres funciones: (i) una función de propagación (también conocida como función de excitación), (ii) una función de activación, que modifica a la anterior, y (iii) una función de transferencia, que se aplica al valor devuelto por la función de activación y se utiliza para acotar la salida de la neurona.

Tabla 1: Aplicaciones para detectar idiomas.

Nombre	Tipo	Lenguaje de desarrollo	Licencia
Word	Aplicación	C++	Propietario
Lingua-Identify	Librería	Perl	Libre
TextCat	Librería	Perl	Libre
Rosette Language Identifier	Aplicación	No determinado	Propietario
Petamem Language Identifier	Aplicación	No determinado	Propietario
lid	Librería	C/C++	Propietario
Talcnknobbel	Aplicación	No determinado	Propietario

En este artículo, se presenta el proyecto que hemos realizado que permite identificar de forma automática el idioma de las páginas web, usando un algoritmo basado en redes neuronales. Este documento está organizado de la siguiente manera: en la Sección 2 se presentan las diferentes aplicaciones y métodos que normalmente son utilizados para la detección del idioma tanto en los documentos textuales, como en las páginas web. La Sección 3 presenta el software desarrollado, así como una breve descripción de las diferentes aplicaciones por las que está compuesto. Los resultados obtenidos se detallan en la Sección 4. Finalmente, la Sección 5 presenta las conclusiones más importantes y el trabajo futuro.

2 DETECCIÓN DEL IDIOMA

Existen diferentes aplicaciones para detectar el idioma de un texto escrito. Por un lado, se puede trabajar con la funcionalidad de una aplicación que use diccionarios ortográficos y gramaticales, reglas de puntuación, además de otros parámetros externos como el idioma del teclado, etc. Un ejemplo sería un procesador de textos como por ejemplo Microsoft Word. Por otro lado, se puede hacer uso de una aplicación externa para identificar el idioma del texto. Ejemplos de esta última forma son: (i) Lingua-Identify [13], (ii) TextCat [9], (iii) lid [7], (iv) PetaMem [12], (v) Rosette [1] y (vi) Talcnknobbel [14].

En la Tabla 1 aparecen algunas de las aplicaciones más conocidas. Este tipo de aplicaciones suelen utilizar algunas técnicas entre las que destacamos:

- Técnica de palabras pequeñas. Esta técnica consiste en ir buscando dentro del texto las palabras más comunes de una lengua, como los pronombres, los artículos, etc. Es la opción más recomendada para textos grandes.
- Análisis de Prefijos. Analiza los prefijos más comunes de los idiomas.

- Análisis de Sufijos. Analiza los sufijos más comunes de los idiomas.

- Categorización Ngram, donde n representa las subsecuencias de n elementos. Dada una secuencia de letras se construyen las posibles palabras y se comparan.

A diferencia de la detección de idiomas en textos, para la detección de idiomas de sitios web, se habla de métodos y no de aplicaciones, ya que los buscadores aplican una serie de técnicas para comprobar el idioma de un sitio web y así indexarlo en su sistema.

En la Tabla 2 se muestran las técnicas que tradicionalmente se usan, haciendo especial hincapié en sus ventajas e inconvenientes. De todas ellas, la técnica más usada por los buscadores es la de los diccionarios de palabras.

Cabe destacar, que ninguna de esas técnicas parece lo suficientemente buena, pues todas presentan ciertos inconvenientes, que en algunos casos pueden ser verdaderamente críticos.

Por otro lado, Google ha puesto a disposición de los usuarios dos productos: (i) una API desarrollada en AJAX con la que acceder mediante JavaScript a un conjunto de herramientas de traducción, detección y transliteración de idiomas [3], y (ii) el Google Translator Toolkit [4] un servicio que ofrece un conjunto de herramientas para facilitar las traducciones, permitiendo el trabajo colaborativo, compartir glosarios, etc.

A diferencia de las propuestas existentes, este trabajo presenta un conjunto de aplicaciones que permiten detectar el idioma de las páginas web haciendo uso de una red neuronal basada en los mapas autoorganizados de Kohonen [6], un sistema mucho más cómodo, fácil de mantener y válido, tanto para trabajar con información textual como con información ubicada en sitios web.

Tabla 2: Técnicas de detección de idiomas usadas en Internet.

Técnicas	Ventajas	Inconvenientes
Cabeceras páginas web	Identificación rápida	Escasa utilización, intervención de las personas
Extensiones del dominio	Identificación rápida	Páginas multiidioma, dominios genéricos (.com, ...)
Direcciones IP	Identificación rápida	Almacenamiento en hosting extranjeros
Diccionarios de palabras	Gran número de aciertos, organización de las páginas en subdominios	Coste temporal

3 DESCRIPCIÓN DE LA APLICACIÓN

Este apartado presenta el proyecto desarrollado para la detección de rasgos de páginas web mediante redes neuronales. La arquitectura software escogida es una arquitectura cliente/servidor de tres capas, que facilita la reducción de tráfico en la red, requiere baja potencia de procesamiento por parte del cliente, y un bajo coste de desarrollo y mantenimiento.

Definidos y desarrollados los distintos escenarios, se identifican y se organizan las clases relevantes del sistema mediante el diagrama de clases (ver Figura 1), donde se muestra la relación de los objetos modelados que componen cada una de las aplicaciones, describiendo sus características, estructura y comportamiento.

El proyecto desarrollado está compuesto de tres aplicaciones desarrolladas con software libre: (i) la aplicación *Self-Organizing Maps Framework* (SOMF), (ii) la aplicación *Detección de Rasgos de Páginas Web* (DRPW), y (iii) una aplicación que permite recoger direcciones web en diferentes idiomas para utilizarlas como ficheros de entrada de datos. A continuación se presentan más en detalle.

3.1 Aplicación SOMF

El principal objetivo de la aplicación SOMF es el de proporcionar una herramienta software, que facilite el manejo de redes neuronales utilizando como base el paquete SOM-PAK (Self-Organizing Maps Program Package) [8].

La aplicación SOMF consta de cuatro operaciones principales: (i) la opción operaciones de red, donde el usuario podrá crear, entrenar y visualizar una red en formato SVG, (ii) la opción operaciones de ficheros, donde el usuario podrá visualizar los diferentes archivos generados por la aplicación, ya sean

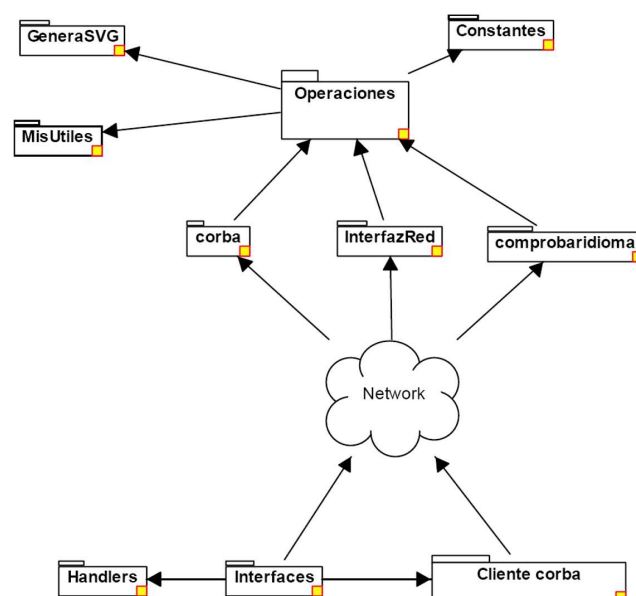


Figura 1: Paquetes que componen la aplicación.

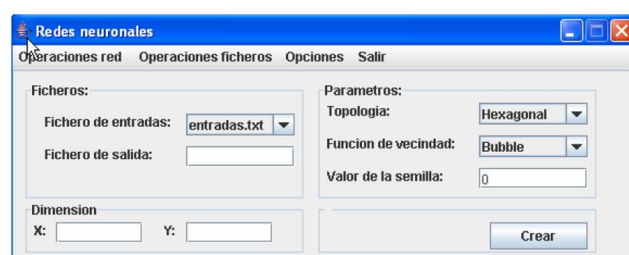


Figura 2: Interfaz de la aplicación SOMF. Crear una red neuronal.

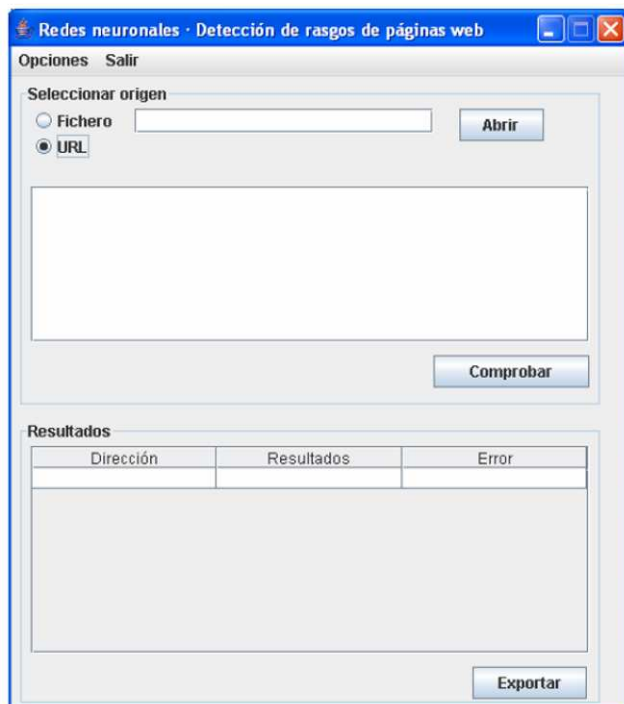


Figura 3: Interfaz de la aplicación DRPW. Pantalla principal.

de texto o visuales, (iii) el apartado de opciones, donde se pueden configurar aspectos tales como la sincronía, la configuración de parámetros de conexión, etc., y (iv) la opción que nos permite salir de la aplicación de forma correcta cerrando todos los hilos de ejecución. La Figura 2, presenta un ejemplo de la interfaz.

3.2 Aplicación DRPW

El principal objetivo de la aplicación DRPW, es el de proporcionar una herramienta software que facilite la identificación del idioma de una página web a través de una red neuronal. A través de la interfaz de esta aplicación (ver Figura 3) el usuario podrá introducir una o varias direcciones URL para que la red neuronal detecte el idioma de la página web. Estas direcciones se podrán introducir de dos maneras diferentes: desde un fichero generado previamente por el usuario, o escribiéndola directamente en la interfaz de la aplicación. Una vez especificada esta información, el programa automáticamente mostrará los resultados al usuario dando la posibilidad de exportar estos resultados a un fichero con formato CSV (Comma Separated Value File Format) [2].

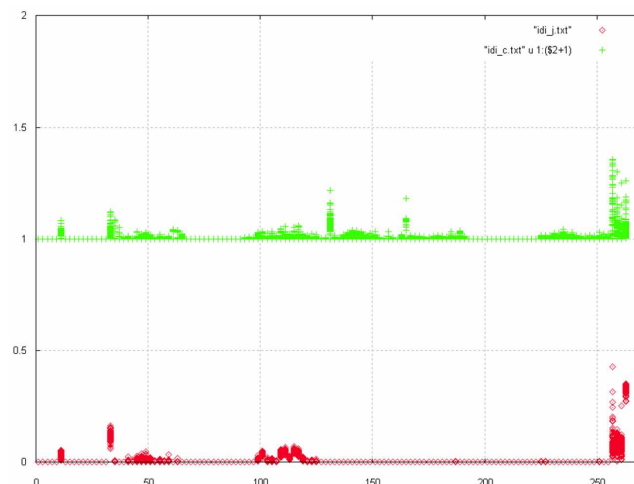


Figura 4: Frecuencia de aparición de los caracteres en castellano (en verde) y japonés (en rojo).

3.3 Aplicación de Recogida de URLs

El principal objetivo de la aplicación de recogida de URLs es el de encargarse de recoger direcciones web en diferentes idiomas, para después poder generar los ficheros de entrada y ahorrarle al usuario la tediosa tarea de ir introduciendo las direcciones, una a una, a través de la interfaz.

En la Tabla 3 se muestra dónde estarían ubicados en la arquitectura Cliente/Servidor, los componentes que forman cada una de las aplicaciones desarrolladas y detalladas con anterioridad.

4 RESULTADOS

Para poder realizar las pruebas, en primer lugar ha sido necesario construir un fichero para poder entrenar la red neuronal. En nuestro caso utilizamos cien páginas por cada idioma. Juntando las estadísticas obtenidas con dos idiomas (castellano y japonés) para el primer juego de pruebas, construimos el fichero de entrada. En la Figura 4 se muestra gráficamente la frecuencia normalizada de aparición de los distintos caracteres en las páginas en castellano (en verde) y las páginas en japonés (en rojo).

En la Tabla 4 aparecen los valores introducidos para ajustar los parámetros de configuración, con la finalidad de establecer un juego de pruebas y entrenamientos válidos, cuyos resultados quedan reflejados en la Tabla 5.

Tras varios entrenamientos, se detectaron algunos problemas a resolver:

Tabla 3: Ubicación de las aplicaciones.

Aplicación	Localización	Función
Interfaz DRPW	Cliente	Se ocupa de interactuar con el usuario y de comunicarse con el servidor DRPW
Interfaz SOM	Cliente	Se encarga de interactuar con el usuario y de comunicarse con el servidor SOMF
Interfaz DRPW	Servidor	Se ocupa de atender las operaciones solicitadas a través de la interfaz DRPW (lógica de la aplicación, manejo de redes SOMF)
Interfaz SOM	Servidor	Se ocupa de atender las operaciones solicitadas a través de la interfaz SOMF (lógica de la aplicación, calcula las estadísticas y proyecta la nueva entrada sobre una red)
Recogida URLs	Servidor	Esta aplicación es la encargada de recoger direcciones web en diferentes idiomas para después poder generar los ficheros de estadísticas
Cálculo de estadísticas	Servidor	A través de este grupo de aplicaciones se generan los diferentes archivos necesarios para el entrenamiento de la red neuronal

- Si no se hace nada para evitarlo, se tienen en cuenta caracteres que no son representativos para la detección del idioma de una página web. Estos caracteres son el salto de línea y el salto de párrafo. Tan sólo están en el texto para mejorar el aspecto de la página, por lo que deben ser eliminados.
- Tal y como se observa en la Figura 4, existe un gran número de caracteres que tienen unos valores muy pequeños por lo que la red no los estará teniendo en cuenta, ya que no conseguirá diferenciarlos. Para ello, aplicamos el logaritmo a los valores para que la red sea capaz de distinguirlos y así aumentar los aciertos.
- Gracias a un proceso de refinado de parámetros, se han ido obteniendo resultados cada vez más precisos comparando distintas lenguas romances, germánicas y aglutinantes (japonés), consiguiendo un porcentaje de aciertos que supera el 90%.

5 CONCLUSIONES Y TRABAJO FUTURO

Para la realización de este trabajo se han programado en Java los algoritmos que permiten la detección del idioma de una página web a través de una red neuronal. Una vez concluida la fase de desarrollo de las aplicaciones necesarias para la captura de direcciones url, el cálculo de estadísticas, la aplicación gráfica donde se visualiza el entrenamiento y los resultados, la aplicación DRPW y la aplicación SOMF, se entrenó la red neuronal con diferentes parámetros para buscar los

Tabla 4: Parámetros usados para realizar las pruebas.

Idiomas estudiados	Castellano, Japonés Francés, Alemán
Dimensión de la red (X)	5, 6, 10
Dimensión de la red (Y)	5, 6, 10
Valor de la semilla	valores aleatorios
Función de vecindad	Bubble
Topología	Hexa
Número de iteraciones	1000, 6000 10000, 15000
Factor de aprendizaje	0.02, 0.05
Radio de aprendizaje	2, 3, 4, 5, 8

más adecuados, consiguiendo un porcentaje de aciertos mayor del 90%.

El presente proyecto ha sido desarrollado para su aplicación en un modelo de calidad orientado a la evaluación de software educativo [10], así como la accesibilidad y usabilidad de sitios web educativos [11]. Debido a los buenos resultados obtenidos en esta primera fase de desarrollo del proyecto, la principal línea de trabajo futuro es realizar una mejora a esta aproximación funcional del soft-computing, basada en redes neuronales, para su adaptación a un sistema híbrido basado en lógica fuzzy que permita predecir los valores de los parámetros de configuración, poder trabajar con más tipos de idiomas, conseguir separar la capa de presentación de los sitios web de la del contenido en aquellos casos que sea imprecisa, para así evitar tener en cuenta caracteres que no sean representativos y facilitar la detección de valores pequeños que sean relevantes y deban ser tenidos en cuenta.

Tabla 5: Resultados obtenidos en las pruebas.

Prueba 1	
Aciertos	Japonés: 58.0882% Castellano: 98.1595% Francés: 97.1014%
	Total: 85.3547%
Fallos	Japonés: 41.9118% Castellano: 1.8405% Francés: 2.8986%
	Total: 14.6453%
Prueba 2	
Aciertos	Japonés: 97.9381% Castellano: 100.0000%
	Total: 98.9691%
Fallos	Japonés: 2.0619% Castellano: 0.0000%
	Total: 1.0309%
Prueba 3	
Aciertos	Alemán: 98.3051% Castellano: 96.6387% Francés: 91.5966%
	Total: 95.5056%
Fallos	Alemán: 1.6949% Castellano: 3.3613% Francés: 8.4034%
	Total: 4.4944%

Referencias

- [1] Basis Technology (2009) Rosette Language Identifier. [Online]. Available: <http://www.basistech.com/language-identification/>
- [2] Creativyst Software. (2007) The Comma Separated Value (CSV) File Format. [Online]. Available: <http://www.creativyst.com/Doc/Articles/CSV/CSV01.htm>
- [3] Google. (2009) Google AJAX Language API. [Online]. Available: <http://www.google.com/uds/samples/language/detect.html>
- [4] Google. (2009) Google Translation Toolkit. [Online]. Available: <http://translate.google.com/toolkit/>
- [5] L. D. Jackel, R. E. Howard, H. P. Graf, B. Straughn, J. S. Denker. Artificial neural networks for computing. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, vol. 4, no. 1, pp. 61-63, 1986.
- [6] Kohonen, T. Exploration of very large databases by self-organizing maps. *International Conference on Neural Networks*, vol. 1, 1997.
- [7] Lingua Systems. (2009) lid Language Identifier. [Online]. Available: <http://www.lingua-systems.com/products.html>
- [8] Neural Networks Research Centre. (2009) SOM_PAK and LVQ_PAK. [Online]. Available: http://www.cis.hut.fi/research/som_lvq_pak.shtml
- [9] G. van Noord. (2009) TextCat. [Online]. Available: <http://www.let.rug.nl/~van Noord/TextCat/>
- [10] I. Plaza, R. Igual, J.J. Marcuello, S. Sánchez, F. Arcega. Proposal of a Quality Model for Educational Software. *XX Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE09)*, Valencia, España, junio 2009
- [11] I. Plaza, F. Naranjo, F. Arcega, P. Garrido, S. Castellote. Calidad en grupos universitarios de investigación como nexo de unión empresa-universidad-sociedad. *XVII Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas*, Valencia, España, septiembre 2009.
- [12] PetaMem (2009) Language identifier by Petamem. [Online]. Available: <http://nlp.petamem.com/langident.cgi>
- [13] A. Simoes. (2009) Lingua-Identify. [Online]. Available: <http://search.cpan.org/~ambs/Lingua-Identify-0.23/lib/Lingua/Identify.pm>.
- [14] (2009) Talenknobbel. [Online]. Available: <http://www.fuzzums.nl/~joost/talenknobbel/>

SISTEMA FUZZY APLICADO AL CAGD BASADO EN RELACIONES DE SIMILITUD

Amadeu Monreal Jordi Recasens Joan Jacas

Sec. Matemàtiques i Informàtica. E.T.S. d'Arquitectura de Barcelona, { amadeu.monreal, j.recasens, joan.jacas }@upc.edu

Resumen

Este artículo presenta una primera aproximación a un sistema fuzzy para el **diseño** de curvas y superficies en el contexto de del Diseño Geométrico Asistido por Ordenador (CAGD). Mediante el uso de relaciones de similitud, se estudian algunos problemas que presentaba un modelo anterior.

Palabras Clave: Relación de similitud, número fuzzy, CADG, Razonamiento aproximado, control fuzzy.

métodos de defuzzificación conocidos como, por ejemplo, el del centro de gravedad. El contenido de este trabajo está ordenado de la siguiente manera. Después de esta introducción, la sección 2 contiene aquellas definiciones y conceptos relacionados con relaciones de similitud y clases borrosas necesarias para la comprensión del trabajo. Las secciones 3 y 4 presentan el modelo general y un modelo asociado al procedimiento de Takagi-Sugeno que trata de superar algunos de los problemas que presentaba nuestra primera propuesta contenida en [16], como por ejemplo la construcción efectiva de los números borrosos asociados al sistema o la solución para el caso de datos dispersos. En la última sección desarrollamos completamente un ejemplo.

1 INTRODUCCIÓN¹

El diseño geométrico asistido por ordenador (CAGD) tiene como finalidad la producción de curvas o superficies cuya forma está controlada por un conjunto de puntos. Puede entenderse como un cierto tipo de procedimiento interpolativo cuyo objetivo no es la generación de una función sino la obtención de su representación gráfica. Por ello en el diseño del procedimiento es esencial el control de la suavidad de la solución. Por otra parte, el usuario de un sistema de CAGD aporta los datos "aproximadamente" ya que usualmente, después de una primera prueba, ajusta el resultado a su punto de vista estético. Por ello, parece natural considerar este problema dentro del área del razonamiento aproximado interpretando los datos de entrada-salida como puntos fuzzy de $\mathbb{R}$ o $\mathbb{R}^2$. Para la construcción de este sistema, centramos nuestra atención a los procesos de "fuzzificación" y "defuzzificación". Para definir los números fuzzy de entrada-salida, suponemos la existencia de una relación de similitud en $\mathbb{R}$ que da granularidad y fuzzifica sus elementos. Con respecto a la defuzzificación, podemos escoger el modelo de Takagi-Sugeno que produce una salida crisp o seleccionar alguno de los

2 PRELIMINARES

Un gran número de aplicaciones de la teoría de conjuntos fuzzy que han gozado de éxito, han generado numerosos conceptos generalmente basados en motivaciones de carácter empírico. Con el fin de dar una justificación teórica para el uso de estos conceptos aplicados, es necesario aportar en este contexto un modelo que permita darles una clara semántica. A este fin, esta sección se dedica a la introducción de una serie de conceptos que justifican las diferentes definiciones y tipos de números fuzzy tal y como se interpretan y utilizan en las aplicaciones.

Suponemos como punto de partida que el concepto de número fuzzy surge de la existencia de una cierta relación de equivalencia fuzzy en $\mathbb{R}$, en el sentido de que números próximos son indistinguibles para el observador [14]. Cuando la equivalencia definida es una relación de indistinguibilidad [7,19], los números borrosos triangulares y trapezoidales entre otros, aparecen de forma natural. El modelo justifica el uso de diferentes tipos de números dependiendo de la escala definida en $\mathbb{R}$ y desarrolla la generación de similitudes asociadas a una o varias escalas. Los números borrosos surgen de forma natural como los singletons asociados a la similitud subyacente [2,12,13]. De aquí que únicamente sean consideradas aquellas relaciones cuyos singletons cumplan una definición general de

¹Partially supported by the project TIN2006-14311

número fuzzy. Recíprocamente, dada una relación de similitud “conveniente” definida en $\mathbb{R}$, solo admitiremos como números fuzzy aquellos que sean singletons de la relación considerada. Como consecuencia natural de este enfoque, estudiamos aquellas familias de números fuzzy que son invariantes por traslaciones.

A lo largo de este artículo se utilizan las definiciones usuales sobre t-normas y sus propiedades. En el texto Π representa la t-norma producto y $\mathcal{L}$ la t-norma de Łukasiewicz. También se recuerdan algunas definiciones sobre T-similitudes y números fuzzy. Para una información más detallada sobre estos temas, el lector puede consultar [1,7,14,15,17].

Definición 2.1. Sea T una t-norma. Una *T-Similitud* E en un conjunto X es una relación fuzzy reflexiva y simétrica en X tal que

$$T(E(x, y), E(y, z)) \leq E(x, z) \text{ T-transitiva.}$$

Para $T = \mathcal{L}$, E se denomina “*likeness*” y para $T = \Pi$, E es una *relación probabilística*.

Las T-similitudes generalizan, en un entorno fuzzy, el concepto de relación de equivalencia y también algunos autores las denominan relaciones de igualdad fuzzy [2,7]. Desde un punto de vista semántico, $E(x, y)$ puede interpretarse como el grado de similitud entre x y y .

Definición 2.2. Dada una T-similitud E en un conjunto X y un elemento x_0 de X , el *singleton* inducido por x_0 es el conjunto fuzzy f_{x_0} definido por $f_{x_0}(x) = E(x_0, x)$.

El singleton de un elemento x de X puede interpretarse como la clase de equivalencia respecto de E [2] a la cual pertenece y también como su fuzzyficación respecto de E [8].

Definición 2.3. Un *número real fuzzy* es una aplicación $f_a : \mathbb{R} \rightarrow [0, 1]$ tal que existe un número $a \in \mathbb{R}$ con $f_a(a) = 1$, donde f_a es no-decreciente en $(-\infty, a)$ y no-creciente en $(a, +\infty)$.

Definición 2.4. Una T-similitud E en $\mathbb{R}$ se denomina *admisible* si y sólo si sus singletons son números reales fuzzy.

El fuzzificado de un número real por medio una T-similitud admisible es por lo tanto un número fuzzy.

A continuación estudiamos la relación entre las T-similitudes admisibles y ciertas familias de números fuzzy.

2.1 T-similitudes admisibles

noindent **Teorema 2.1.2.**[13] Sea $f : \mathbb{R} \rightarrow \mathbb{R}$ una función monótona y t un generador de una t-norma arquimediana T ,

$$E_f(x, y) = t^{[-1]}(|f(x) - f(y)|)$$

es una T-similitud admisible.

Dos funciones monótonas f, g generan la misma T-similitud si y sólo si existe una constante k tal que $f = \pm g + k$.

En aplicaciones reales en las que diferentes expertos producen escalas diferentes en $\mathbb{R}$ es necesario combinar sus informaciones. El siguiente teorema muestra una manera de hacerlo.

Teorema 2.1.5. Si $\{E_i\}_{i \in I}$ es una familia de T-similitudes admisibles en $\mathbb{R}$, entonces

$$E = \inf_{i \in I} E_i$$

es también una t-similitud admisible en $\mathbb{R}$.

Por lo tanto, podemos combinar “likeness” o relaciones probabilísticas generadas por funciones monótonas para obtener otra relación del mismo tipo.

2.2 Números fuzzy invariantes por traslaciones

Una tipo especial de números fuzzy lo constituye la familia de los que son invariantes por traslaciones. Grosso modo, familias formadas por números fuzzy que tienen la misma “forma”. Del mismo modo, podemos estudiar T-similitudes invariantes por traslaciones i.e.: el grado de similitud $E(x, y)$ entre dos números x, y depende únicamente de su distancia.

En esta sección, se estudia la relación entre familias de números fuzzy y familias de T-similitudes invariantes por traslaciones.

Definición 2.2.1. Una familia $\{f_a\}_{a \in \mathbb{R}}$ de números fuzzy asociada a una T-similitud admisible es *invariante por traslaciones* si y sólo si

$$f_a(x) = f_b(x - a + b).$$

Una t-similitud E es *invariante por traslaciones* si y sólo si

$$E(x + a, y + a) = E(x, y).$$

Veamos cuales son las condiciones que una familia $\{f_a\}_{a \in \mathbb{R}}$ debe cumplir para que sea el conjunto de los singletons asociado a una T-similitud, para una t-norma arquimediana.

Teorema 2.2.4.[7] Sea T una t -norma arquimediana y t uno de sus generadores aditivos, la familia $\{f_a\}_{a \in \mathbb{R}}$ de números fuzzy invariantes por translaciones $\{f_a\}_{a \in \mathbb{R}}$ formada por los singletons de una T -similitud sobre $\mathbb{R}$ se obtiene a través de la función f_0 (mediante la definición 2.2.1) definida por

$$f_0 = t^{[-1]} \circ F$$

siendo F una función par, subaditiva y no decreciente en $\mathbb{R}^+$ con $F(0) = 0$.

La T -similitud E asociada se define de la siguiente forma:

$$E(x, y) = t^{[-1]} \circ F(y - x)$$

que puede comprobarse trivialmente que es invariante por translaciones.

En esta sección, se ha introducido una metodología general para generar números fuzzy asociados a T -similitudes. Esta metodología permite la construcción efectiva de familias de números fuzzy apropiados para aplicaciones ya que las T -similitudes subyacentes pueden deducirse de escalas propuestas por expertos adaptadas especialmente a una aplicación concreta.

3 CONSTRUCCIÓN DE FIGURAS EN CAGD

Para generar una curva o superficie mediante un sistema de dibujo asistido por ordenador, el usuario suele suministrar al sistema un conjunto de puntos y este produce una curva o superficie cuya forma está controlada por los datos suministrados. En general no produce una figura interpoladora desde un punto de vista clásico ya que ello implica que la curva o superficie contenga los datos. En nuestra propuesta consideramos el problema en una forma menos rígida en el sentido de que los valores de la curva o superficie se obtienen a partir de los datos sin imponer la condición interpoladora respondiendo así al verdadero significado del proceso de diseño tal y como se ha explicado en la introducción. Por otra parte, siendo un proceso de diseño, es esencial el control de la suavidad de la solución.

En esta comunicación proponemos la solución de los siguientes problemas:

Problema 1a) (*Curvas funcionales*). Dados $n + 1$ puntos de control $\vec{P}_i = (a_i, b_i) \in \mathbb{R}^2$, $i = 0, 1, \dots, n$, con $a_0 < a_1 < \dots < a_n$, se desea construir la gráfica de una función $f : [a_0, a_n] \rightarrow \mathbb{R}$ tal que $\{a_i\}$ son los valores de la variable independiente y $\{b_i\}$ son los valores o aproximaciones de $\{f(a_i)\}$.

Problema 1b) (*Curvas parametrizadas*). Dados $n + 1$ puntos de control $\vec{P}_i = (a_i, b_i) \in \mathbb{R}^2$, o $\vec{P}_i = (a_i, b_i, c_i) \in \mathbb{R}^3$,

$i = 0, 1, \dots, n$ se busca la construcción de $n + 1$ valores $t_0 < t_1 < \dots < t_n$ de un parámetro y una curva parametrizada $\vec{f} : [t_0, t_n] \rightarrow \mathbb{R}^2$ o $\mathbb{R}^3$ tal que los puntos $\{\vec{P}_i\}$ son los valores o aproximaciones de $\{\vec{f}(t_i)\}$.

Problema 2a) (*Superficies funcionales*). Dados $n + 1$ puntos de control distribuidos arbitrariamente $\vec{P}_i = (a_i, b_i, c_i) \in \mathbb{R}^3$, $i = 0, 1, \dots, n$, se busca la construcción de una superficie que contenga o esté controlada por los puntos dados que sea la gráfica de una función $f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ (por ejemplo, $A = [a_{\min}, a_{\max}] \times [b_{\min}, b_{\max}]$) tal que $\{(a_i, b_i)\}$ son los valores de la variable independiente y $\{c_i\}$ valores o aproximaciones de $\{f(a_i, b_i)\}$.

Problema 2b) (*Superficies parametrizadas*). Dada una parrilla de $(n + 1) \times (m + 1)$ puntos $\vec{P}_{ij} = (a_{ij}, b_{ij}, c_{ij}) \in \mathbb{R}^3$ $i = 0, \dots, n$, $j = 0, \dots, m$, queremos construir $n + 1$ valores $s_0 < s_1 < \dots < s_n$ de un parámetro s y $m + 1$ valores $t_0 < t_1 < \dots < t_m$ de un parámetro t y una superficie parametrizada $\vec{f} : [s_0, s_n] \times [t_0, t_m] \rightarrow \mathbb{R}^3$ de tal manera que los puntos $\{\vec{P}_{ij}\}$ son valores o aproximaciones de $\{\vec{f}(s_i, t_j)\}$.

Existen múltiples técnicas para resolver los problemas propuestos. Para una información más completa sobre ellas así como sobre los métodos para calcular los valores de los parámetros, recomendamos al lector las referencias [9,10,11].

El algoritmo común a todos los métodos de diseño que hace de interficie entre el procedimiento y el usuario se resume en la figura 1.

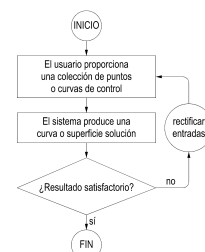


Figura 1: algoritmo para el diseño

El algoritmo muestra la vaguedad intrínseca en los datos y su relación con las curvas objetivo. Por lo tanto, parece natural situar este problema dentro del contexto del Razonamiento Aproximado. El modelo propuesto en la sección 4 utiliza los números fuzzy y la técnica de Takagi-Sugeno proveniente del control fuzzy para diseñar curvas y superficies.

4 El Modelo Takagi-Sugeno

Proponemos un sistema basado en reglas borrosas similar al usado en control fuzzy [5,17]. Se ha optado por la utilización de este método puesto que evita utilizar un proceso

de defuzzificación [21].

La idea central consiste en utilizar reglas del tipo:

Si $\vec{x}$ está **próximo** a $\vec{x}_i$ en el dominio, entonces $\vec{y}$ está **próximo** a $\vec{y}_i$ en la imagen. Esta idea se desarrolla en el siguiente **modelo**:

- Utilizar $n + 1$ reglas

R_i : Si $\vec{x} \simeq \vec{x}_i$, entonces $\vec{y} \simeq \vec{y}_i$; $i = 0, 1, \dots, n$.

- Modelar el grado de cumplimiento del antecedente de la i -ésima regla mediante el conjunto fuzzy α_i tal que $\alpha_i(\vec{x}_i) = 1$ y $\alpha_i(\vec{x})$ sea una función decreciente de la distancia a $\vec{x}_i$ [6].
- La función consecuencia de cada regla, $f_i(\vec{x})$, es la constante $\vec{y}_i$. (Es decir, el sistema utiliza el método de defuzzificación de **altura máxima**).
- La salida del sistema se obtiene como media ponderada de las salidas de cada regla cuyos pesos son los conjuntos fuzzy α_i :

$$\vec{y}(\vec{x}) = \frac{\sum_{i=0}^n \alpha_i(\vec{x}) \cdot \vec{y}_i}{\sum_{i=0}^n \alpha_i(\vec{x})} = \sum_{i=0}^n F_i(\vec{x}) \cdot \vec{y}_i \quad (2.1)$$

$$\text{donde } F_i(\vec{x}) = \frac{\alpha_i(\vec{x})}{\sum_{j=1}^n \alpha_j(\vec{x})}.$$

El resultado es una combinación lineal convexa de las salidas de las reglas cuyos coeficientes dependen de los datos y de los conjuntos fuzzy α_i . Ésta es la descripción general de nuestra propuesta que ya fue iniciada en [16].

Sin embargo, antes de seguir adelante es necesario formular una serie de observaciones así como poner de manifiesto los pros y los contras que inicialmente presenta el modelo para su implementación

4.1 Observaciones

4.1.1. Si el dominio está contenido en $\mathbb{R}$ (caso de una curva), entonces $\alpha_i = \mu_i$ son **números fuzzy**.

4.1.2. Si el dominio se halla en $\mathbb{R}^2$ (caso de una superficie), entonces la proximidad α_i de un punto $\vec{x} = (x, y)$ a $\vec{x}_i = (x_i, y_i)$ puede modelarse de la siguiente forma: $\vec{x} = (x, y)$ está **cerca** de $\vec{x}_i = (x_i, y_i)$ si y sólo si x está **cerca** de x_i y y está **cerca** de y_i . Es decir, si μ_i y ν_i son números fuzzy que modelan la proximidad de x y y a x_i y y_i respectivamente, entonces el conjunto fuzzy α_i se define mediante $\alpha_i(\vec{x}) = T(\mu_i(x), \nu_i(y))$, siendo T una t-norma convenientemente elegida que juega el papel semántico de una

conjunción. A fin de conservar el grado de suavidad de la superficie objetivo que ha heredado de la suavidad de los números fuzzy, es conveniente elegir una t-norma que sea C^∞ . En nuestro ejemplo hemos elegido la t-norma **producto**.

4.1.3. Los conjuntos fuzzy $\{\mu_i\}$ deben formar un recubrimiento del dominio en el sentido de que para cada punto $\vec{x}$ del dominio, debe existir un conjunto fuzzy μ_i con $\mu_i(\vec{x}) \neq 0$. Este requerimiento que inicialmente aparece como un requisito puramente técnico a fin de evitar una división por cero en la fórmula (2.1), puede interpretarse en el sentido de que cada punto del dominio debe ser controlado como mínimo por uno de los datos $\{\vec{x}_i\}$. Esta observación es fundamental a la hora de aplicar nuestro modelo cuando se tiene un conjunto disperso de datos.

4.1.4. Las curvas del tipo 1b) generadas por este modelo no pueden ser, simultáneamente, interpoladoras y suaves (i.e. con tangente continua). Efectivamente, si una curva $\vec{y}(t_i) = \vec{P}_i$ es interpoladora, $\mu_i(t_i) = 1$ y $\mu_j(t_i) = 0 \quad j \neq i$, pero debido a la monotonicidad de μ_i , $\mu_i(t) \equiv 0 \quad \forall t \notin (t_{i-1}, t_{i+1})$. Esto significa que para cada intervalo $[t_{i-1}, t_i]$ existen únicamente dos números fuzzy diferentes de cero y como consecuencia, dentro de este intervalo la curva es una combinación convexa de los puntos $\vec{P}_{i-1}$ and $\vec{P}_i$ dando como resultado una curva poligonal.

Sin embargo, no todo son inconvenientes

4.1.5. Observemos que, mientras que en los métodos clásicos utilizados en CAGD únicamente tiene importancia la elección de los puntos de control, en nuestro modelo también existe la posibilidad de elegir, dentro de un mismo conjunto de nodos, el tipo y el dominio del número fuzzy asociado a cada nodo como consecuencia de la T-similitud que los define. Este aspecto da una mayor versatilidad a nuestro modelo y nos permite solucionar algunos de los problemas presentados en estas observaciones.

De estas observaciones podemos deducir ciertos problemas que nos impiden obtener un procedimiento general.

El primer problema surge de la construcción efectiva de los números fuzzy asociados a los datos; es decir, el proceso de fuzzificación.

Podemos distinguir dos casos:

Si todos los nodos tienen el mismo peso en la construcción de la curva o superficie, parece natural aceptar que los puntos fuzzy son invariantes por translaciones y podemos aplicar el proceso explicado en la subsección 2.2.

En caso contrario, podemos utilizar escalas locales para ponderar la importancia de los puntos contenidos en los

distintos intervalos del dominio. En este sentido, el proceso de fuzzificación de $\mathbb{R}$ está plenamente justificado por las relaciones de igualdad fuzzy subyacentes.

Un segundo e importante problema es el relacionado con la observación 4.1.3, ya que nuestra proposición inicial no es aplicable cuando se tiene un conjunto disperso de datos. Este problema ha sido extensamente tratado en la literatura [8,10,18,19] desde diferentes puntos de vista. Incluso se pueden encontrar muchos artículos en donde se hacen comparaciones entre ellos [3,4,6,20]. A fin de superar este problema podemos distinguir dos situaciones:

- Los valores de la variable independiente son equidistantes. Para resolver este caso, se propone la construcción de una familia parametrizada de igualdades fuzzy en $\mathbb{R}$ tal que para cierto valor del parámetro, sus números fuzzy asociados recubran todo el dominio y que por lo tanto, para cada $\vec{x}_i$ de los datos existe un número fuzzy μ_i tal que $\mu_i(\vec{x}_j) = 1$ si $i = j$ y $\mu_i(\vec{x}_j) = 0$ en otro caso. Bajo esta hipótesis, se puede construir una curva que incluso interpola en el sentido clásico (véase el ejemplo de la sección 5).
- Los datos son dispersos con valores de la variable independiente no equidistantes. Nuestra propuesta consiste en la construcción de una función monótona f entre un subconjunto de números naturales y nuestro conjunto numerable (finito) de valores de la variable y mediante la equivalencia fuzzy $E(x_i, x_j) = t^{-1} \circ F(f^{-1}(x_i) - f^{-1}(x_j))$ obtener los números fuzzy asociados a los datos.

Bajo estos supuestos, proponemos las siguientes soluciones a los problemas propuestos:

4.2 Soluciones.

Problema 1a) (Curvas funcionales). En este caso, $\vec{P}_i = (a_i, b_i)$, el dominio es un cierto intervalo de $\mathbb{R}$, $\vec{x}_i = a_i$, $\vec{y}_i = b_i$ y $\alpha_i = \mu_i$ son números fuzzy normalizados. La curva solución es la gráfica de la función

$$y = f(x) = \frac{\sum_{i=0}^n \mu_i(x) \cdot b_i}{\sum_{i=0}^n \mu_i(x)} = \sum_{i=0}^n F_i(x) \cdot b_i. \quad (2.2)$$

Problema 1b) (Curvas parametrizadas). En este caso, $\vec{P}_i = (a_i, b_i) \in \mathbb{R}^2$ o $\vec{P}_i = (a_i, b_i, c_i) \in \mathbb{R}^3$. El dominio es un intervalo en $\mathbb{R}$, la variable es un parámetro t , $\vec{x}_i = t_i$, $\vec{y}_i = \vec{P}_i$ y $\alpha_i = \mu_i$ son números fuzzy normalizados. La curva solución es

$$\vec{y} = \vec{f}(t) = \frac{\sum_{i=0}^n \mu_i(t) \cdot \vec{P}_i}{\sum_{i=0}^n \mu_i(t)} = \sum_{i=0}^n F_i(t) \cdot \vec{P}_i. \quad (2.3)$$

Problem 2a). (Superficies funcionales). En este caso, $\vec{P}_i = (a_i, b_i, c_i) \in \mathbb{R}^3$, el dominio es una cierta región de $\mathbb{R}^2$, $\vec{x}_i = (a_i, b_i)$, $\vec{y}_i = c_i$ y $\alpha_i(x, y) = \mu_i(x) \cdot \nu_i(y)$ siendo μ_i, ν_i números fuzzy normalizados.

La superficie solución es la gráfica de la función

$$z = f(x, y) = \frac{\sum_{i=0}^n \alpha_i(x, y) \cdot c_i}{\sum_{i=0}^n \alpha_i(x, y)} = \sum_{i=0}^n F_i(x, y) \cdot c_i. \quad (2.4)$$

Problem 2b). (Superficies parametrizadas).

En este último caso, $\vec{P}_{ij} = (a_{ij}, b_{ij}, c_{ij})$, el dominio es un rectángulo de $\mathbb{R}^2$, las variables son los parámetros $s \in [s_0, s_n]$, $t \in [t_0, t_m]$, $\vec{x}_{ij} = (s_i, t_j)$, $\vec{y}_{ij} = \vec{P}_{ij}$ y $\alpha_{ij}(s, t) = \mu_i(s) \cdot \nu_j(t)$ siendo μ_i, ν_j números fuzzy normalizados.

La superficie solución es

$$\vec{P}(s, t) = \frac{\sum_{i=0}^n \sum_{j=0}^m \alpha_{ij}(s, t) \cdot \vec{P}_{ij}}{\sum_{i=0}^n \sum_{j=0}^m \alpha_{ij}(s, t)} = \sum_{i=0}^n \sum_{j=0}^m F_{ij}(s, t) \vec{P}_{ij} \quad (2.5)$$

donde

$$F_{ij}(s, t) = \frac{\alpha_{ij}(s, t)}{\sum_{i=0}^n \sum_{j=0}^m \alpha_{ij}(s, t)}$$

Nota: Puesto que en este caso, los datos forman una parrilla, podemos usar un doble índice para mayor comodidad. Si se desea utilizar un único índice k , se puede, por ejemplo, reasignar los índices de la siguiente manera: $\{i, j\} \rightarrow \{k\}$ con $k = i \cdot (m + 1) + j$.

5 UN EJEMPLO

En esta sección se desarrolla un ejemplo de curvas y superficies tanto en su forma funcional como paramétrica. Para la generación de la T-similitud subyacente en $\mathbb{R}$ invariante por translaciones, seleccionamos una función par, subaditiva y no decreciente en $\mathbb{R}^+$. En el ejemplo se ha elegido $F = \text{Arctan } |x|$ que cumple todas las condiciones del teorema 2.2.4. Por otra parte, se ha seleccionado una familia parametrizada de funciones estrictamente decrecientes $f(x, k) : [0, 1] \rightarrow \mathbb{R}^+$ tal que, juntamente con sus inversas (figura 2, nos permite la construcción de una familia de T-similitudes siendo T la t-norma representada por la figura 3 que también contiene una representación de la T-similitud asociada para $k=5$. Utilizando esta fuzzificación y el procedimiento explicado en la sección 4, generamos una curva funcional interpoladora con datos equidistantes.

La figura 4 presenta dos ejemplos de una curva parametrizada. Tal y como se puntualizó en las observa-

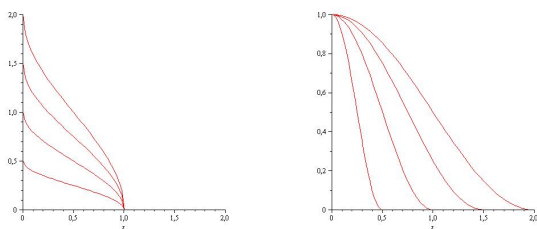


Figura 2: Generadores y sus inversos

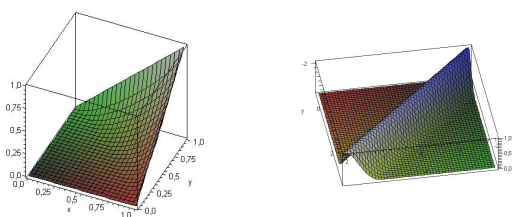


Figura 3: T-norma y T-similitud asociada

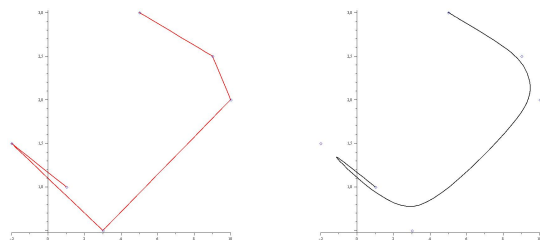


Figura 4: Curvas parametrizadas

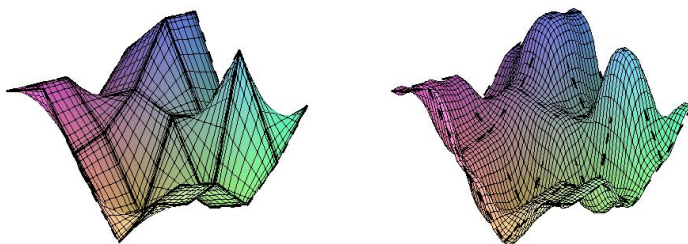


Figura 5: Superficies parametrizada interpoladora y superficie funcional interpoladora

ciones, estas curvas no pueden ser simultáneamente suaves e interpoladoras.

De forma análoga podemos construir superficies. Recordemos que, análogamente a lo que ocurre con curvas, se mantiene la incompatibilidad entre interpolación y suavidad para el caso paramétrico. La figura 5 muestra una superficie interpoladora para el caso paramétrico y una superficie funcional interpoladora.

Referencias

- [1] C. Alsina, On a family of connectives for fuzzy sets. *Fuzzy Sets and Systems* **16**, (1985) 231-235.
- [2] D. Boixader, J. Jacas, J. Recasens, Fuzzy equivalence relations: advanced material. In *Dubois and Prade Eds. Fundamentals of fuzzy sets*. (2000) Kluwer 261-290.
- [3] B. Bouchon-Meunier et al. A comparative view of interpolation methods between sparse fuzzy rules. *Proc. IFSA'01 World Congress*, Vancouver (2001) 2499-2504.
- [4] B. Bouchon-Meunier, C. Marsala, M. Rifqi. Interpolative reasoning based on granularity. *Proc. FUZZ-IEEE'2000* (2000) 483-487.
- [5] J.L. Castro, M. Delgado, Fuzzy Systems with defuzzification are universal approximators, *IEEE Trans. Systems, Man and Cybernetics*, **26** (1996) 149-152.
- [6] D. Dubois, H. Prade On Fuzzy interpolation, *Int. J. General Systems* **28** (1999) 103-114.
- [7] M. Dermirci, J. Recasens, Fuzzy groups, fuzzy functions and Fuzzy equivalent relations, *Fuzzy Sets and Systems* **144** (2004) 441-450.
- [8] F. Esteva et al. Similarity-based fuzzy interpolation method. *Proc. IPMU 2004*. Perugia (2004) 1443-1449.
- [9] G. Farin, *Curves and Surfaces for CAGD*. Academic Press, 1990.
- [10] I.D. Faux, M.J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd., 1985.
- [11] D. Hearn, M.P. Baker, *Computer Graphics*, Prentice Hall, 1994.
- [12] J. Jacas, J. Recasens, On some kind of probabilistic relations. *Proc IPMU Congress*, Palma de Mallorca, (1992) 365-368.
- [13] J. Jacas, J. Recasens, Fuzzy Numbers and Equality Relations, *Proc. IEEE International Conference on Fuzzy Systems* San Francisco (1993), 1298-1301..
- [14] J. Jacas, J. Recasens, Fixed Points and Generators of Fuzzy Relations, *J. Math. Anal. and Appl.* **186** (1994) 21-29.
- [15] J. Jacas, J. Recasens, Fuzzy T-transitive relations: eigenvectors and generators, *Fuzzy Sets and Systems* **72** (1995) 147-154.
- [16] J. Jacas, A. Monreal, J. Recasens, A model for CAGD using fuzzy logic, *Int. J. Approximate Reasoning* **72** (1997) 289-308.
- [17] F. Klawoon, R. Kruse, Equality Relations as a Basis for Fuzzy Control. *Proc. IEEE International Conference on Fuzzy Systems*, (1992).
- [18] L. T. Kôckzy, K. Hirota. Interpolative reasoning with insufficient evidence in sparse rulebases. *Information Sciences* **71**(1993) 169-200.
- [19] S. Mizik, P. Baranyi, P. Korondi, H and L. T. Kczy. A comparative of fuzzy interpolation techniques. *4th Eurofuzzy-Sic'99* (1999) Budapest 544-549
- [20] H. Takagi, M. Sugeno, Fuzzy identification of Systems and its applications to modelling and control, *IEEE Trans. on Systems, Man and Cybernetics*, **15**, 116-132, (1985).

Índice de Interpretabilidad Semántica para el Ajuste de Sistemas Basados en Reglas Difusas y Selección de Reglas Mediante un Algoritmo Evolutivo Multi-Objetivo

M.J. Gacto¹ R. Alcalá² F. Herrera²

¹ Dept. Informática, Universidad de Jaén, Jaén, España, mjgacto@ugr.es

² Dept. Ciencias de la Computación e I. A., Universidad de Granada, Granada, España, {alcala,herrera}@decsai.ugr.es

Resumen

En esta contribución proponemos un índice de interpretabilidad formado por la unión de tres métricas para preservar la interpretabilidad semántica de un Sistema Basado en Reglas Difusas mientras se realiza un ajuste de las funciones de pertenencia. Dicha medida puede ser combinada con una medida de complejidad cuando se aplica de manera conjunta una selección de reglas.

Para ello presentamos un Algoritmo Evolutivo Multi-Objetivo que realiza un ajuste de las funciones de pertenencia junto con una selección de reglas, considerando tres objetivos: maximizar la precisión y la interpretabilidad semántica y minimizar la complejidad.

El algoritmo propuesto ha sido comparado con un algoritmo mono-objetivo guiado por precisión en dos problemas reales, mostrando que muchas soluciones en el frente de Pareto dominan a las soluciones obtenidas por el algoritmo mono-objetivo.

Palabras Clave: Sistemas Basados en Reglas Difusas, Interpretabilidad Semántica, Complejidad, Algoritmos Evolutivos Multi-Objetivo.

1 Introducción

Usualmente, el Modelado Difuso se ha centrado en la mejora de la precisión de los modelos obtenidos sin prestar gran atención a la interpretabilidad, un aspecto esencial en los Sistemas Basados en Reglas Difusas (SBRD). Sin embargo, en los últimos años ha crecido el interés de los investigadores por resolver el

problema de encontrar el equilibrio entre interpretabilidad y precisión [1].

Una de las técnicas más usadas para mejorar el comportamiento de un SBRD es el ajuste de las Funciones de Pertenencia (FPs), que consiste en refinar una definición previa de la Base de Datos (BD) una vez que se ha obtenido la Base de Reglas (BR). Clásicamente, los métodos de ajuste refinan los parámetros que identifican las FPs asociados con las etiquetas que componen la BD [2, 3, 4].

Para que un método de ajuste tenga en cuenta la interpretabilidad es necesario usar una medida que cuantifique la interpretabilidad de la BD ajustada. Esta medida puede usarse como un objetivo adicional para mantener la interpretabilidad de las particiones difusas cuando se realiza el ajuste de las FPs.

Muchos autores buscan mejorar el difícil equilibrio entre interpretabilidad y precisión de un SBRD obteniendo modelos lingüísticos no solo precisos sino también interpretables. Se pueden distinguir dos formas de manejar la interpretabilidad:

1. Controlar la complejidad del modelo [2, 5, 6] (usando medidas como el número de reglas, de variables, de etiquetas por regla, etc.)
2. Medir la interpretabilidad de las particiones difusas [5, 7, 8] usando medidas de interpretabilidad semántica.

Una manera natural de trabajar es el uso de Algoritmos Evolutivos Multi-Objetivo (AEMOs) [9, 10], ya que nos permiten obtener un conjunto de soluciones con diferentes grados de precisión e interpretabilidad [2, 6, 7].

Mientras que en los trabajos anteriores se usan restricciones [8] o medidas absolutas [7], en este trabajo se propone un índice de interpretabilidad semántica relativo que permite mantener la interpretabilidad del sistema por medio de la agregación de varias métricas, con el objetivo de preservar la forma original de las

FPs que definiría un experto, mientras son ajustadas. Estas métricas tratan de minimizar el desplazamiento del punto central de las FPs y de mantener el ratio de amplitud lateral y el área de las FPs tanto como sea posible respecto a las originales. Para que pueda ser combinada con una medida de complejidad, proponemos un AEMO específico que realiza una selección de reglas junto con un ajuste de los parámetros de las FPs con tres objetivos:

- Minimizar el error del sistema
- Maximizar el índice de interpretabilidad propuesto
- Minimizar el número de reglas

El algoritmo propuesto está basado en el conocido *SPEA2* [11] y se llama *TS_{SP2-SI}* (*Tuning and Selection by SPEA2 for Semantic Interpretability*). La combinación de estas dos técnicas, ajuste y selección en el mismo proceso, permite obtener modelos incluso más precisos con una reducción importante en el número de reglas [12].

Para mostrar el buen comportamiento del método propuesto se compara con un algoritmo de ajuste y selección mono-objetivo orientado a precisión [12]. Los experimentos han sido realizados sobre dos problemas reales, mostrando que la solución del algoritmo mono-objetivo está dominada por varias de las soluciones obtenidas mediante *TS_{SP2-SI}*.

En la siguiente sección se presenta el índice de interpretabilidad de las FPs. La sección 3 muestra el algoritmo *TS_{SP2-SI}* describiendo sus principales características. Finalmente, las secciones 4 y 5 muestran un estudio experimental y las conclusiones obtenidas.

2 Índice de Interpretabilidad Semántica

En esta sección proponemos varias métricas para medir la interpretabilidad cuando se realiza un ajuste de la BD. Para asegurar la interpretabilidad semántica durante el proceso de optimización de las FPs [1, 8] algunos investigadores han propuesto varias propiedades. Considerando una o más de estas propiedades, varias restricciones se aplican en el proceso de diseño para obtener una BD que mantenga la compresibilidad del modelo lingüístico al nivel más alto posible [13, 14].

Para mostrar la efectividad del índice propuesto, preservando la interpretabilidad semántica de particiones previamente definidas, en este trabajo, usamos particiones difusas fuertes con FPs triangulares definidas mediante tres parámetros (ver Figura 1). Para mantener la integridad semántica de las FPs consideremos restricciones en la definición

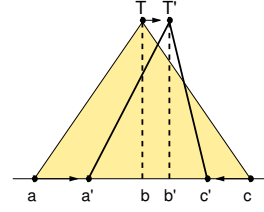


Figura 1: Ajuste clásico de una FP

de los intervalos de variación de cada parámetro. Para cada $FP_j = (a_j, b_j, c_j)$ donde $j=(1,...,m)$ y m es el número de FPs en una BD dada, los intervalos de variación son calculados como:

$$\begin{aligned} [I_{a_j}^l, I_{a_j}^r] &= [a_j - (b_j - a_j)/2, a_j + (b_j - a_j)/2] , \\ [I_{b_j}^l, I_{b_j}^r] &= [b_j - (b_j - a_j)/2, b_j + (c_j - b_j)/2] , \\ [I_{c_j}^l, I_{c_j}^r] &= [c_j - (c_j - b_j)/2, c_j + (c_j - b_j)/2] . \end{aligned}$$

De esta manera las métricas que se proponen están basadas en la existencia de éstos intervalos de variación (restricciones de integridad). Estas métricas permiten medir ciertas características de las FPs ajustadas respecto a las originales. Este índice y las métricas están propuestos para FPs triangulares simétricas, pero se pueden extender con unos pequeños cambios en la formulación a gaussianas o trapezoidales.

Las métricas propuestas son:

- Desplazamiento de las FPs (δ): Mide la proximidad de los puntos centrales de las FPs respecto a las posiciones originales.
- Ratio de amplitud lateral de las FPs (γ): Mide la diferencia del ratio izquierda/derecha de las FPs ajustadas respecto a las FPs originales.
- Similitud del área de las FPs (ρ): Mide la similitud del área de las FPs ajustadas respecto a las FPs originales.

2.1 Desplazamiento de las FPs (δ)

Esta métrica controla el desplazamiento del punto central de las FPs. Está basada en el cálculo de la distancia entre el punto central de la FP ajustada y el punto central de la FP original, y se calcula como la máxima distancia obtenida en todas las FPs. Para cada FP_j : $\delta_j = |b_j - b'_j|/I$, donde $I = (I_{b_j}^r - I_{b_j}^l)/2$ representa la máxima variación para cada parámetro central. Se define δ^* como: $\delta^* = \max_j \{\delta_j\}$. δ^* toma valores entre 0 y 1, donde valores cercanos a 1 indican que las FPs presentan un gran desplazamiento. Para que la métrica represente proximidad (maximización) se realiza la siguiente transformación:

$$\delta = 1 - \delta^*$$

2.2 Ratio de amplitud lateral de las FPs (γ)

Esta métrica es usada para controlar el ratio de amplitud lateral de las FPs. Está basada en la relación de las partes izquierda y derecha del soporte de las FPs original y ajustada. Si definimos $leftS_j = |a_j - b_j|$ como la parte izquierda de la FP original y $rightS_j = |b_j - c_j|$ como la parte derecha de la amplitud de la FP original. Y si definimos $leftS'_j = |a'_j - b'_j|$ y $rightS'_j = |b'_j - c'_j|$ como las correspondientes partes de la FP ajustada. γ_j se calcula usando la siguiente ecuación para cada FP:

$$\gamma_j = \frac{\min\{leftS_j/rightS_j, leftS'_j/rightS'_j\}}{\max\{leftS_j/rightS_j, leftS'_j/rightS'_j\}}$$

Valores cercanos a 1 significan que el ratio izquierda y derecha de las FPs originales se mantiene en las FPs ajustadas. γ se calcula obteniendo el mínimo γ_j :

$$\gamma = \min_j\{\gamma_j\}$$

2.3 Similitud del área de las FPs (ρ)

Esta métrica se usa para controlar el área de las FPs. Está basada en la relación entre las áreas de las FPs original y ajustada. Si definimos A_j como el área del triángulo que representa la FP original, y A'_j como la nueva área, ρ_j se calcula usando la siguiente ecuación para cada FP:

$$\rho_j = \min\{A_j, A'_j\} / \max\{A_j, A'_j\}$$

Valores cercanos a 1 significan que las áreas de las FPs original y ajustada son más cercanas. Esta métrica se calcula como el mínimo valor obtenido:

$$\rho = \min_j\{\rho_j\}$$

2.4 Índice de Interpretabilidad Semántica

En este trabajo se propone la agregación de las métricas en un índice global llamado índice GM3M. Para ello se utiliza la media geométrica porque en el caso de que sólo una de las métricas tenga valores muy bajos (ocasionando baja interpretabilidad) se obtiene también valores pequeños de GM3M:

$$GM3M = \sqrt[3]{\delta * \gamma * \rho}$$

GM3M toma valores entre 0 (menor nivel de interpretabilidad) y 1 (mayor nivel de interpretabilidad).

3 Algoritmo propuesto: TS_{SP2-SI}

El algoritmo propuesto realiza un ajuste de las FPs combinado con una selección de reglas y usa el índice GM3M para tratar de preservar la interpretabilidad semántica. Este algoritmo se llama TS_{SP2-SI} (*Tuning and Selection by SPEA2 for Semantic Interpretability*)

y esta basado en el conocido algoritmo SPEA2[11]. Implementa conceptos como el mecanismo de ventana [2], la prevención de incesto [15] y el reinicio que permiten enfocar la búsqueda en las soluciones más precisas y ayudan a obtener un frente de Pareto amplio. Se ha escogido como base el algoritmo SPEA2, debido a que en el estudio realizado en [2] se comprobó que los métodos basados en SPEA2 obtenían mejores resultados para este problema. En las siguientes subsecciones se describen sus principales características específicas.

3.1 Codificación y población inicial

Se utiliza un esquema de codificación doble para la *selección de reglas* (C_S) y el *ajuste* (C_T) donde:

$$C^p = C_S^p C_T^p$$

En el cual, $C_S^p = (c_{S1}, \dots, c_{Sm})$ es un vector binario en el que cada celda representa la existencia o no ('1' ó '0') de la regla correspondiente. Y donde $C_T^p = C_1 C_2 \dots C_n$ es un vector de números reales:

$$C_i = (a_1^i, b_1^i, c_1^i, \dots, a_{m^i}^i, b_{m^i}^i, c_{m^i}^i), \quad i = 1, \dots, n$$

donde m^i es el número de etiquetas de cada una de las n variables que componen la BD.

En la parte C_S la población inicial se obtiene con el primer individuo presentando en todos sus genes el valor '1' y el resto de individuos se generan de forma aleatoria. En la parte C_T la BD inicial se incluye como primer individuo y el resto de individuos se generan aleatoriamente dentro de los intervalos de variación.

3.2 Objetivos

Los objetivos considerados en este algoritmo son:

- Maximizar el Índice de Interpretabilidad Semántica ($GM3M$).
- Minimizar el Número de Reglas (NR).
- Minimizar el Error Cuadrático Medio (ECM):

$$ECM = \frac{1}{2 \cdot |E|} \sum_{l=1}^{|E|} (F(x^l) - y^l)^2,$$

donde $|E|$ es el tamaño del conjunto de datos, $F(x^l)$ es la salida obtenida por el SBRD para un ejemplo l y y^l es la salida esperada. Se utiliza el centro de gravedad ponderado por el grado de emparejamiento como operador de defuzzificación y la t-norma del mínimo como operador de implicación y conjunción.

3.3 Operadores de Cruce y Mutación

Este método utiliza un *operador de cruce* que permite aprovechar mejor la información contenida en

los padres, mejorando el equilibrio entre exploración y explotación. Como se usan codificaciones distintas en cada una de las partes del cromosoma, es necesario aplicar operadores distintos en cada zona. Para obtener cada hijo se aplican los siguientes operadores:

- Se aplica el cruce BLX-0.5 [16] para obtener la parte real del hijo.
- Una vez obtenida la parte real, la parte binaria se obtiene aprovechando la información de la parte real en padres e hijo. Para cada gen de la parte binaria (que representa un regla):
 1. Se obtienen las FPs de dicha regla a partir de los valores codificados en C_T para cada uno de los tres individuos (padres e hijo).
 2. Posteriormente se calcula la distancia euclídea normalizada entre el hijo y cada padre, considerando el punto central de las FPs.
 3. El padre más cercano determina si una regla es seleccionada o no para el hijo, copiando su valor en C_S para el gen correspondiente.

Este proceso se repite hasta que todos los valores de C_S son asignados en el hijo. Aplicando este operador la exploración se realiza en C_T . La parte C_S se obtiene basada en el conocimiento previo de cada padre.

El *operador de mutación* no añade reglas sino que directamente fija a cero un gen seleccionado aleatoriamente en C_S . En la parte C_T el operador de mutación cambia el valor de un gen de forma aleatoria. De los cuatro hijos obtenidos en el cruce y la mutación sólo los dos más precisos son considerados como descendientes.

3.4 Principales características de TS_{SP2-SI}

Este algoritmo hace uso del esquema de selección de *SPEA2*. Para mejorar la capacidad de búsqueda del algoritmo se realizan las siguientes cambios:

- Se incluye un mecanismo para la prevención de incesto basado en CHC [15], para mantener la diversidad de la población y evitar la convergencia prematura. Sólo los padres cuya distancia de hamming dividida por 4 sea mayor que un umbral son cruzados. Debido a que se utiliza un esquema de codificación real en C_T , cada gen es transformado a código gray con un número fijo de bits por gen ($BGenes$). De esta forma el valor del umbral es inicializado así: $L = (\#C_T * BGene)/4$, donde $\#C_T$ es el número de genes de la parte C_T del cromosoma. En cada iteración del algoritmo L se decrementa en 1 permitiendo el cruce de soluciones más cercanas.
- El operador de reinicio se aplica manteniendo el individuo más preciso y el más interpretable como

parte de la nueva población. Los restantes individuos toman el valor del individuo más preciso en C_S y valores generados aleatoriamente en C_T . El primer reinicio se produce cuando se realizan el 50 por ciento de los cruces en una generación (éste porcentaje se actualiza cada vez de la siguiente manera $r\% = 100 + r\%$). Sin embargo, no se aplica cuando no se mejore la solución más precisa o en las últimas evaluaciones del algoritmo para permitir obtener un frente de Pareto completo.

- Se incluye un mecanismo de ventana [2] cuyo objetivo es centrarse en las soluciones más precisas para ello se reduce progresivamente el número de soluciones en la población externa desde el 100% al principio hasta el 50% al final. En las últimas evaluaciones este mecanismo es desactivado para dejar que se forme un frente de Pareto amplio.

4 Experimentación

Para evaluar la bondad de la técnica propuesta se han realizado experimentos considerando dos problemas del mundo real con diferentes complejidades. Estos problemas (disponibles junto con sus descripciones en <http://www.keel.es/>), son:

- Estimación del costo de mantenimiento de las líneas eléctricas de media tensión (ELE). Este problema tiene 5 variables y 1056 ejemplos.
- Predicción de la edad del *Abalon* (ABA). Este problema tiene 8 variables y 4052 ejemplos.

Los métodos considerados en los experimentos son:

- *TS* realiza una selección de reglas y un ajuste de los parámetros de las FPs considerando la precisión como único objetivo [12].
- TS_{SP2-SI} propuesto en este trabajo para el ajuste combinado con una selección de reglas considerando los tres objetivos anteriormente mencionados.

En ambos casos, el método de Wang y Mendel (*WM*) [17] es usado para obtener la BR inicial que va a ser ajustada. Se utiliza un modelo de validación cruzada de 5 particiones generadas aleatoriamente a partir del conjunto de datos al 20% y se combinan 4 de ellas (80%) para formar el conjunto de entrenamiento y la restante se usa en el test. Para cada partición los métodos se han ejecutado 6 veces, mostrando los resultados medios de 30 ejecuciones. En TS_{SP2-SI} los valores medios se calculan considerando la solución más precisa de cada Pareto.

Los valores de los parámetros considerados para *TS* son: población de tamaño 61, 100000 evaluaciones, 0,6 como probabilidad de cruce y 0,2 como probabilidad de

mutación. Los valores de los parámetros considerados para TS_{SP2-SI} son: población de tamaño 200, población externa de tamaño 61, 100000 evaluaciones, 0,2 como probabilidad de mutación y 30 para $BGenes$.

La tabla 1 muestra los resultados obtenidos con WM , donde NR es el número de reglas, $ECM_{ent/tst}$ son los errores medios en entrenamiento y test, σ la desviación típica y $GM3M$ es el índice de interpretabilidad. Para el método WM el índice de interpretabilidad obtiene el valor 1 (máximo nivel de interpretabilidad).

Los resultados obtenidos por los métodos analizados se muestran en la tabla 2. En esta tabla se incluyen las tres métricas, y t que representa el resultado de aplicar un test t-student (con 95% de confianza) para comprobar si hay diferencias significativas entre los resultados. La interpretación de la columna t es: $[\star]$ indica el resultado con la mejor media y $[+]$ indica un comportamiento significativamente peor que el mejor.

Analizando los resultados mostrados en la tabla 2 podemos extraer las siguientes conclusiones:

- El algoritmo TS_{SP2-SI} obtiene mejores resultados en entrenamiento y test respecto al algoritmo TS en ambos problemas.
- En las soluciones más precisas del AEMO se obtienen mejoras en el índice de interpretabilidad de 36,5%(ELE) y 95%(ABA) respecto al método TS .
- El método propuesto ha obtenido BRs con una media de 12 reglas menos que TS . Además, se ha eliminado un gran número de reglas (más de la mitad) con respecto a la BR inicial obtenida con WM , mejorando incluso la precisión del sistema.

La figura 2 muestra el frente de Pareto obtenido con TS_{SP2-SI} y la solución obtenida con TS para ELE. La solución obtenida con TS es dominada por varias soluciones de TS_{SP2-SI} . Además se puede observar que el frente de Pareto es muy amplio, esto permite seleccionar soluciones con diferentes grados de precisión e interpretabilidad. La figura 2 muestra también como una mejora en interpretabilidad produce una pérdida de precisión y viceversa.

La figura 3 muestra la BD obtenida con TS y varias BDs obtenidas con TS_{SP2-SI} en ELE. Para TS_{SP2-SI} se incluyen tres BDs, una con la solución más precisa, otra con una solución no solo precisa sino también interpretable y otra con una BD muy interpretable, que

Tabla 1: Resultados obtenidos con el método WM

Problema	NR	ECM_{ent}	σ_{ent}	ECM_{tst}	σ_{tst}	GM3M
ELE	65	56136	1498	56359	4685	1
ABA	68	8,407	0,443	8,422	0,545	1

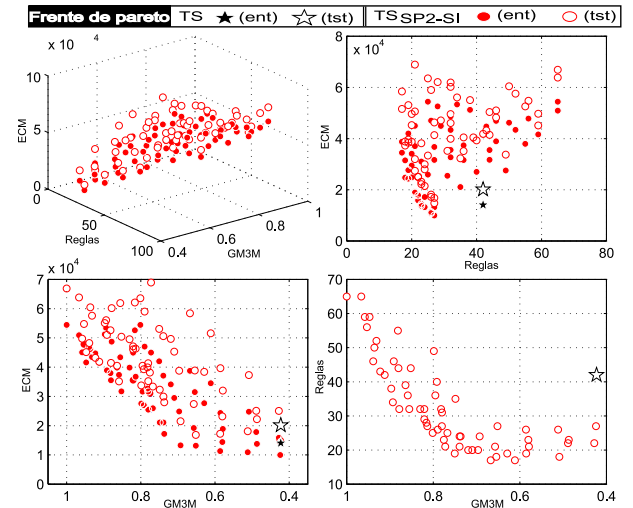


Figura 2: Frente de Pareto obtenido en ELE

obtiene un 29,5% de mejora con respecto a WM con un valor de índice de interpretabilidad cercano a 1.

5 Conclusiones

En este trabajo se presenta un índice para preservar la forma original de las FPs manteniendo la interpretabilidad de la BD a un nivel aceptable mientras que se mejora la precisión y se reduce la complejidad del sistema.

El AEMO propuesto realiza un ajuste de las FPs junto a una selección de reglas y permite obtener un frente de Pareto amplio con una variedad de soluciones desde las soluciones más precisas hasta las más interpretables. El algoritmo es capaz de obtener un conjunto de soluciones que claramente dominan a las obtenidas por un algoritmo mono-objetivo guiado por precisión, es decir, no sólo más interpretables sino también más precisas.

Agradecimientos

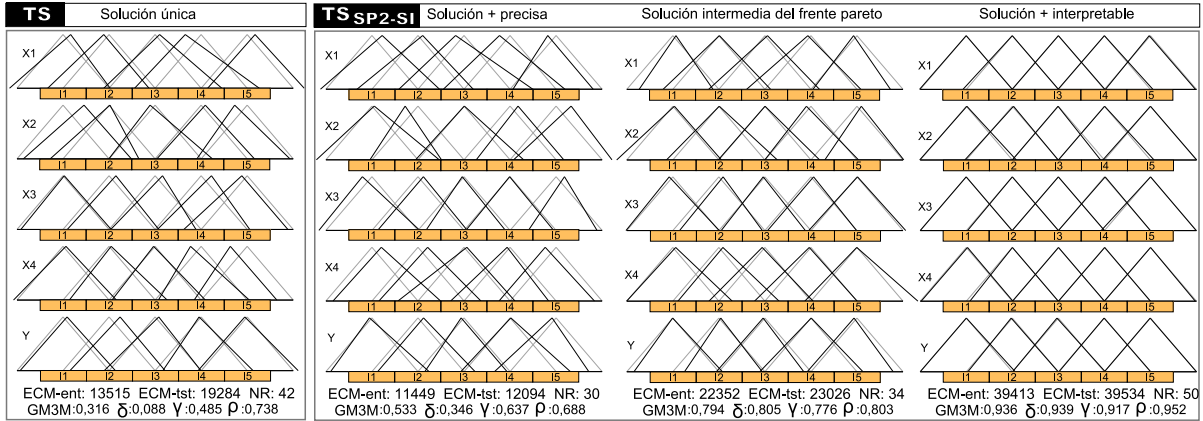
Soportado por el proyecto del Ministerio de Educación y Ciencia no. TIN2008-06681-C06-01

Referencias

- [1] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, editors. *Interpretability issues in fuzzy modeling*, volume 128 of *Studies in Fuzziness and Soft Computing*. Springer, 2003.
- [2] R. Alcalá, M. J. Gacto, F. Herrera, and J. Alcalá-Fdez. A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(5):539–557, 2007.

Tabla 2: Resultados obtenidos por los métodos analizados en ambos problemas

Problema	Método	NR	ECM _{ent}	σ_{ent}	t-test	ECM _{tst}	σ_{tst}	t-test	GM3M	σ_{GM3M}	t-test	(δ	γ	ρ)
ELE	TS	41,3	13387	1153	+	17784	3344	+	0,335	0,064	+	(0,145	0,387	0,738)
	TS_{SP2-SI}	29,3	11611	1191	*	14851	2892	*	0,528	0,113	*	(0,406	0,581	0,697)
ABA	TS	28,4	2,473	0,097	+	2,582	0,290	=	0,021	0,012	+	(0,082	0,358	0,738)
	TS_{SP2-SI}	16,3	2,386	0,073	*	2,513	0,208	*	0,450	0,110	*	(0,290	0,538	0,684)

Figura 3: BDs obtenidas por TS y TS_{SP2-SI} en la misma semilla y partición para el problema ELE

- [3] F. Herrera, M. Lozano, and J. L. Verdegay. Tuning fuzzy logic controllers by genetic algorithms. *International J. of Approximate Reasoning*, 12:299–315, 1995.
- [4] C.L. Karr. Genetic algorithms for fuzzy controllers. *AI Expert*, 6(2):26–33, 1991.
- [5] José M. Alonso, Luis Magdalena, and Serge Guillaume. Hilk: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems*, 23(7):761–794, 2008.
- [6] H. Ishibuchi and Y. Nojima. Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning*, 44(1):4–31, 2007.
- [7] A. Botta, B. Lazzerini, F. Marcelloni, and D. C. Stefanescu. Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index. *Soft Computing*, 13(5):437–449, 2008.
- [8] J. Valente de Oliveira. Towards neuro-linguistic modeling: constraints for optimization of membership functions. *Fuzzy Sets and Systems*, 106(3):357–380, 1999.
- [9] C. A. Coello, D. A. Van Veldhuizen, and G. B. Lamont, editors. *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers, 2002.
- [10] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, NY, USA, 2001.
- [11] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Proc. Evolutionary Methods for Design, Optimization and Control with App. to Industrial Problems*, pages 95–100, Barcelona, Spain, 2001.
- [12] J. Casillas, O. Cordon, M. J. Del Jesus, and F. Herrera. Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Trans. Fuzzy Syst.*, 13(1):13–29, 2005.
- [13] U. Bodenhofer and P. Bauer. A formal model of interpretability of linguistic variables. In Casillas et al. [1], pages 524–545.
- [14] J. Espinosa and J. Vandewalle. Constructing fuzzy models with linguistic integrity from numerical data: AFRELI algorithm. *IEEE Trans. Fuzzy Syst.*, 8(5):591–600, 2000.
- [15] L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G.J.E. Rawlin, editor, *Foundations of genetic Algorithms*, volume 1, pages 265–283. Morgan Kaufman, 1991.
- [16] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval-schemata. *Foundations of Genetic Algorithms*, 2:187–202, 1993.
- [17] L. X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. Syst., Man, Cybern.*, 22(6):1414–1427, 1992.

Mejora del equilibrio precisión-interpretabilidad en modelos difusos

M. Galende¹ G. I. Sainz^{1,2}

¹ Fundación CARTIF, Parque Tecnológico de Boecillo. Valladolid (España) {margal,gresai}@cartif.es

² Dpto. Ingeniería de Sistemas y Automática, E. I. Industriales. Universidad de Valladolid (España) gresai@eis.uva.es

Resumen

El presente artículo afronta el problema de la búsqueda de equilibrio entre precisión e interpretabilidad en el modelado preciso de sistemas basados en reglas difusas. Tomando como base un modelado neurodifuso, muy preciso pero poco interpretable, se presenta un método de mejora de la solución cuyo objetivo es mejorar la interpretabilidad manteniendo, e incluso mejorando, la precisión.

Se repasan diversas características de interpretabilidad que deben cumplir los sistemas basados en reglas difusas y se presenta una metodología que tiene en cuenta gran parte de esas características, proponiendo diversas formas de cálculo para las mismas y analizando los resultados sobre diversos casos de estudio.

Palabras Clave: Modelado Difuso, Algoritmo Genético, Equilibrio Precisión-Interpretabilidad.

lingüísticas en las que prima la interpretabilidad de la regla.

- **Modelado difuso preciso** (*Precise Fuzzy Modeling*), orientado a generar modelos cuyas reglas son aproximadas a partir de datos, compuestas por variables borrosas en las que prima la precisión. Este tipo de modelado es el más habitual en aplicaciones técnicas.

Teniendo en cuenta el Principio de Incompatibilidad de Zadeh [17], según el cual la precisión y la interpretabilidad son objetivos contradictorios, lo ideal sería entonces obtener sistemas con buenas prestaciones para ambas características.

El presente artículo propone un método en dos pasos que intenta alcanzar un equilibrio adecuado entre precisión e interpretabilidad, tomando como base el modelado difuso preciso y realizando un postprocesamiento de selección de reglas basado en algoritmos genéticos.

En el apartado 2 se muestra un listado de características de interpretabilidad propias de los sistemas difusos, en el 3 se presenta la metodología propuesta y en los apartados 4 y 5 los casos de estudio utilizados y algunos de los resultados obtenidos. Para finalizar el último apartado presenta las principales conclusiones obtenidas en la investigación y los futuros trabajos a realizar.

1 INTRODUCCIÓN

Actualmente el modelado de sistemas basados en reglas difusas, y más concretamente la búsqueda de interpretabilidad en dichos modelos, se ha convertido en un campo de investigación en auge [2, 3].

Inicialmente, y en función del objetivo principal que guía el proceso, existen dos tipos básicos de modelado difuso:

- **Modelado difuso lingüístico** (*Linguistic Fuzzy Modeling*), orientado a la generación de modelos difusos cuyas reglas son extraídas del conocimiento del experto, compuestas por variables

2 ÍNDICES DE INTERPRETABILIDAD

Definir la interpretabilidad es una tarea difícil y muy subjetiva. Sin embargo muchos autores [18, 16] han tratado el tema e intentado establecer una serie de características/propiedades que deben tener los sistemas basados en reglas difusas para que sean fáciles de interpretar. Entre otras las principales son:

- **Propiedades de las funciones de pertenencia:**

- Cobertura (*Coverage*): cada valor del universo de discurso debe pertenecer, al menos, a un término lingüístico/función de pertenencia/conjunto difuso.
- Sencillez (*Simplicity*): usar mínimo número de etiquetas/funciones de pertenencia/conjuntos difusos diferentes.
- Normalidad (*Normality*): cada término lingüístico/función de pertenencia debe alcanzar valor pleno significado (máximo valor) en al menos un elemento del universo de discurso.
- Distinguibilidad (*Distinguishability*): cada término lingüístico debe tener un significado claro y la función de pertenencia asociada estar claramente definida, de forma que cada una de ellas se distinga de todas las demás.
- Concisión (*Conciseness*): medida que evalúa la forma y la simetría de los conjuntos difusos.
- No usar conjuntos difusos irrelevantes, evitando funciones de pertenencia constantes para todo el universo de discurso.

- **Propiedades de la regla (de forma individual):**

- Reglas Mandani, en las que el consecuente es también un término lingüístico/conjunto difuso. Estas reglas son más interpretables que las reglas TSK.
- Sencillez (*Simplicity*): usar mínimo número de variables (antecedentes y/o consecuentes) diferentes.
- No usar reglas irrelevantes que se activen en todos los casos.

- **Propiedades del conjunto de reglas (de forma global):**

- Compactitud (*Compactness*): usar el mínimo número de reglas que asegure un buen resultado. Implica eliminar reglas redundantes.
- Consistencia (*Consistency*): todas las posibles combinaciones de antecedentes deben tener solo un consecuente. Es decir que no debe haber reglas incoherentes.
- Completitud (*Completeness*): cada dato debe activar al menos una regla. Para ello las reglas difusas deben cubrir todo el universo de discurso, de forma que para cualquier entrada se active al menos una regla.

3 METODOLOGIA PROPUESTA

Para intentar alcanzar un equilibrio adecuado entre precisión e interpretabilidad, se propone aquí un método en dos pasos (figura 1) en el que:

1. A partir de cualquiera de los métodos existentes se genera un modelo difuso basado en reglas.
2. Se refina y se mejoran las prestaciones del modelo difuso en cuanto a interpretabilidad, conservando un nivel de precisión adecuado.

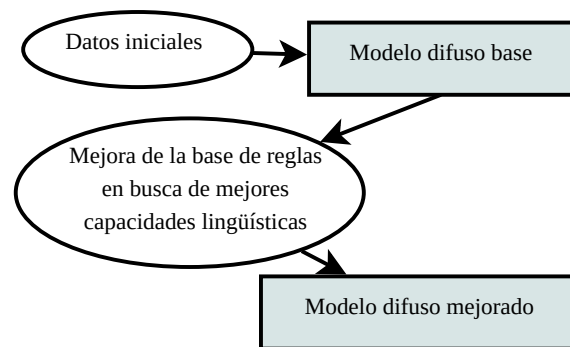


Figura 1: Proceso de mejora.

3.1 FASE I: MODELO DIFUSO BASE

En esta fase del método, a partir de datos o del conocimiento del experto, se aplica cualquier algoritmo de **modelado difuso existente** para generar un modelo difuso base que posteriormente en mejorado/refinado en la fase II.

Las ventajas de poder utilizar cualquier algoritmo de modelado ya existente son evidentes: los algoritmos ya han sido probados, generan modelos difusos precisos y existe gran variedad.

3.2 FASE II: MODELO DIFUSO MEJORADO

Para alcanzar el objetivo de obtener un sistema difuso con un buen equilibrio entre precisión e interpretabilidad en la fase II se emplea una técnica de mejora basada en **algoritmos genéticos multiobjetivo**. Concretamente se definen dos funciones objetivo que guían el proceso de selección de reglas en esta fase, una para medir la precisión y otra para medir la interpretabilidad.

3.2.1 Interpretabilidad

Teniendo en cuenta el listado de características de la sección 2 se establecen una serie de índices que per-

miten medir cuantitativamente la interpretabilidad en un sistema difuso.

- Compactitud medida a través del **numero de reglas** (ecuación 1) como parte de la simplicidad final del modelo, de forma que al disminuir las reglas aumenta la interpretabilidad.

$$Compactitud = Numero\ de\ reglas \quad (1)$$

- Distinguibilidad medida a través de la **similitud**, ya que cuanto menos similares sean entre sí las reglas más fáciles de entender por el ser humano. La similitud de dos reglas cualesquiera se calcula aplicando la medida propuesta en [15] sobre los antecedentes de las reglas (ecuación 2). Posteriormente se calcula la similitud del conjunto de reglas difusas como el promedio o el producto de la similitud en todos los antecedentes (ecuación 3).

$$S_k(R_i, R_j) = \frac{\sum_{i,j} R_{ik}(x) \wedge R_{jk}(x)}{\sum_{i,j} R_{ik}(x) \vee R_{jk}(x)} = \frac{\sum_{i,j} \min(R_{ik}, R_{jk})}{\sum_{i,j} \max(R_{ik}, R_{jk})}$$

$$\forall 1 \leq i < j \leq NumReglas$$

$$\forall 1 \leq k \leq NumAntecedentes \quad (2)$$

$$Similitud = Promedio_{i,j}(Promedio_k(S_k(R_i, R_j)))$$

$$Similitud = Producto_{i,j}(Producto_k(S_k(R_i, R_j)))$$

$$\forall 1 \leq i < j \leq NumReglas$$

$$\forall 1 \leq k \leq NumAntecedentes \quad (3)$$

Dicha medida ya ha sido usada con éxito por otros autores en el proceso de generación de reglas con el objetivo de reducir la complejidad de los sistemas difusos resultantes [3, 14, 11]. Sin embargo no es la única forma de calcular la similitud existiendo otras opciones que también pueden usarse [10, 12].

- Eliminación de **redundancias**, utilizando así el menor numero de reglas posibles y facilitando la interpretación final. Se dice que un par de reglas es redundante si tanto sus antecedentes como sus consecuentes son similares. Para calcular la redundancia total del conjunto de reglas se establece un umbral de redundancia β_R ($0 < \beta_R < 1$) y se aplica la ecuación 4.

$$Redundancia = \frac{Card(S_A(R_i, R_j) > \beta_R \vee S_C(R_i, R_j) > \beta_R)}{(NumReglas-1)!}$$

$$\forall 1 \leq i < j \leq NumReglas$$

$$\forall 1 \leq A \leq NumAntecedentes$$

$$\forall 1 \leq C \leq NumConsecuentes \quad (4)$$

- Eliminación de **incoherencias**, aumentando así la consistencia del modelo. Se establece un umbral de incoherencia $\beta_I = 1 - \beta_R$ y se aplica la ecuación 5.

$$Incoherencia = \frac{Card(S_A(R_i, R_j) > \beta_R \vee S_C(R_i, R_j) < \beta_I)}{(NumReglas-1)!}$$

$$\forall 1 \leq i < j \leq NumReglas$$

$$\forall 1 \leq A \leq NumAntecedentes$$

$$\forall 1 \leq C \leq NumConsecuentes \quad (5)$$

- Completitud o **cobertura**, de forma que si todas las entradas tienen al menos una salida definida el modelo es más interpretable (ecuación 6).

$$NoCobertura = Promedio_k(NoCoberturaParticion_k)$$

$$NoCoberturaParticion_k = \frac{PuntosSinCobertura}{PuntosTotales}$$

$$\forall 1 \leq k \leq NumAntecedentes + NumConsecuentes \quad (6)$$

3.2.2 Método de selección de reglas

Todos los índices anteriores se combinan en dos funciones objetivo que guían el **proceso de selección de reglas** que realiza en algoritmo genético en la fase II.

Por un lado todas las propiedades de interpretabilidad se balancean ($\lambda_i * IndiceInterpretabilidad$) y se unen (*promedio* o *producto*) en un solo valor de interpretabilidad global del modelo difuso (ecuación 7):

$$Interpretabilidad = f(\lambda_{nr} * NumReglas,$$

$$\lambda_s * Similitud, \lambda_r * Redundancia,$$

$$\lambda_i * Incoherencia, \lambda_c * NoCobertura) \quad (7)$$

$$f \Rightarrow Promedio\ o\ Producto$$

Por su parte para cuantificar la precisión del modelo basado en reglas difusas se utiliza el error cuadrático medio, medida ampliamente aceptada para medir la exactitud en problemas de modelado:

$$ECM = \frac{1}{N} \sum_{i=1}^N (y_i - y'_i)^2 \quad (8)$$

En todos los casos se considera la diferencia normalizada de los índices propuestos (ecuación 9). Este $Indice_{nor}$ se usa directamente con el operador promedio y como $Indice_{nor} + 1$ para el producto.

$$Indice_{nor} = \frac{IndiceActual - IndiceOrigen}{IndiceActual} \quad (9)$$

Los individuos se codifican a través de un string binario donde 1 indica la presencia de una regla y 0 su ausencia. Se generan poblaciones iniciales aleatorias con

al menos un individuo igual al sistema difuso inicial y se consideran individuos compuesto por subconjuntos de reglas que evolucionan independientemente.

En todos los casos los operadores de selección, cruce y mutación se ejecutan en sus opciones por defecto y el criterio de parada es el número máximo de generaciones exploradas.

4 CASOS DE ESTUDIO

El método propuesto se valida sobre dos casos de estudio:

1. Horno de gas Box-Jenkins¹ con seis entradas ($x(k-1), x(k-2), x(k-3), y(k-1), y(k-2), y(k-3)$) y una salida ($y(k)$) según el proceso descrito en [7].
2. Proceso biotecnológico de una estación de depuración de aguas residuales (EDAR) con dos entradas (Caudal de entrada a la planta y Caudal recirculado) y una salida (Substrato).

5 EXPERIMENTACIÓN

A modo de ejemplo en fase I de la metodología propuesta se emplean dos algoritmos neurodifusos de modelado bien conocidos: **FasArt** (*Fuzzy Adaptive System ART based*) [1] y **NefProx**² (*Neuro-Fuzzy Function Approximation*) [13]. Para cada caso de estudio generamos varios sistemas difusos iniciales, unos más compactos y otros más complejos.

Posteriormente en la fase II se utiliza el algoritmo genérico multiobjetivo **NSGA-II** [4]. Se realizan 30 ejecuciones diferentes para cada una de las alternativas consideradas con validación cruzada, codificación Gray, poblaciones con $\sqrt{\text{NumReglas}}$ individuos y un número máximo de generaciones exploradas de 100.

En cuanto a la medida de interpretabilidad propuesta (ecuación 7) se considera que todos los índices tienen la misma relevancia, por lo que $\lambda = 1$ en todos los casos. Por su parte $\beta_R = 0.8$ [3, 14].

5.1 Resultados

Por falta de espacio solo se presentan en detalle algunas de las soluciones obtenidas más importantes. De esta forma se selecciona un punto de interés del Frente de Pareto, concretamente aquel con mejor equilibrio precisión-interpretabilidad, y se calculan los valores

¹<http://www.stat.wisc.edu/~reinsel/bjr-data/gas-furnace>

²<http://fuzzy.cs.uni-magdeburg.de/nefprox/>

medios de todas las ejecuciones obtenidos al utilizar el promedio en las ecuaciones 3 y 7.

5.1.1 Box Jenkins

Fase I. A partir de los datos del benchmark y mediante validación cruzada se generan dos modelos difusos con FasArt y dos modelos difusos con NefProx sobre los que realizar los experimentos:

- FasArt: BJ1 ($\rho_A = \rho_B = 0.6$ $\gamma_A = \gamma_B = 3$) y BJ2 ($\rho_A = \rho_B = 1$ $\gamma_A = \gamma_B = 12$)
- NefProx: BJ3 (NefProx $imsf/omsf = 5$ $maxrulex = 0$) y BJ4 ($imsf/omsf = 7$ $maxrulex = -1$)

La tabla 1 muestra las prestaciones iniciales de cada uno de estos modelos base. Los modelos BJ1 y BJ3 se caracterizan por ser más compacto, mientras que los modelos BJ2 y BJ4 son más complejos y difusos. Los sistemas compactos tienen un número bajo de reglas con un error pequeño mientras que los sistemas más complejos se caracterizan por tener gran número de reglas y bajo error. La similitud entre reglas puede mejorarse en la mayor parte de los casos, al igual que la redundancia y la incoherencia, y las particiones difusas de partidas suelen ser completas.

Tabla 1: Prestaciones iniciales Box-Jenkins.

	BJ1	BJ2	BJ3	BJ4
Nº Reglas	31	293	86	222
Error _{tra}	0.0030	0.00002	0.0040	0.0001
Error _{tst}	0.0030	0.00002	0.0040	0.0001
Similitud	0.1835	0.1420	0.3045	0.2073
Redundancia	0	0.00002	0.0216	0.0068
Incoherencia	0	0	0.0077	0.0023
Completitud	100%	100%	99.74%	100%

Fase II. La tabla 2 muestra valores medios del mejor punto de equilibrio obtenido en las ejecuciones realizadas con los modelos de Box-Jenkins. Como se puede observar la interpretabilidad mejora en todos los casos manteniendo un nivel de precisión adecuado.

Se reduce en número de reglas entre un 5.8% y un 40.7%, la similitud entre 1.1% y 3.6%, la redundancia entre 7.2% y 100%, la incoherencia entre 10.2% y 79.5%. Las particiones difusas se mantienen completas en como mínimo un 99.7%. En cuanto al error este aumenta ligeramente en algunos casos (de 0.00002 a 0.001), pero este incremento es admisible teniendo en cuenta la mejora obtenida en la interpretabilidad y que el valor final del error sigue siendo muy pequeño.

Tabla 2: Mejora modelos Box-Jenkins.

	BJ1	BJ2	BJ3	BJ4
NR	25 -19.4%	276 -5.8%	51 -40.7%	188 -15.3%
E_{tra}	0.0031 3.1%	0.001 283.3%	0.0040 -0.3%	0.001 9.2%
E_{tst}	0.0031 3.1%	0.0001 283.3%	0.0040 -0.3%	0.0001 9.2%
S	0.1769 -3.6%	0.1395 -1.8%	0.2950 -3.1%	0.2050 -1.1%
R	0 -	0 -100%	0.0173 -20.2%	0.0063 -7.2%
I	0 -	0 -	0.0016 -79.5%	0.0026 10.2%
C	100%	100%	99.7%	100%

5.1.2 EDAR

Fase I. A partir de los datos de la planta y mediante validación cruzada se generan tres modelos difusos con FasArt y uno con NefProx:

- FasArt: D1 ($\rho_A = \rho_B = 0.7$ $\gamma_A = \gamma_B = 10$), D2 ($\rho_A = \rho_B = 0.9$ $\gamma_A = \gamma_B = 10$) y D3 ($\rho_A = \rho_B = 1$ $\gamma_A = \gamma_B = 10$)
- NefProx: D4($imsf/omsf = 5$ $maxrulex = -1$)

Al igual que antes se tienen modelos más compactos (D1, D2, D4) y modelos más complejos (D3). La tabla 3 muestra las prestaciones iniciales de cada uno de los modelos base.

Tabla 3: Prestaciones iniciales EDAR.

	D1	D2	D3	D4
Nº Reglas	9	30	237	11
Error _{tra}	0.0164	0.0119	0.0109	0.0139
Error _{tst}	0.0172	0.0124	0.0115	0.0146
Similitud	0.3464	0.3668	0.4494	0.2896
Redundancia	0	0	0.0317	0
Incoherencia	0	0	0.0011	0
Compleitud	100%	100%	100%	100%

Fase II. La tabla 4 muestra valores medios del mejor punto de equilibrio obtenido en las ejecuciones realizadas con los modelos de la EDAR. En todos los casos se obtienen sistemas que mejoran tanto la precisión como la interpretabilidad del modelo base de forma simultánea. De forma individual se mejoran todos los índices de interpretabilidad (número de reglas en 71.2%, similitud en 60.8%, redundancia en 8.3% e incoherencia en 82.3%) a excepción de la completitud

que disminuye ligeramente (hasta un 82.85%). El error disminuye hasta en un 29%.

Tabla 4: Mejora modelos EDAR.

	D1	D2	D3	D4
NR	3.03 -66.3%	8.91 -70.3%	117.63 -50.4%	3.17 -71.2%
E_{tra}	0.0116 -29.0%	0.0110 -7.5%	0.0106 -2.4%	0.0117 -15.4%
E_{tst}	0.0123 -28.7%	0.0122 -1.2%	0.0115 0.4%	0.0123 -15.8%
S	0.3396 -1.9%	0.3606 -1.7%	0.4406 -2.0%	0.1136 -60.8%
R	0 -	0 -	0.0291 -8.3%	0 -
I	0 -	0 -	0.0002 -82.3%	0 -
C	82.85%	91.98%	95.92%	89.74%

6 CONCLUSIONES Y TRABAJO FUTURO

El presente artículo presenta una metodología en dos pasos que busca obtener sistemas difusos con un buen equilibrio precisión-interpretabilidad. Los autores ya han expuesto previamente esta metodología en [6, 5], aunque aquí se utilizan índices de interpretabilidad optimizados y otros casos de estudio.

Las principales ventajas del método en relación a otros ya existentes [8, 9] son:

- Metodología abierta, sencilla y fácilmente implemtable con cualquier alternativa.
- Mejora simultánea de precisión e interpretabilidad.
- Utilización de índices más complejos para definir el concepto de interpretabilidad.
- Sistema Mandani en vez de Takagi-Sugeno.
- Problemas de modelado en vez de clasificación.

Los experimentos realizados con dos casos de estudio diferentes han demostrado que es posible obtener un mejor balance precisión-interpretabilidad tanto en modelos compactos, y por tanto más difícil de conseguir una mejora, como en modelos mucho más complejos, demostrándose que la metodología y criterios propuestos son suficientemente válidos.

Cara al futuro más inmediato están en marcha las siguientes iniciativas:

- Ampliación de la metodología a un mayor número de casos de estudio.
- Realización de test estadísticos no paramétricos.
- Mejora y refinamiento de los índices de interpretabilidad utilizados.

Referencias

- [1] J. M. Cano Izquierdo, Y. A. Dimitriadis, E. Gómez Sánchez, and J. López Coronado. Learning from noisy information in FasArt and fasback neuro-fuzzy systems. *Neural Networks*, 14(4-5):407–425, May 2001.
- [2] J. Casillas, O. Cordon, Herrera F., and L. Magdalena. *Accuracy Improvements in Linguistic Fuzzy Modelling*, volume 129 of *Studies in Fuzziness and SoftComputing*, chapter Accuracy Improvements to Find the Balance Interpretability-Accuracy in Linguistic Fuzzy Modeling: An Overview, pages 3–24. Springer-Verlag, Berlin Heidelberg, 2003.
- [3] J. Casillas, O. Cordon, Herrera F., and L. Magdalena. *Interpretability Issues in Fuzzy Modelling*, volume 128 of *Studies in Fuzziness and SoftComputing*, chapter Interpretability Improvements to Find the Balance Interpretability-Accuracy in Fuzzy Modeling: An Overview, pages 3–22. Springer-Verlag, Berlin Heidelberg, 2003.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [5] M. Galende, G. Sainz, and M.J. Fuente. Accuracy-interpretability balance in fuzzy models based on multiobjective genetic algorithm. In *Proceedings of European Control Conference 2009 (ECC'09)*, pages 3915 – 3920, Budapest, Hungary, 23 - 26 August 2009.
- [6] M. Galende, G. I. Sainz, M. J. Fuente, and A. Herreros. Interpretability-accuracy improvement in a neuro-fuzzy ART based model of a DC motor. In *Proceedings of the 17th IFAC World Congress*, pages 7034 – 7039, Seoul, Korea, 6-11 July 2008.
- [7] A. E. Gaweda and J. M. Zurada. Data-driven linguistic modeling using relational fuzzy rules. *IEEE Transactions on Fuzzy Systems*, 11(1):121–134, 2003.
- [8] H. Ishibuchi, T. Murata, and I. B. Türkmen. Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems*, 89(2):135 – 150, 16 July 1997.
- [9] H. Ishibuchi, T. Nakashima, and T. Murata. Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences*, 136(1-4):109 – 133, August 2001.
- [10] Y. Ji, W. Seelen, and B. Sendhoff. On generating FC^3 fuzzy rule systems from data using evolution strategies. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, 29(6):829–845, December 1999.
- [11] F. Jimenez, A. F. Gómez-Skarmeta, G. Sanchez, H. Roubos, and R. Babuska. *Interpretability Issues in Fuzzy Modelling*, volume 128 of *Studies in Fuzziness and SoftComputing*, chapter Accurate, Transparent and Compact Fuzzy Models by Multi-Objective Evolutionary Algorithms, pages 431–451. Springer-Verlag, Berlin Heidelberg, 2003.
- [12] Y. Jin. Fuzzy Modeling of High-Dimensional Systems: Complexity Reduction and Interpretability Improvement. *IEEE Transactions on Fuzzy Systems*, 8(2):212–221, April 2000.
- [13] D. Nauck and R. Kruse. Neuro-fuzzy systems for function approximation. In *4th International Workshop Fuzzy-Neuro-Systeme '97 (FNS'97). Proceedings in Artificial Intelligence*, pages 316–323, Soest, 1997.
- [14] M. Setnes. *Interpretability Issues in Fuzzy Modelling*, volume 128 of *Studies in Fuzziness and SoftComputing*, chapter Simplification and Reduction of Fuzzy Rules, pages 278–302. Springer-Verlag, Berlin Heidelberg, 2003.
- [15] M. Setnes, R. Babuska, U. Kaymak, and H. R. Lemke. Similarity measures in fuzzy rule base simplification. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 28(3):376–386, June 1998.
- [16] T. Suzuki and T. Furuhashi. *Interpretability Issues in Fuzzy Modelling*, volume 128 of *Studies in Fuzziness and SoftComputing*, chapter Conciseness of Fuzzy Models, pages 569–586. Springer-Verlag, Berlin Heidelberg, 2003.
- [17] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, 3(1):28 – 44, January 1973.
- [18] S.-M. Zhou and J. Q. Ganb. Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets and Systems*, 159:3091 – 3131, 2008.

INTRODUCCION DE UN MECANISMO PARA LA MEJORA DE LA INTERPRETABILIDAD EN SISTEMAS DIFUSOS LINGÜÍSTICOS CON DEFUZZIFICACION ADAPTATIVA

A.A. Márquez F.A. Márquez A. Peregrín

Dpto. de Tecnologías de la Información. Universidad de Huelva 21071 – Huelva,
{amarquez, alfredo.marquez, peregrin}@dti.uhu.es,

Resumen

En este trabajo se propone un mecanismo para la mejora de la interpretabilidad en sistemas difusos lingüísticos con defuzzificación adaptativa. El uso de parámetros en el operador de defuzzificación introduce una serie de valores o pesos asociados indirectamente a cada regla, que mejoran su precisión pero aumentan la complejidad y por tanto perjudican a la interpretabilidad del sistema. En este trabajo se trata de reducir dicha complejidad pretendiendo obtener tanto las ventajas de una defuzzificación adaptativa en cuanto a precisión, como reducir la pérdida de interpretabilidad. Para tal fin, utilizaremos un modelo evolutivo multiobjetivo que proporcionará un conjunto de sistemas difusos lingüísticos, con el mejor equilibrio entre el número de reglas finales, el número de reglas con peso y la precisión del sistema.

Palabras Clave: Modelado Difuso Lingüístico, Interpretabilidad, Algoritmos Genéticos Multiobjetivo, Sistema de Defuzzificación Adaptativo, Selección de Reglas, Reglas Ponderadas.

1 INTRODUCCION

En la actualidad, el problema de encontrar un equilibrio entre interpretabilidad y precisión en Sistemas Basados en Reglas Difusas Lingüísticas, a pesar de la naturaleza intrínseca de la lógica difusa, ha dado lugar a un creciente interés en métodos que tienen en cuenta ambos aspectos [4, 5]. Por supuesto, lo ideal sería satisfacer ambos criterios en un grado alto, pero debido a que son características contradictorias, por lo general esto no es

posible. Un camino para conseguirlo es mejorar la precisión del sistema mientras se mantiene en lo posible la interpretabilidad a un nivel aceptable [4].

En este ámbito, el sistema de inferencia y los métodos de defuzzificación adaptativos han demostrado ser unos elementos capaces de mejorar fácilmente la precisión del sistema [1, 8, 15] debido a que buscan la mejor forma de inferir la contribución de cada regla de forma apropiada [8] y mejorar la cooperación entre las reglas [1]. Estos pueden también adaptar especialmente el comportamiento de los operadores difusos a la base de reglas (BR) o aprender los operadores difusos y la BR conjuntamente [15], obteniendo una sinergia positiva entre ambos elementos que permite al sistema alcanzar un nivel alto de precisión con la pérdida de interpretabilidad asociada al uso de operadores adaptativos con sus correspondientes parámetros.

Por otro lado, en la literatura se distinguen dos clases de aproximaciones en interpretabilidad [3, 17]:

1. Interpretabilidad Global (estructura del sistema): La complejidad del modelo (generalmente mediada como número de reglas, variables, etiquetas por variable, etc.)
2. Interpretabilidad Local (comprensión del modelo): Una medida semántica de la interpretabilidad (sistema de inferencia utilizado, método de defuzzificación, operador de conjunción, etc.)

Por tanto, la defuzzificación adaptativa [8], introducirá una pérdida de interpretabilidad en principio en el sentido local, con efectos en sentido global, en tanto los parámetros asociados a cada regla aumentan la estructura del sistema con la inclusión de un peso. Sería deseable pues, minimizar los elementos que aumenten la complejidad del sistema.

En este trabajo, se propone introducir, en primer lugar, un mecanismo que permita mejorar la interpretabilidad cuando se utilizan defuzzificación adaptativa. Dicho mecanismo se basará en dos índices que permitan

eliminar el peso de la regla, o bien la regla completa en el mismo proceso de aprendizaje evolutivo de la defuzzificación adaptativa.

En segundo lugar, se propone utilizar un Algoritmo Genético Multi-objetivo (AGMO) [7, 9] como modelo evolutivo para obtener modelos difusos lingüísticos precisos e interpretables utilizando la defuzzificación adaptativa con tres objetivos (minimizar la precisión, y minimizar los dos índices de interpretabilidad comentados anteriormente). De esta forma, se obtendrá un conjunto de soluciones con distinto equilibrio entre precisión e interpretabilidad [11, 12, 13, 14].

El trabajo se estructura de la siguiente forma: en la Sección 2 se repasa la construcción de la defuzzificación adaptativa utilizada, sus componentes y efectos. En la Sección 3, se presenta las bases del mecanismo de mejora de interpretabilidad propuesto. La Sección 4 presenta el modelo multi-objetivo, describiendo sus principales características y operadores genéticos considerados. La Sección 5 se dedica a un estudio experimental en el que se emplean las técnicas anteriormente descritas en dos problemas reales, para finalmente presentar las conclusiones del estudio desarrollado.

2 MÉTODOS DE DEFUZZIFICACIÓN ADAPTATIVOS EVOLUTIVOS

Existen varias tendencias en la literatura en el desarrollo de métodos de defuzzificación adaptativa. Éstos emplean uno o más parámetros en sus expresiones con el fin de modificar el comportamiento de la defuzzificación o, en la mayoría de los casos, mejorar la precisión del sistema de forma considerable.

Siguiendo los estudios desarrollados en [8], y debido a su fácil implementación y buen comportamiento, en el presente trabajo hemos considerado la utilización de un método de defuzzificación adaptativo basado en (1):

$$y_0 = \frac{\sum_i^N f(h_i) \cdot V_i}{\sum_i^N f(h_i)}, \quad (1)$$

donde h_i es el grado de emparejamiento, $f(h_i)$ es un término funcional del grado de emparejamiento y V_i es el máximo valor (MV_i) o el centro de gravedad (CG_i). Como puede observarse, se trata de la expresión de un método de defuzzificación que actúa en Modo B, es decir, primero se defuzzifica la aportación individual de la inferencia con cada regla y posteriormente se computa el resultado final, concretamente mediante una suma ponderada.

Concretamente, hemos considerado el uso de un término funcional tipo producto $f(h_i) = h_i \cdot \alpha_i$ donde α_i corresponde a un parámetro para cada regla R_i , $i=1$ a N , y como valor característico el Centro de Gravedad (CG), debido a su eficiencia computacional y obtener similares resultados a otro tipo de funciones [8]. La expresión de dicho método de defuzzificación adaptativa se muestra en (2),

$$y_0 = \frac{\sum_i^N h_i \cdot \alpha_i \cdot CG_i}{\sum_i^N h_i \cdot \alpha_i}, \quad (2)$$

Como se puede apreciar, la influencia del parámetro individual α se puede interpretar como una modulación de la influencia del grado de emparejamiento, potenciándolo o penalizándolo. Particularmente, el estudio zonal del efecto de α es el siguiente:

- $\alpha \cdot h_i$, $\alpha \in [1, \infty)$: potenciación de h_i ,
- $\alpha \cdot h_i$, $\alpha \in [0, 1]$: penalización de h_i .

Además, el funcional de tipo producto con un parámetro para cada regla tiene un efecto equivalente a utilizar reglas con pesos [6]. El empleo de pesos que acompañan a cada regla de la base de conocimiento consiste en añadir un valor adicional que tiene el significado de indicar qué grado de importancia debe tener esa regla en el proceso de inferencia. Generalmente el objetivo de esta ampliación es mejorar la precisión.

3 MECANISMO PARA LA MEJORA DE LA INTERPRETABILIDAD EN LOS SISTEMAS DIFUSOS LINGÜÍSTICOS CON DEFUZZIFICACIÓN ADAPTATIVA

En esta sección, proponemos introducir un mecanismo para mejorar la interpretabilidad en sistemas difusos lingüísticos con defuzzificación adaptativa, basado en una métrica.

Tal y como se ha indicado en la sección anterior, la utilización de métodos de defuzzificación adaptativos con funcional de tipo producto tiene un efecto equivalente al empleo de reglas con pesos [6]. Estos pesos en las reglas tienen un efecto negativo en la interpretabilidad [3, 17] ya que extienden la estructura del sistema. Por este motivo, y con objeto de reducir este efecto negativo, la métrica que se utilice debe tenerlo en cuenta. En este trabajo se ha optado por utilizar el número de reglas final como medida útil en este sentido, así como el número de reglas con peso, considerando que ambos valores son mejores

cuando menores sean. En las siguientes subsecciones se describirán con más detalle estas métricas.

3.1. NÚMERO DE REGLAS FINAL ($\#R_F$)

Esta métrica está basada en la idea de que aquellos pesos de las reglas próximos a 0 suponen una baja influencia de dicha regla y por tanto, podrían señalar una regla prescindible, continuando el sistema evolutivo con el aprendizaje del resto del sistema sin ella. Eliminar la regla en la fase de aprendizaje permite al modelo evolutivo ajustar el resto de pesos para eliminar dicha regla. Si se llevase a cabo la eliminación de las reglas con poco peso sólo al final del proceso, la precisión del sistema, en la mayoría de los casos se vería perjudicada por la naturaleza de los sistemas difusos lingüísticos.

Para aplicar dicho concepto los parámetros del funcional tipo producto actuarán en el rango $[0,1]$ (véase expresión 2). La expresión de dicho índice será:

$$\#R_F = \#R - (\text{número de pesos próximos a } 0) \quad (3)$$

donde $\#R$ es el número de reglas iniciales del sistema.

3.2. NUMERO DE REGLAS CON PESO ($\#R_P$)

Esta métrica en cambio, se basa en la idea de que los valores de los pesos próximos a 1 son aquellos en los que la regla tiene importancia y podría utilizarse sin necesidad de ninguna ponderación o peso, y por tanto eliminar este valor reduciendo así la complejidad del sistema. Cuanto mayor sea el número de reglas ponderadas por un peso, mejor será la interpretabilidad del sistema. La expresión de dicho índice será:

$$\#R_P = \#R - (\text{número de pesos próximos a } 1) \quad (4)$$

3.3. INTRODUCCION DE LOS INDICES EN EL PROCESO DE APRENDIZAJE

Como se ha apuntado anteriormente, la introducción de dichos índices en el proceso de aprendizaje es un elemento importante, ya que el sistema debe readaptarse tanto a la eliminación de reglas con baja aportación, como a la de algunos de los pesos en las reglas más importantes.

El mecanismo que se utiliza en este trabajo es proporcionar al sistema unos umbrales fijos para ambos índices, los cuales son tenidos en cuenta en la evaluación del sistema difuso. Los pesos que superen un valor determinado próximo a 1, serán evaluados como 1, mientras que los valores del peso que desciendan por debajo de un umbral cercano a 0, serán reglas no consideradas.

4 MODELO MULTI OBJETIVO

En esta Sección se presenta el algoritmo multiobjetivo propuesto donde se introducirá el mecanismo de mejora de interpretabilidad en sistemas con defuzzificación adaptativa descrito en la Sección anterior.

Los objetivos que utiliza el modelo multiobjetivo son tres: minimizar los dos índices de interpretabilidad, $\#R_F$ y $\#R_P$ además de minimizar el habitual error cuadrático medio del sistema para la precisión. En esta ocasión se utiliza un modelo evolutivo basado en el conocido NSGA-II [10]. En las próximas subsecciones se expondrán los principales componentes de dicho algoritmo.

3.1. ESQUEMA DE CODIFICACIÓN Y POBLACIÓN INICIAL

En este trabajo se utiliza un esquema de codificación real, siendo m parámetros α_i , uno para cada R_i de la BR. Cada uno toma valores en el intervalo $[0,1]$.

$$C = (\alpha_1, \dots, \alpha_m) \mid \alpha_i \in \{0, 1\}$$

La población inicial se obtiene de la siguiente forma: Un individuo de la población inicial tiene todos los genes inicialmente fijados a 1 con el fin de partir en el proceso evolutivo con todas las reglas sin peso. El resto de individuos de la población inicial se crean aleatoriamente.

4.2 OBJETIVOS Y UMBRALES DEL MECANISMO DE MEJORA

Tal y como hemos comentado anteriormente, en este algoritmo hemos usado estos objetivos a minimizar:

- El índice de interpretabilidad ($\#R_F$) que representa el número de reglas finales en el sistema.
- El índice de interpretabilidad ($\#R_P$) que representa el número de reglas que actúan con peso.
- El Error Cuadrático Medio (ECM) que mide la precisión del sistema:

$$ECM(S[i]) = \frac{\frac{1}{2} \sum_{k=1}^N (y_k - S[i](x_k))^2}{N} \quad (5)$$

donde $S[i]$ es el modelo difuso. Esta medida precisa del empleo de un conjunto de datos de evaluación constituido por N pares de datos numéricos $Z_k = (x_k, y_k)$, $k=1, \dots, N$, siendo x_k los valores de las variables de entrada e y_k los correspondientes a sus salidas asociadas.

En cuanto a los umbrales considerados para los índices $\#R_F$ y $\#R_P$, se han seleccionado los siguientes:

- Cuando el valor del parámetro es ≥ 0.9 se considera que dicho parámetro es 1 y por tanto dicha regla no tiene peso.

- Cuando el valor del parámetro sea ≤ 0.1 , evaluaremos dicho parámetro como 0 y dicha regla será eliminada.

Ambos umbrales han sido escogidos empíricamente realizando distintas pruebas.

4.3 ALGORITMO GENÉTICO MULTI OBJETIVO

El algoritmo genético elegido, basado en el NSGA-II [10], es uno de los AGMO más conocidos y utilizados en la literatura para la resolución de problemas multiobjetivos. La población descendiente se genera desde la población actual a través de la selección, el cruce y la mutación. La siguiente generación se construye pues desde la población actual y la descendiente hasta que se cumpla la condición de parada, en este trabajo, por número de evaluaciones.

El algoritmo NSGA-II posee dos características que lo hacen uno de los principales y más importantes AGMO: Una es la asignación del *fitness* basada en el *ranking* del pareto y en el operador de *crowding*, y la otra es el procedimiento de actualización de cada generación a través del elitismo.

5 ESTUDIO EXPERIMENTAL

Para analizar el funcionamiento del método propuesto, se han escogido dos problemas reales con diferentes complejidades (diferente número de variables y datos). Dichos problemas están disponibles en <http://www.keel.es/> [2]:

- La estimación del cálculo de los costes óptimos teóricos de la línea de media tensión en núcleos urbano (ELE).
- Predicción de la edad del Abalone (ABA).

En ambos casos, se ha empleado el algoritmo de Wang y Mendel [16] para obtener la BR inicial. Una vez generado, el algoritmo propuesto fue aplicado. Los Métodos considerados para el estudio experimental fueron:

- D método con defuzzificación adaptativa utilizando un algoritmo genético mono-objetivo y considerando sólo precisión como único objetivo [8].
- D₁ es el que utiliza la defuzzificación adaptativa con la mejora propuesta en la interpretabilidad y aprendizaje multiobjetivo considerando tres los objetivos, minimizar ECM #R_F y #R_P.

Se ha considerado un modelo experimental 5-fold cross-validation. Los métodos han sido ejecutados 6 veces, mostrándose la media de las 30 ejecuciones.

Las particiones lingüísticas consideradas están formadas por términos lingüísticos con forma triangular, 5 para el problema eléctrico y 3 para el Abalone. Como operador de conjunción e implicación se ha utilizado la t-norma del mínimo. Los valores de los parámetros utilizados en todos los experimentos presentados son: 200 de longitud de la población, 200 de longitud de la descendiente, 1000 generaciones y 0.2 como probabilidad de mutación por cromosoma.

5.1 RESULTADOS Y ANÁLISIS

En el caso de los métodos con enfoque multi-objetivo, las medias han sido obtenidas considerando la solución más precisa de cada pareto obtenido. Los algoritmos propuestos se han comparado con varios métodos guiados por un único objetivo.

La Tabla 1 muestran, a nivel de referencia, los ECMs de los conjuntos de entrenamiento y prueba en cada aplicación, donde #R representa el número de reglas, ECM_{ent} y ECM_{pru} representan el error obtenido en entrenamiento y en prueba respectivamente. Este método obtiene la BR inicial sobre la que D, y D₁ introducirán sus pesos. WM toma para el índice de #R_P el más alto nivel que representa que todas las reglas no contienen pesos.

Tabla 1: ECMs de referencia

Problema	#R	ECM _{ent}	ECM _{pru}
ELE	65	56136	56359
ABA	68	8.407	8.422

Los resultados obtenidos con los métodos de defuzzificación adaptativos se muestran en la Tablas 2 donde adicionalmente se añade dos columnas con los valores de #R_F (representa el número de reglas finales) y #R_P (el número de reglas con peso) como índices de interpretabilidad, y dos columnas denominadas t-test que son los resultados de aplicar un test t-student (con 95% de confianza) al mejor resultado medio de la correspondiente columna comparado 1 a 1 con el resto de resultados medios para cada problema. La interpretación de ellas es:

* Indica el resultado con la mejor media.

+ Indica un comportamiento significativamente peor que el mejor.

= Indica un comportamiento estadísticamente sin diferencias significativas con el mejor.

Observando los resultados presentados en la Tabla 2 podemos destacar:

- El modelo que considera las medidas de interpretabilidad obtiene el mejor comportamiento en cuanto a interpretabilidad respecto al que no, D, en ambos problemas. D₁ mejora tanto en #R_F como en #R_P, con descensos significativos en el número de

reglas, y siendo particularmente interesante en ambos problemas que sólo aproximadamente la mitad de las reglas finales tienen pesos.

- En cuanto a la precisión, en el problema del ABA se obtiene similares niveles de precisión tanto en entrenamiento como prueba, mientras que en el problema ELE la precisión es algo mayor en el modelo mono-objetivo orientado a precisión, señalando por tanto la necesidad de mejoras en el modelo multiobjetivo para alcanzar los mismos niveles de precisión.

En la Figura 1 se muestra la superficie de la frontera del pareto del algoritmo D_1 del problema ABA. Se puede observar que es bastante extenso y que no existe sobreaprendizaje en los resultados obtenidos con el modelo propuesto.

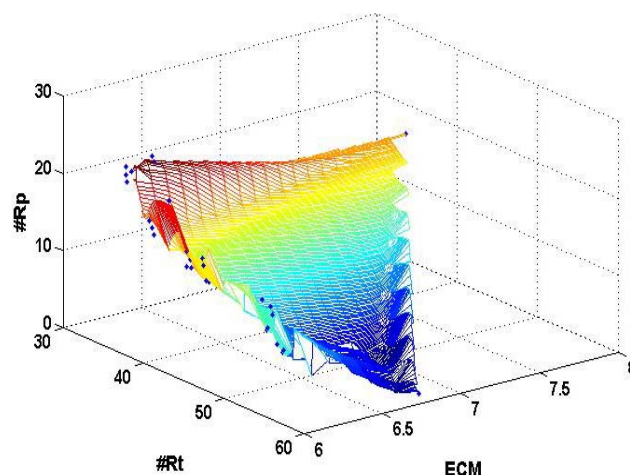


Figura 1: Frontera de pareto obtenida en el problema ABA.

Tabla 2: Resultados obtenidos en los dos problemas

Problema	Método	ECM _{ent}	t-test	ECM _{pru}	t-test	#R _F	t-test	#R _P	t-test
ELE	D	32791	*	35362	*	65	+	65	+
	D_1	34921	+	37695	+	45,97	*	19,63	*
ABA	D	4,823	=	4,826	*	68	+	68	+
	D_1	4,918	=	4,924	=	31,57	*	17,70	*

6 CONCLUSIONES

La defuzzificación adaptativa que utiliza factores de ponderación en el grado de emparejamiento a través de un producto es un mecanismo sencillo para mejorar la precisión de los modelos difusos lingüísticos, el cual, tiene como desventaja el aumento de la complejidad del sistema, resultado del efecto de añadir distintos factores o pesos en cada una de las reglas de la base de conocimiento. Para reducir este efecto, en este trabajo introducimos un mecanismo para reducir complejidad basado en utilizar umbrales sobre los citados pesos, de modo que niveles muy altos eliminan el peso de esa regla y niveles muy bajos eliminan la regla por completo. Aprendiendo mediante algoritmos evolutivos multiobjetivo los parámetros tratando de obtener la mayor precisión, el menor número de reglas y el menor número de reglas con pesos, se obtienen resultados interesantes por la reducción en la complejidad, manteniendo un buen nivel de precisión, como se ha visto experimentalmente.

No obstante, observando la Figura 1, consideramos ampliar el estudio tratando de centrar el algoritmo evolutivo multiobjetivo en la zona más interesante de la frontera del pareto. Además, como trabajo futuro se desea tener en cuenta dos aspectos:

- La elección de los umbrales empleados podría ser más eficaz si se eligieran en base al problema y su dominio.
- Mejorar la propuesta de interpretabilidad teniendo en cuenta otros índices como son la inconsistencia, redundancia y similitud de las reglas.

Agradecimientos

Este trabajo está soportado por el proyecto TIN2008-06681-C06-06 del Ministerio de Educación y Ciencia y el proyecto P07-TIC-03179 de la Consejería de Innovación Ciencia y Empresa de la Junta de Andalucía.

Referencias

- [1] J. Alcalá-Fdez, F. Herrera, F. Márquez, A. Peregrín. Increasing Fuzzy Rules Cooperation Based On Evolutionary Adaptive Inference Systems. *International Journal of Intelligent Systems* 22:9, Pág. 1035-1064, 2007.
- [2] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesús, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, and F. Herrera. Keel: A software tool to assess evolutionary

- algorithms for data mining problems. *Soft Computing*, 13:3, Pág. 307–318, 2009.
- [3] J.M. Alonso, L. Magdalena. A Conceptual Framework for Understanding Fuzzy Systems. *Proceedings of IFSA World Congress (IFSA-EUSFLAT'09)* Pág. 119-124, Lisbon, Portugal, 2009
- [4] J. Casillas, O. Cordón, F. Herrera, L. Magdalena(Eds). Interpretability issues in fuzzy modeling, volume 128 of Studies in Fuzziness and Soft Computing. *Springer*, 2003.
- [5] J. Casillas, O. Cordón, F. Herrera, L. Magdalena(Eds). Accuracy improvements in linguistic fuzzy modeling. *Springer-Verlag*, 2003.
- [6] J.S. Cho, D.J. Park. Novel fuzzy logic control based on weighting of partially inconsistent rules using neural network. *Journal of Intelligent and Fuzzy Systems*, 8, Pág. 99-100, 2000
- [7] C. A. Coe llo, D. A. Van Veldhuizen, and G. B. Lamont, editors. Evolutionary algorithms for solving multi-objective problems. *Kluwer Academic Publishers*, 2002.
- [8] O. Cordón , F. Herrera , F. A. Márquez, A. Peregrín. A study on the evolutionary adaptive defuzzification methods in fuzzy modelling. *International Journal of Hybrid Intelligent Systems*. 1:1. Pág. 36–48, 2004.
- [9] K. Deb. Multi-objective optimization using evolutionary algorithms. *John Wiley & Sons*, NY, USA, 2001.
- [10] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation* 6:2, Pág.182-197, 2002.
- [11] M J Gacto, R Alcalá, F Herrera. Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Computing* 13. Pág. 419-436, 2009.
- [12] H. Ishibuchi, T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets Syst* 141:1. Pág. 59–88, 2004.
- [13] A.A. Márquez, F.A. Márquez, A. Peregrín. Cooperation between the Inference System and the Rule Base by using Multi-objective Genetic Algorithms. *Proceedings of Third International of Hybrid Artificial Intelligence System,(HAIS'08)*. Pág. 739-746 Burgos, Spain, 2008.
- [14] A.A. Márquez, F.A. Márquez, A. Peregrín. Rule Base and Inference System Cooperative Learning of Mamdani Fuzzy Systems with Multiobjective Genetic Algorithms. *Proceedings of IFSA World Congress (IFSA-EUSFLAT'09)* Pág. 1045-1050, Lisbon, Portugal, 2009.
- [15] F.A. Márquez, A. Peregrín y F. Herrera. Cooperative Evolutionary Learning of Fuzzy Rules and Parametric Aggregation Connectors for Mamdani Linguistic Fuzzy Systems. *IEEE Transactions on Fuzzy Systems*. 15:6 Pág. 1162-1178, 2007.
- [16] L.X. Wang, J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man, and Cybernetics* 22:6, Pág.1414-1427, 1992.
- [17] M. Zhou and J. Q. Gan. Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets and Systems*, 159:23. Pág. 3091–3131, 2008.

Criterios para el empleo de transformaciones ortogonales en la mejora de modelos difusos. Aplicación a modelos basados en ART

M.I. Rey¹ M. Galende² G.I. Sainz^{2,3}

¹ INDOMAUT, S.L. Pol.Ind.San Cristóbal. Valladolid (España), i.rey@indomaut.com

² Fundación CARTIF, Parque Tecnológico de Boecillo. Valladolid (España) {margal,gresai}@cartif.es

³ Dpto.Ingeniería de Sistemas y Automática, E.I.Industriales Universidad de Valladolid (España), gresai@eis.uva.es

Resumen

El propósito de este trabajo es la mejora de modelos difusos a través de transformaciones ortogonales, tratando de encontrar un equilibrio entre la precisión del modelo y su interpretabilidad. El modelado difuso preciso genera modelos con una buena precisión pero no tiene en cuenta otros aspectos básicos de la lógica difusa, como la interpretabilidad de las reglas, de ahí que estos modelos contengan un excesivo número de reglas, redundancia, incoherencia, etc. Así, el procedimiento de mejora de la simplificación se realiza intentando mantener las ventajas del modelo original (precisión) y tratando de mejorar otros aspectos bajo la idea de interpretabilidad: compactitud, distinguibilidad, etc. Para ello, se utiliza una metodología para simplificar y ordenar la base de reglas del sistema difuso haciendo uso de las transformaciones ortogonales y definiendo unos criterios que rijan el proceso de mejora de los modelos difusos. Esa propuesta se ha probado en dos casos de estudio: Motor CC y BoxJenkins.

Palabras Clave: Modelado difuso, Balance Precisión-Interpretabilidad, Transformaciones Ortogonales

1 INTRODUCCIÓN

En la actualidad es muy común la aplicación de sistemas lógico difusos en problemas reales [15]. El estudio de un sistema difuso se suele hacer a través de su conjunto de reglas difusas, que nos "describen" el comportamiento de dicho sistema, teniendo en cuenta la *Precisión* y la *Interpretabilidad*.

Si vemos modelos difusos propuestos en la literatura científica, analizados bajo estos puntos de vista, la gran mayoría de ellos muestran una buena precisión pero muy mala interpretabilidad. A partir de aquí se definen dos tipos de modelado difuso [1, 5, 7]:

- Modelado difuso preciso (*Precise Fuzzy Modeling*), donde la precisión es el principal objetivo del proceso pero la interpretabilidad es mala. Este tipo de modelado es el más habitual en aplicaciones técnicas.
- Modelado difuso lingüístico (*Linguistic Fuzzy Modeling*), donde la interpretabilidad es muy buena pero la precisión es muy pobre.

En nuestro caso, partimos de modelados precisos y queremos conseguir modelados lingüísticos. Para reducir la complejidad del modelo se utilizan transformaciones ortogonales y a continuación se aplican un conjunto de criterios que se definen en este artículo, teniendo en cuenta aspectos basados en la interpretabilidad como la redundancia o incoherencia. Se considera como ejemplo un sistema neuro-difuso FasART (Fuzzy Adaptive System Adaptive Resonance Theory based) con las ventajas que tiene un algoritmo de modelado ya existente (se podría utilizar cualquier otro) y se estudian dos casos: Motor CC y Box-Jenkins.

Cabe destacar que se ha aplicado a modelos Mandani ya que, si revisamos la literatura científica vemos que la mayoría de los trabajos están basados en TSK.

El artículo está organizado del siguiente modo: en primer lugar, una breve descripción del modelado difuso junto con transformaciones ortogonales, los criterios propuestos y las definiciones de precisión e interpretabilidad. En segundo lugar una breve descripción del sistema neuro-difuso FasART. En tercer lugar la metodología usada en el artículo, la descripción de los casos de estudio y los principales resultados obtenidos. Y por último las conclusiones más importantes a las que se ha llegado.

2 MODELADO DIFUSO: PRECISIÓN VS. INTERPRETABILIDAD

Cuando se revisan los algoritmos de modelado difuso, se encuentra el *modelado difuso preciso*, generado por datos, cuyas reglas están compuestas por variables borrosas en las que prima la precisión y el *modelado difuso lingüístico*, guiado por el conocimiento de expertos, que genera modelos difusos cuyas reglas están compuestas por variables lingüísticas en las que prima la interpretabilidad. El modelado difuso preciso intenta obtener mejor nivel de interpretabilidad mejorando este aspecto a partir de las reglas difusas [4, 7], mientras que la aproximación lingüística intenta mejorar la precisión [5, 7].

En general, la mejora de la interpretabilidad está basada en la reducción y simplificación de las reglas y sus componentes. Una de las principales aproximaciones que aparecen en la literatura científica es la simplificación basada en transformaciones ortogonales que es lo que nos ocupa en este trabajo [4, 18, 19, 21]. Se consideran tres métodos de transformaciones ortogonales [11]: SVD-QR (QR Singular Value Decomposition) [20, 21], QR-P (Pivoted QR Decomposition) [20, 21] y OLS (Orthogonal Least Square) [8, 20, 21], y se incorporan varios criterios para selección de reglas que nos llevan a un modelo reducido. Para comprobar estos criterios se utilizan el error del modelo y la interpretabilidad de las reglas teniendo en cuenta la redundancia, incoherencia y similitud de las mismas.

2.1 Transformaciones Ortogonales

Lo principal de las transformaciones ortogonales es descubrir la dimensionalidad intrínseca de los datos [11, 18], que en el caso de modelos difusos se traduce en encontrar las reglas más relevantes del sistema. Se consideran dos grupos:

- Métodos de búsqueda de rango, donde el rango efectivo se obtiene a partir de los valores singulares de la matriz de disparo: SVD-QR [20, 21].
- Métodos de evaluación individual, donde la contribución de cada regla se evalúa a partir del orden de las mismas: OLS [8, 20, 21].

El método QR-P [20, 21] se puede considerar como un método intermedio que genera un ordenamiento de las reglas con el cálculo de SVD.

2.2 Criterios

Un aspecto crítico de las aproximaciones ortogonales es determinar el número adecuado de valores singula-

res correctos que escoge cada algoritmo y por tanto, el número de reglas asociadas.

En este trabajo se proponen tres criterios:

1. Consiste en eliminar las reglas no relevantes: se toma el conjunto de reglas del modelo y se eliminan aquellas cuyos valores singulares o varianzas sean menores a un valor umbral, $\beta_1 \leq 8\%$.

$$\begin{aligned} ValorSingular/Varianza_{Regla_j} &\leq \beta_1 \\ \Rightarrow Regla_j &\notin ModeloReducido \end{aligned} \quad (1)$$

2. Se trata de encontrar un "gap" entre las reglas y asegurarse de que las reglas seleccionadas tienen suficiente información: se comprueba si un valor singular o varianza es el doble del siguiente y se ve si la suma de los valores considerados es superior a un determinado valor umbral, $\beta_2 \geq 20\%$.

$$\begin{aligned} \text{For } (Regla_1 \dots Regla_n) \\ ValorSing/Var_n &\geq 2 * ValorSing/Var_{n+1} \\ \sum_{j=1}^{j=n} ValorSingular/Varianza_{Regla_j} &\geq \beta_2 \\ \Rightarrow Regla_1, \dots, Regla_n &\in ModeloDifuso \end{aligned} \quad (2)$$

3. Se trata de seleccionar las reglas mas relevantes: se consideran las reglas cuyos valores singulares o varianzas son mayores a un valor umbral, $\beta_3 \geq 80\%$.

$$\begin{aligned} ValorSingular/Varianza_{Regla_j} &\geq \beta_3 \\ \Rightarrow Regla_j &\in ModeloReducido \end{aligned} \quad (3)$$

Los valores de estos umbrales, β_1 , β_2 y β_3 están analizados en la sección de experimentos de este trabajo a partir de los valores de precisión e interpretabilidad.

2.3 Precisión

Para la precisión se utiliza el error cuadrático medio, medida ampliamente aceptada para medir la exactitud en problemas de modelado:

$$ECM = \frac{1}{N} \sum_{i=1}^N (Y_i - Y'_i)^2 \quad (4)$$

2.4 Interpretabilidad

En este artículo la interpretabilidad está basada en los conceptos de completitud y distinguibilidad [6, 12, 13, 14, 16], expresados como sigue:

$$\begin{aligned} \gamma_1 &< Similitud(A_i, A_{i+1}) < \gamma_2 \\ Similitud(A_i, A_{i+1}) &= \frac{|A_i \cap A_{i+1}|}{|A_i \cup A_{i+1}|} = \\ &= \frac{\sum_i^j [\mu_{a_i}(x_j) \wedge \mu_{a_{i+1}}(x_j)]}{\sum_i^j [\mu_{a_i}(x_j) \vee \mu_{a_{i+1}}(x_j)]} \end{aligned} \quad (5)$$

Si γ_1 es pequeña pero no cero, la completitud está garantizada y si γ_2 tiene un valor superior a 1, se obtiene una buena distinguibilidad. Por otra parte, también hay que tener en cuenta la incoherencia y la redundancia entre las reglas (en [9] se da una definición de estos índices). En el caso que nos ocupa, estos índices son usados para evaluar los criterios propuestos y conseguir una reducción de las reglas a partir de las transformaciones ortogonales. Cuando una regla tiene varios antecedentes, la similitud entre dos reglas se puede calcular de diferentes maneras: la suma normalizada de la similitud de los antecedentes (eq. 6), el máximo de la similitud de los antecedentes (eq. 7) o el mínimo de la similitud en los antecedentes (eq. 8) (expresiones también usadas cuando se usan consecuentes):

$$\begin{aligned} S_A(R_i, R_j) &= \frac{\sum_k S_{A_k}(R_i, R_j)}{NumAntecedentes} \\ \forall 1 \leq i < j \leq NumReglas \\ \forall 1 \leq k \leq NumAntecedentes \end{aligned} \quad (6)$$

$$\begin{aligned} S_A(R_i, R_j) &= \max_k S_{A_k}(R_i, R_j) \\ \forall 1 \leq i < j \leq NumReglas \\ \forall 1 \leq k \leq NumAntecedentes \end{aligned} \quad (7)$$

$$\begin{aligned} S_A(R_i, R_j) &= \min_k S_{A_k}(R_i, R_j) \\ \forall 1 \leq i < j \leq NumReglas \\ \forall 1 \leq k \leq NumAntecedentes \end{aligned} \quad (8)$$

Finalmente, la similitud de todas las reglas difusas se calcula como el número de pares de reglas ($Card$) cuya similitud es mayor que un determinado umbral β_S (eq. 9) o como la media aritmética de la similitud de los antecedentes (eq. 10):

$$Similitud = \frac{Card(S_A(R_i, R_j) > \beta_S)}{(NumReglas-1)!} \quad (9)$$

$$Similitud = Media(S_A(R_i, R_j)) \quad (10)$$

Desde el punto de vista del balance precisión-interpretabilidad, las expresiones más utilizadas son el Máximo y la Media.

2.5 Sistema Neuro-Difuso FasART

El modelo FasART [2, 17] es un modelo neuro-difuso basado en la teoría de resonancia adaptativa (ART): Fuzzy ARTMAP [3]. FasART introduce una equivalencia entre la función de activación de cada neurona FasART y una función de pertenencia. FasART sería el equivalente a un sistema basado en reglas difusas Mamdani con fuzificación por singleton, inferencia por producto y defuzificación a través de la media de los centros de los conjuntos difusos. Se puede encontrar una descripción completa de este modelo en [2, 17].

3 METODOLOGÍA

En este artículo se propone un método en dos pasos:

1. Generación de un modelo difuso, usando cualquiera de los algoritmos descritos en la bibliografía científica para modelado difuso preciso o lingüístico. En nuestro caso se ha utilizado el sistema neuro-difuso FasART pero se podría utilizar cualquier otro.
2. Mejora del modelo difuso, a través del conjunto de reglas y funciones de pertenencia, obteniendo un modelo más compacto y más interpretable. En este trabajo se utilizan las transformaciones ortogonales y criterios descritos en 2.2 para tal fin.

4 EXPERIMENTOS Y RESULTADOS

El método propuesto se valida en dos casos de estudio:

1. Motor eléctrico CC Admira DR-300 ¹ con tres entradas (Tensión (U), ΔU , Intensidad (I_{arm})) y una salida (Velocidad de Giro (ω)).
2. Datos Box-Jenkins ² con seis entradas ($x(k-1)$, $x(k-2)$, $x(k-3)$, $y(k-1)$, $y(k-2)$, $y(k-3)$) y una salida ($y(k)$) según lo descrito en [10].

Para cada caso de estudio tenemos dos modelos, uno compacto y otro complejo, para estimar la bondad de la propuesta realizada con los cuatro casos. De este modo y de acuerdo con la metodología propuesta, *en primer lugar* se generan los modelos difusos para el Motor CC y Box-Jenkins, ambos generados con FasART, y *en segundo lugar* se mejoran estos modelos usando las transformaciones ortogonales y los criterios basados en interpretabilidad. Para ver esta interpretabilidad, se define la siguiente función de coste donde cada parámetro se toma normalizado:

$$\begin{aligned} F_1 &= Error + NumReglas + SimilitudReglas \\ &\quad + IncoherenciaReglas + RedundanciaReglas \\ &\quad + NumParticionesCompletas(Cobertura) \\ &= E + NR + S + I + R + C \end{aligned}$$

Por otra parte, los parámetros FasART considerados a la hora de generar ambos modelos son:

Caso de estudio A. Motor CC:

1. Modelo 1A, con parámetros FasART $\rho_A = \rho_B = 0.9$ y $\gamma_A = \gamma_B = 11$, y prestaciones iniciales: NR: 16, E: 0.0022, S: 0.0138, R: 0.0077, I: 0.0000 y Cobertura completa.

¹http://www.amira.de/dr300_engl.html

²<http://www.stat.wisc.edu/~reinsel/bjr-data/gas-furnace>

2. Modelo 1B, generado con parámetros FasART $\rho_A = \rho_B = 1$ y $\gamma_A = \gamma_B = 11$, y prestaciones iniciales: NR: 505, E: 0.0045, S: 0.3142, R: 0.0000, I: 0.0000 y Cobertura completa.

Caso de estudio B. Horno de gas Box-Jenkins:

1. Modelo 2A, con parámetros FasART $\rho_A = \rho_B = 0.6$ y $\gamma_A = \gamma_B = 3$, y prestaciones iniciales: NR: 31, E: 0.0030, S: 0.0688, R: 0.0193, I: 0.0494 y Cobertura completa.
2. Modelo 2B, con parámetros FasART $\rho_A = \rho_B = 1$ y $\gamma_A = \gamma_B = 12$, y prestaciones iniciales: NR: 293, E: 0.00002, S: 0.00009, R: 0.00002, I: 0.0000 y Cobertura completa.

A partir de aquí, los mejores resultados obtenidos teniendo en cuenta transformaciones ortogonales, criterios y la función de coste son:

4.1 CRITERIO I

Para testear β_1 se ha tomado el rango 8% – 20% para todos los métodos de transformaciones ortogonales citados (SVD-QR, QRP y OLS), y se ha evaluado la función de coste F_1 para los dos casos de estudio. En general los mejores resultados se han obtenido para $\beta_1 = 8\%$ aunque hay alguna excepción con $\beta_1 = 16\%$:

4.1.1 Modelo 1A

- **SVD-QR**: NR: 16 $\rightarrow$ 6, E: 0.0022 $\rightarrow$ 0.0065, S: 0.3142 $\rightarrow$ 0.2866. No Redundancia ni Incoherencia y la Cobertura es completa.
- **QRP**: NR 16 $\rightarrow$ 9, E: 0.0022 $\rightarrow$ 0.0043, S: 0.3142 $\rightarrow$ 0.3037. No Redundancia ni Incoherencia y la Cobertura es completa.
- **OLS**: NR: 16 $\rightarrow$ 5, E: 0.0022 $\rightarrow$ 0.0080, S: 0.3142 $\rightarrow$ 0.3044. La Redundancia, Incoherencia y Cobertura siguen como en el modelo inicial.

4.1.2 Modelo 1B

- **SVD-QR**: NR: 505 $\rightarrow$ 190, E: 0.0045 $\rightarrow$ 0.0076, S: 0.0138 $\rightarrow$ 0.123, R: 0.0077 $\rightarrow$ 0.0050. No Incoherencia y Cobertura se mantiene completa.
- **QRP**: NR: 505 $\rightarrow$ 201, E: 0.0045 $\rightarrow$ 0.0070, S: 0.0138 $\rightarrow$ 0.0117, R: 0.0077 $\rightarrow$ 0.0049. No Incoherencia y Cobertura se mantiene completa.
- **OLS**: NR: 505 $\rightarrow$ 6, E: 0.0045 $\rightarrow$ 0.3191, S: 0.0138 $\rightarrow$ 0.029, R: 0.0077 $\rightarrow$ 0.0021. No Incoherencia y Cobertura completa excepto en dos casos.

4.1.3 Modelo 2A

- **SVD-QR**, NR: 31 $\rightarrow$ 19, E: 0.0030 $\rightarrow$ 0.00361, S: 0.0688 $\rightarrow$ 0.0643, R: 0.0193 $\rightarrow$ 0.0175, I: 0.0494 $\rightarrow$ 0.0467. Las particiones se mantienen completas.
- **QRP**, NR: 31 $\rightarrow$ 19, E: 0.0030 $\rightarrow$ 0.0036, S: 0.0688 $\rightarrow$ 0.0643, R: 0.0193 $\rightarrow$ 0.0175, I: 0.0494 $\rightarrow$ 0.0468. La Cobertura permanece completa.
- **OLS**, NR: 31 $\rightarrow$ 18, E: 0.0030 $\rightarrow$ 0.0060, S: 0.0688 $\rightarrow$ 0.0182, R: 0.0193 $\rightarrow$ 0.0175, I: 0.0494 $\rightarrow$ 0.0468. Las particiones siguen completas.

4.1.4 Modelo 2B

- **SVD-QR**, NR: 293 $\rightarrow$ 289, E: 0.00002 $\rightarrow$ 0.00003, S: 0.1423 $\rightarrow$ 0.1397, R: 0.00002 $\rightarrow$ 0.00002. No Incoherencia y Cobertura completa.
- **QRP**, NR: 293 $\rightarrow$ 288, E: 0.00002 $\rightarrow$ 0.00002, S: 0.00009 $\rightarrow$ 0.0001, R: 0.00002 $\rightarrow$ 0.00002. No Incoherencia y Cobertura completa.
- **OLS**, NR: 293 $\rightarrow$ 180, E: 0.00002 $\rightarrow$ 0.0252, S: 0.00009 $\rightarrow$ 0.00012, R: 0.00002 $\rightarrow$ 0.0000. No Incoherencia y Cobertura completa.

En general este criterio ha permitido simplificar los modelos difusos: El número de reglas ha disminuido y la interpretabilidad se ha incrementado. Esto supone una subida del error pero como los valores de error siguen siendo muy pequeños no se tiene en cuenta. En el caso del *Modelo 2B* los resultados han sido diferentes al resto de los modelos: el número de reglas no se ha decrementado mucho con SVD-QR y QRP mientras que con OLS los resultados son similares a los *Modelos 1A, 1B y 2B*.

4.2 CRITERIO II

Este criterio es el más similar al "gap" entre valores singulares, o equivalentes, descritos en otros trabajos. Para los *Modelos 1A y 1B*, la reducción en el número de reglas ha sido extremo lo cuál no está bien desde el punto de vista lingüístico. En el caso de los *Modelos 2A y 2B* la reducción ha sido mínima. Para un rango de β_2 de 20% – 40% los mejores resultados se han obtenido para $\beta_2 = 20\%$ y son:

4.2.1 Modelo 1A

- **SVD-QR**, NR: 16 $\Rightarrow$ 1, E: 0.0022 $\Rightarrow$ 0.4761; S: 0.4000 $\Rightarrow$ 0.0000; R: 0.0417 $\Rightarrow$ 0.0000, I: 0.2083 $\Rightarrow$ 0.0000; Cobertura completa.
- **QRP**, NR: 16 $\Rightarrow$ 1, E: 0.0022 $\Rightarrow$ 0.4761; S: 0.4000 $\Rightarrow$ 0.0000; R: 0.0417 $\Rightarrow$ 0.0000, I: 0.2083 $\Rightarrow$ 0.0000; Cobertura completa.

- **OLS**, no se han obtenido resultados.

4.2.2 Modelo 1B

- **SVD-QR, QRP** no se han obtenido resultados.
- **OLS**, NR: $505 \Rightarrow 1$, E: $0.0045 \Rightarrow 0.4761$; S: $0.0138 \Rightarrow 0.0000$; R: $0.0077 \Rightarrow 0.0000$, I: $0.2083 \Rightarrow 0.0000$; Cobertura completa.

4.2.3 Modelo 2A

Los resultados obtenidos para **SVD-QR, QRP** and **OLS** han sido iguales: NR: $31 \Rightarrow 30$, E: $0.0030 \Rightarrow 0.0029$; S: $0.0000 \Rightarrow 0.0000$; R: $0.0417 \Rightarrow 0.0000$, I: $0.2083 \Rightarrow 0.0000$; Cobertura completa.

4.2.4 Modelo 2B

- **SVD-QR, QRP**, no obtenido resultados.
- **OLS**, NR: $293 \Rightarrow 290$, E: $0.00002 \Rightarrow 0.00003$; S: $0.0000 \Rightarrow 0.0000$; R: $0.0000 \Rightarrow 0.0000$, I: $0.2083 \Rightarrow 0.0000$; Cobertura completa.

4.3 CRITERIO III

Para testear β_3 se ha tomado el rango 60% – 90% y los mejores resultados se han obtenido para $\beta_3 = 80\%$ donde podemos ver que ha disminuido el número de reglas aumentando la interpretabilidad con valores de error totalmente aceptables:

4.3.1 Modelo 1A

- **SVD-QR**, NR: $16 \Rightarrow 6$, E: $0.0022 \Rightarrow 0.0065$; S: $0.3142 \Rightarrow 0.2866$; R: $0.0000 \Rightarrow 0.0000$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa excepto una partición.
- **QRP**, NR: $16 \Rightarrow 7$, E: $0.0022 \Rightarrow 0.0060$; S: $0.3142 \Rightarrow 0.0192$; R: $0.0000 \Rightarrow 0.0000$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa excepto una partición.
- **OLS**, NR: $16 \Rightarrow 5$, E: $0.0022 \Rightarrow 0.0080$; S: $0.0415 \Rightarrow 0.0107$; R: $0.0000 \Rightarrow 0.0000$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa.

4.3.2 Modelo 1B

- **SVD-QR**, NR: $505 \Rightarrow 210$, E: $0.0045 \Rightarrow 0.0063$; S: $0.0138 \Rightarrow 0.0111$; R: $0.0077 \Rightarrow 0.0046$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa.
- **QRP**, NR: $505 \Rightarrow 236$, E: $0.0045 \Rightarrow 0.0049$; S: $0.0138 \Rightarrow 0.0098$; R: $0.0077 \Rightarrow 0.0044$, No Incoherencia y Cobertura completa.
- **OLS**, NR: $505 \Rightarrow 192$, E: $0.0045 \Rightarrow 0.0079$; S: $0.0138 \Rightarrow 0.0149$; R: $0.0021 \Rightarrow 0.0019$, I: $0.0077 \Rightarrow 0.0063$; Cobertura completa.

4.3.3 Modelo 2A

- **SVD-QR**, NR: $31 \Rightarrow 19$, E: $0.0030 \Rightarrow 0.0036$; S: $0.0000 \Rightarrow 0.0000$; R: $0.0000 \Rightarrow 0.0000$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa.
- **QRP**, NR: $31 \Rightarrow 17$, E: $0.0030 \Rightarrow 0.0055$; S: $0.1835 \Rightarrow 0.1770$; R: $0.0000 \Rightarrow 0.0000$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa.
- **OLS**, NR: $31 \Rightarrow 18$, E: $0.0030 \Rightarrow 0.0064$; S: $0.0000 \Rightarrow 0.0000$; R: $0.0000 \Rightarrow 0.0000$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa.

4.3.4 Modelo 2B

- **SVD-QR**, NR: $293 \Rightarrow 188$, E: $0.0002 \Rightarrow 0.0276$; S: $0.1508 \Rightarrow 0.1258$; R: $0.0095 \Rightarrow 0.0090$, I: $0.0963 \Rightarrow 0.0773$; Cobertura completa.
- **QRP**, NR: $293 \Rightarrow 204$, E: $0.00002 \Rightarrow 0.0133$; S: $0.000009 \Rightarrow 0.00005$; R: $0.00002 \Rightarrow 0.0000$, I: $0.0000 \Rightarrow 0.0000$; Cobertura completa.
- **OLS**, NR: $293 \Rightarrow 145$, E: $0.00002 \Rightarrow 0.0416$; S: $0.000009 \Rightarrow 0.00005$; No R ni I y C completa.

5 CONCLUSIONES

Este trabajo define un camino a seguir para mejorar el balance precisión-interpretabilidad en modelos basados en reglas difusas, haciendo uso de transformaciones ortogonales, de conceptos de interpretabilidad y del conjunto de criterios descritos para reducir el número de reglas, criterios que son una alternativa a los gráficos en los que se busca un "gap" entre dos valores singulares, que es lo utilizado en otros trabajos.

Esta propuesta se ha aplicado para dos casos de estudio (Motor CC y Box-Jenkins) obteniendo para cada caso dos modelos con diferentes complejidades los cuales se han mejorado posteriormente con tres métodos de transformación ortogonal: SVD-QR, QR-P y OLS.

A continuación se han aplicado los tres criterios definidos de manera que en los criterios I y III, excepto en el modelo 2B, el número de reglas se ha reducido incluso más del 50% lo que lleva a que otros aspectos de la interpretabilidad como similitud, redundancia, etc, también hayan mejorado incluso se hayan eliminado en algunos casos. Esta mejora ha implicado una subida del error pero no se han considerado importante puesto que los valores de error siguen siendo muy pequeños.

Por otra parte, el criterio II ha generado peores resultados desde el punto de vista de la interpretabilidad, ya que, se han puesto unas restricciones muy severas a la hora de buscar el "gap" lo que ha hecho que la reducción del número de reglas sea extremo.

Referencias

- [1] R. Alcalá, J. Alcalá-Fdez, M.J. Gacto, and F. Herrera. Hybrid learning methods to get the interpretability-accuracy trade-off in fuzzy modeling. *Soft Computing*, 10 (9):717–734, 2006.
- [2] J. M. Cano Izquierdo, Y. A. Dimitriadis, E. Gómez Sánchez, and J. López Coronado. Learning from noisy information in FasArt and Fasback neuro-fuzzy systems. *Neural Networks*, 14(4-5):407–425, May 2001.
- [3] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(4):698–713, September 1992.
- [4] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena. *Accuracy Improvements in Linguistic Fuzzy Modelling*, volume 129 of *Studies in Fuzziness and Soft Computing*, chapter Accuracy Improvements to Find the Balance Interpretability-Accuracy in Fuzzy Modeling: An Overview, pages 3–24. Springer-Verlag, Berlin Heidelberg, 2003.
- [5] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena. *Interpretability Issues in Fuzzy Modelling*, volume 128 of *Studies in Fuzziness and Soft Computing*, chapter Interpretability Improvements to Find the Balance Interpretability-Accuracy in Fuzzy Modeling: An Overview, pages 3–22. Springer-Verlag, Berlin Heidelberg, 2003.
- [6] M.Y. Chen and D.A. Linkens. Rule-base self-generation and simplification for data-driven fuzzy models. *Fuzzy sets and systems*, (142):243–265, 2004.
- [7] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, volume 19 of *Advances in Fuzzy Systems - Applications and Theory*. World Scientific, Singapore., 2001.
- [8] S. Destercke, S. Guillaume, and B. Charnomordic. Building an interpretable fuzzy rule base from data using orthogonal least squares-application to a depollution problem. *Fuzzy Sets and Systems*, 158:2078 – 2094, 2007.
- [9] M. Galende, G. I. Sainz, M. J. Fuente, and A. Herreros. Interpretability-accuracy improvement in a neuro-fuzzy ART based model of a DC motor. In *Proceedings of the 17th IFAC World Congress*, pages 7034 – 7039, Seoul, Korea, 6-11 July 2008.
- [10] A.E. Gaweda and J.M. Zurada. Data-driven linguistic modeling using relational fuzzy rules. *IEEE Transactions on Fuzzy Systems*, 11(1):121–134, 2003.
- [11] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press (Third edition), 1996.
- [12] S.H. Huang and H. Xing. Extract intelligible and concise fuzzy rules from neural networks. *Fuzzy Sets and Systems*, (132):233–243, 2002.
- [13] Y. Jin. Fuzzy modelling of high-dimensional systems: Complexity reduction and interpretability improvement. *IEEE Transactions on Fuzzy Systems*, 8(2):212–221, April 2000.
- [14] Y. Jin and B. Sendhoff. Extracting interpretable fuzzy rules from RBF networks. *Neural Processing Letters*, 17(2):149–164, 2003.
- [15] F. Karray and C. De Silva. *Soft Computing and Intelligent Systems Design, Theory, Tools and Applications*. Addison Wesley Publishing, 2004.
- [16] R. Mikut, J. Jäkel, and L. Gröll. Interpretability issues in data-based learning of fuzzy systems. *Fuzzy Sets and Systems*, 150:179 – 197, 2005.
- [17] G.I. Sainz Palmero, Y. Dimitriadis, J.M. Cano Izquierdo, E. Gómez Sánchez, and E. Parrado Hernández. ART based model set for pattern recognition: FasArt family. In H. Bunke and A. Kandel, editors, *Neuro-fuzzy pattern recognition*, pages 147–177. World Scientific Pub. Co., 2000.
- [18] M. Setnes and R. Babuska. Rule base reduction: Some comments on the use of orthogonal transforms. *IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and reviews*, 31(2):199 – 206, May 2001.
- [19] M. Setnes, R. Babuska, U. Kaymak, and H.R. Lemke. Similarity measures in fuzzy rule base simplification. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 28(3):376 – 386, June 1998.
- [20] M. Setnes and H. Roubos. Ga-fuzzy modeling and classification: Complexity and performance. *IEEE Transactions on Fuzzy Systems*, 6(8):509 – 522, May 2000.
- [21] J. Yen and L. Wang. Simplifying fuzzy rule-based models using orthogonal transformation methods. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 29(1):13 – 24, February 1999.

Resolución de Problemas Multi-clase con Sistemas de Clasificación Basados en Reglas Difusas Lingüísticos Basados en Aprendizaje por Parejas y Relaciones de Preferencia

Alberto Fernández¹ María Calderón² Edurne Barrenechea² Humberto Bustince² Francisco Herrera¹

¹ Dept. de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, España, {alberto,herrera}@decsai.ugr.es

² Dept. de Automática y Computación, Universidad Pública de Navarra, España, Maria.CalderonTurullols@cajanavarra.es, {edurne.barrenechea, bustince}@unavarra.es

Resumen

Esta contribución versa sobre el uso de Sistemas de Clasificación Basados en Reglas Difusas Lingüísticas en problemas con múltiples clases. La idea es decomponer el conjunto de datos original en problemas de clasificación binarios usando la metodología de aprendizaje por parejas (enfrentando todas las parejas de clases), y obtener un sistema difuso independiente para cada una de ellos. A lo largo del proceso de inferencia, cada subclasificador genera un grado de asociación para sus dos clases y estos valores se codifican en una relación de preferencia difusa.

Nuestro análisis se centra en el paso de predicción de la clase. Concretamente, proponemos el manejo de la relación de preferencia difusa usando un criterio de no dominancia sobre las diferentes alternativas, contrastando el comportamiento de este modelo con una estrategia de voto. El objetivo es mostrar la bondad del uso del aprendizaje por parejas y estudiar cómo afecta cada una de las alternativas de clasificación seleccionadas en el rendimiento final del modelo.

Palabras Clave: Sistemas de clasificación basados en reglas difusas, problemas multi-clase, relaciones de preferencia difusas, multi-clasificadores, aprendizaje por parejas.

1 Introducción

Los Sistemas de Clasificación Basados en Reglas Difusas (SCBRDs)[12] son una herramienta popular en Aprendizaje Automático debido a la ventaja del uso de términos lingüísticos para incrementar la interpretabilidad global del modelo de salida.

En este trabajo, centramos nuestra atención en el problema de múltiples clases, dado que este tipo de datos son muy comunes en aplicaciones del mundo real y pueden representar una dificultad añadida para los clasificadores difusos para encontrar una correcta separabilidad en un espacio de búsqueda con un alto número de clases. Para simplificar el problema original, proponemos el uso del enfoque de aprendizaje por parejas [8], que consiste en entrenar un clasificador para cada posible par de clases ignorando los ejemplos que no pertenecen a las clases relacionadas. Para determinar la salida entre todas las predicciones para los clasificadores asociados, los representaremos en una relación de preferencia difusa transformando el problema de clasificación en un problema de toma de decisión. En este trabajo, proponemos el uso de un criterio de no dominancia maximal [16] para el proceso de decisión final, dado que hemos mostrado previamente su buen comportamiento en sistemas de clasificación lingüísticos [4]. El marco conceptual de la clasificación basada en relaciones de preferencia difusas fue propuesto por Hüllermeier y Brinker en [11]. Adicionalmente, Hühn y Hüllermeier propusieron en [10] el algoritmo FR3, un SCBRD aproximativo que usa una regla de decisión basada en una estrategia de voto para tomar la decisión final en el proceso de clasificación.

En nuestro estudio, determinaremos la bondad del enfoque de aprendizaje por parejas para sistemas difusos lingüísticos analizando las diferencias en rendimiento obtenidas por el modelo clásico de SCBRD y el enfoque de multi-clasificación utilizando un método de aprendizaje de reglas difusas con un buen comportamiento, el algoritmo “Fuzzy Hybrid Genetics-Based Machine Learning” (FH-GBML) [14, 15]. Además, analizaremos el mencionado criterio de no dominancia que proponemos en contraste con la estrategia de voto introducida por Hühn y Hüllermeier en [10], para tener conclusiones sobre la utilidad del aprendizaje por parejas para SCBRD lingüísticos.

Hemos seleccionado 14 conjuntos de datos multiclase del repositorio UCI [2] dentro del marco experimental. La medida de rendimiento se basa en el porcentaje de acierto y la calidad de los resultados está apoyada por un correcto análisis estadístico tal y como se sugiere en la literatura [3, 7].

Para ello, este trabajo se organiza como sigue. En la Sección 2 presentamos el concepto de multi-clasificación, e introducimos nuestra propuesta de aprendizaje por parejas usando relaciones de preferencia, también describiendo el algoritmo FH-GBML seleccionado para nuestro estudio. La Sección 3 incluye el marco experimental, esto es, los conjuntos de datos para el testeo, los parámetros de configuración y los tests estadísticos para la comparativa del rendimiento. En la Sección 4 presentamos nuestro análisis empírico. La Sección 5 concluye el trabajo.

2 Sistemas de Multi-Clasificación Basados en Reglas Difusas

En este trabajo, buscamos mejorar el rendimiento de los SCBRDs introduciendo una propuesta de multi-clasificación basada en aprendizaje por parejas [5, 8]. Esta metodología permite una mejor exploración del dominio del problema dividiendo el conjunto de datos original en subproblemas binarios que son más fáciles de discriminar.

En esta sección introduciremos primero el concepto de multi-clasificación y el esquema de aprendizaje para un sistema difuso lingüístico basado en aprendizaje por parejas. A continuación, definiremos el procedimiento usado para obtener la clasificación final entre todos los clasificadores por medio de las relaciones de preferencia difusas. Finalmente, describiremos el algoritmo FH-GML, que será empleado como la base del modelo difuso.

2.1 Problemas Multi-Clase Via Aprendizaje por Parejas

Existe una gran cantidad de aplicaciones que requieren una categorización multi-clase. Para simplificar el proceso de clasificación, podemos transformar las fronteras del problema distinguiendo sólo entre dos clases dividiendo el problema inicial en múltiples conjuntos de dos clases que pueden ser resueltos por separado.

Concretamente, hemos considerado el enfoque de aprendizaje por parejas [8], que consiste en entrenar un clasificador para cada posible pareja de clases ignorando los ejemplos que no pertenecen a las clases asociadas. En el momento de la clasificación, se somete una instancia de consulta a todos los modelos binarios, y las predicciones de estos modelos se com-

binan en una clasificación conjunta.

Cada uno de estos clasificadores difusos tendrá su propia Base de Conocimiento (BC), compuesta por una Base de Datos (BD) y una Base de Reglas (BR). Hemos decidido centrarnos en la precisión del modelo contextualizando las particiones difusas para cada subproblema por separado, tomando como universo de discurso el rango de los valores de las variables del subconjunto de ejemplos para cada subconjunto de entrenamiento.

La BR para cada clasificador se aprende usando un método de aprendizaje de reglas, que puede seleccionarse entre las diferentes propuestas de la literatura especializada.

Una vez que todas las BCs han sido aprendidas, procedemos con el paso final de inferencia. Cuando un nuevo patrón de entrada se presenta al sistema, cada SCBRD es disparado para definir el grado de salida para su pareja de clases asociadas.

2.2 Proceso de Clasificación con Relaciones de Preferencia Difusas Basado en No-Dominancia

Consideraremos el problema de clasificación como un problema de toma de decisión, y definiremos una relación de preferencia difusa R^b [16] con las salidas correspondientes de los SCBRDs. De este modo, el cálculo de cada grado de preferencia se basa en la función de agregación que combina los grados de asociación positivos entre las reglas difusas y el patrón de entrada, lo que se conoce como Método de Razonamiento Difuso.

$$R^b = \begin{bmatrix} - & r_{1,2} & \dots & r_{1,m} \\ r_{2,1} & - & \dots & r_{2,m} \\ \vdots & \ddots & \ddots & \vdots \\ r_{m,1} & r_{m,1} & \dots & - \end{bmatrix} \quad (1)$$

Consideramos el emparejamiento máximo, donde cada nuevo patrón x_p se clasifica como la clase consecuente de una única regla ganadora ($Class(x_p) = C_w$) que se determina como

$$\mu_{A_w}(x_p) \cdot RW_w = \max\{\mu_{A_q}(x_p) \cdot RW_q, Regla_q \in BR\} \quad (2)$$

donde $\mu_{A_q}(x_p)$ es el grado de pertenencia del ejemplo patrón $x_p = (x_{p1}, \dots, x_{pn})$ con el antecedente de la regla R_q y RW_q es el peso de la regla.

Por lo tanto, $R^b(i, j)$ (el grado de preferencia difuso entre las clases i y j) es el máximo grado de asociación para todas las reglas en la BR que concluyen la clase i . $R^b(i, j)$ será normalizado a $[0, 1]$ mediante la expresión (3), obteniendo la relación $R(i, j) = 1 - R(j, i)$.

$$R(i, j) = \frac{R^b(i, j)}{R^b(i, j) + R^b(j, i)} \quad (3)$$

A partir de la relación de preferencia difusa podemos extraer un conjunto de alternativas (clases) no dominadas como solución del problema de toma de decisión difuso y de este modo, nuestra salida de clasificación. Concretamente, los elementos maximales no dominados de R se calculan por medio de las siguientes operaciones, de acuerdo al criterio de no dominancia propuesto por Orlovsky [16]:

- Primero, calculamos la relación estricta de preferencia difusa R' que es igual a :

$$R'(i, j) = \begin{cases} R(i, j) - R(j, i), & \text{cuando } R(i, j) > R(j, i), \\ 0, & \text{en otro caso.} \end{cases} \quad (4)$$

- Entonces, calculamos el grado de no dominancia de cada clase ND_i , que se obtiene simplemente como:

$$ND_i = 1 - \sup_{j \in C} [R'(j, i)] \quad (5)$$

Este valor representa el grado en el que la clase i no es dominada por ninguna de las clases restantes. C representa el conjunto total de clases en el conjunto de datos. La clase de salida se calcula como el índice del valor maximal de no dominancia:

$$Clase(x_p) = \arg \max_{i=1, \dots, m} \{ND_i\} \quad (6)$$

2.3 Algoritmo de Generación de Reglas “Fuzzy Hybrid Genetics Based Machine Learning”

El método FH-GBML [14] consiste en un enfoque Pittsburgh donde cada conjunto de reglas se maneja como un individuo. También contiene un enfoque de aprendizaje genético Cooperativo-Competitivo (un individuo representa una única regla), que se usa como un tipo de mutación heurística para modificar parcialmente cada conjunto de reglas.

Este método utilizar reglas difusas estándar con pesos en las reglas [13] donde cada variable de entrada x_i se representa mediante un término lingüístico o etiqueta. El sistema define 14 posibles términos lingüísticos para cada atributo así como un conjunto especial “do not care”.

En el proceso de aprendizaje, N_{pop} conjuntos de reglas se crean seleccionando aleatoriamente N_{rule} patrones de entrenamiento. Entonces, se genera una regla difusa para cada uno de los patrones de entrenamiento seleccionados eligiendo probabilísticamente un

conjunto difuso antecedente a partir de los 14 candidatos ($P(B_k) = \frac{\mu_{B_k}(x_{pi})}{\sum_{j=1}^{14} \mu_{B_j}(x_{pi})}$) y cada conjunto difuso antecedente de la regla difusa generada se reemplaza con *don't care* utilizando una probabilidad predeterminada $P_{don't\ care}$.

$N_{pop} - 1$ conjuntos de reglas son generados mediante selección, cruce y mutación en el mismo modo que el algoritmo estilo Pittsburgh. A continuación, con una probabilidad predeterminada, se aplica una única iteración del algoritmo genético Cooperativo-Competitivo para cada uno de los conjuntos de reglas generados.

Finalmente, el mejor conjunto de reglas se añade de la población actual a los nuevos ($N_{pop} - 1$) conjuntos de reglas generados para formar la siguiente población y, si el criterio de parada no se ha satisfecho aún, el proceso genético se repite de nuevo. La clasificación se realiza siguiendo el Método de Razonamiento Difuso del emparejamiento máximo (regla ganadora).

Para más detalles acerca de esta propuesta, por favor refiérase a [14]. Este algoritmo está disponible dentro de la herramienta software KEEL [1] (<http://www.keel.es>).

3 Marco Experimental

En esta sección, primero daremos detalles de los problemas multi-clase reales escogidos para la experimentación y los parámetros de configuración de los SCBRDs (subsecciones 3.1 y 3.2 respectivamente). Finalmente, introduciremos los tests estadísticos aplicados para comparar los resultados obtenidos a lo largo del estudio experimental (subsección 3.3).

3.1 Conjuntos de datos

La Tabla 1 resume las propiedades de los conjuntos de datos seleccionados. Muestra, para cada conjunto de datos, el número de ejemplos (#Ej.), el número de atributos (#Atrs.) y el número de clases (#Cl.). Los conjuntos de datos *letter*, *penbased* y *page-blocks* han sido sampleados mediante estratificación al 10% para reducir su tamaño para el entrenamiento. En el caso de los valores perdidos (*cleveland*) hemos eliminado esas instancias del conjunto de datos.

La estimación del ratio de acierto fue obtenida por medio de una validación cruzada de 5 partes. Esto es, dividimos el conjunto de datos en 5 partes, cada una conteniendo el 20 % de los patrones del conjunto de datos. Para cada parte, el algoritmo fue entrenado con los ejemplos de las particiones restantes y entonces, testeado con la partición actual.

Tabla 1: Descripción resumen de los conjuntos de datos

id	Conjunto	#Eh.	#Atrs.	#Cl.
bal	balance scale	625	4	3
cle	cleveland	297	13	5
eco	ecoli	336	7	8
gla	glass identification	214	9	7
iri	iris	150	4	3
let	letter	2000	16	26
new	new-thyroid	215	5	3
pag	page-blocks	548	10	5
pen	pen-based recognition	1,099	16	10
seg	segment	2310	19	7
shu	shuttle	2,175	9	5
veh	vehicle	846	18	4
win	wine	178	13	3
yea	yeast	1,484	8	10

3.2 Parámetros

La configuración seleccionada para el algoritmo FH-GBML consiste en la T-norma producto como operador de conjunción, junto con el Factor de Certeza Penalizado [13] para el peso de la regla y MRD de la regla ganadora. Respecto a los parámetros específicos para el proceso genético, hemos escogido los siguientes valores:

- Número de reglas difusas: $5 \cdot d$ reglas.
- Número de conjuntos de reglas: 200 conjuntos.
- Probabilidad de Cruce: 0,9.
- Probabilidad de Mutación: $1/d$.
- Número de reglas reemplazadas: Todas las reglas excepto la mejor (parte Pittsburgh), número de reglas / 5 (parte Cooperativa-Competitiva).
- Número total de generaciones: 1.000.
- Probabilidad de “Don’t care”: 0,5.
- Probabilidad de la aplicación de la iteración Cooperativa-Competitiva: 0,5.

donde d representa la dimensionalidad del problema (número de variables).

3.3 Tests estadísticos para la comparativa del rendimiento

En esta contribución, usamos las técnicas de comprobación de hipótesis para proporcionar un soporte estadístico al análisis de los resultados [6, 17]. Concretamente, usaremos tests no paramétrico, dado que las condiciones iniciales que garantizan la fiabilidad de los tests paramétricos puede no satisfacerse, haciendo que el análisis estadísticos pierda credibilidad con estos tipos de tests [3, 7]. Concretamente, usaremos el test de rangos de Wilcoxon [17] como procedimiento estadístico no paramétrico para realizar comparaciones

entre dos algoritmos y, para comparaciones múltiples, usamos el test de Iman-Davenport [17] para detectar diferencias estadísticas entre un grupo de resultados y el test post-hoc de Holm [9] para encontrar qué algoritmos rechazan la hipótesis de igualdad con respecto a un método de control seleccionado.

4 Análisis Experimental

Los resultados de los experimentos para el algoritmo FH-GBML se muestran en la Tabla 2 donde, por columnas, podemos observar los resultados de entrenamiento y test para el algoritmo FH-GBML básico y para FH-GBML con los enfoques de aprendizaje por parejas, tanto para el criterio de clasificación de no dominancia (notada con el sufijo ND), y para la regla de decisión basada en una estrategia de voto (notada con el sufijo EV).

Dividimos nuestro análisis experimental en dos partes:

- Primero, queremos determinar si la propuesta del multi-clasificador mejora el rendimiento del SC-BRD lingüístico que maneja todas las clases independientemente.
- A continuación, nuestro objetivo es analizar el comportamiento de nuestra propuesta para el proceso de decisión de la salida basada en un criterio de no dominancia contra una estrategia de voto.

4.1 Análisis de la utilidad de la propuesta de aprendizaje por parejas para sistemas de clasificación basados en reglas difusas lingüísticas

Para llevar a cabo el primero estudio, mostramos el ranking de los enfoques FH-GBML asignando las posiciones de los algoritmos en función del rendimiento para cada conjunto de datos y obteniendo a continuación el valor medio. La Figura 1 muestra el ranking medio calculado para las tres diferentes alternativas: el enfoque básico y las dos técnicas multi-clasificador. Podemos observar que tanto los métodos FH-GBML-ND como FH-GBML-EV son la mejor opción, puesto que el FH-GBML básico obtiene el peor ranking con un valor mucho mayor que los anteriores.

A continuación, realizamos un test de Iman-Davenport (Tabla 3) para detectar diferencias significativas entre los resultados de estas propuestas. El “p-value” asociado es cercano a cero, lo que implica que hay diferencias significativas entre los resultados y así, debemos aplicar un test post-hoc, en este caso el test de Holm. Queremos comparar el algoritmo FH-GBML básico contra ambos enfoques de aprendizaje por parejas.

Tabla 2: Resultados medios en acierto para el algoritmo FH-GBML con el enfoque básico y con la versión de multi-clasificación

Conjunto	#Cl.	FH-GBML		FH-GBML-ND		FH-GBML-EV	
		Entr.	Test	Entr.	Test	Entr.	Test
bal	3	85.64 $\pm$ 2.49	82.24 $\pm$ 2.85	87.04 $\pm$ 1.97	84.96 $\pm$ 2.55	86.84 $\pm$ 2.12	84.96 $\pm$ 2.55
iri	3	99.33 $\pm$ 0.70	93.33 $\pm$ 4.08	99.50 $\pm$ 0.46	94.67 $\pm$ 2.98	98.17 $\pm$ 1.09	94.00 $\pm$ 2.79
new	3	96.74 $\pm$ 0.66	91.16 $\pm$ 3.03	99.53 $\pm$ 0.49	95.81 $\pm$ 1.95	93.72 $\pm$ 1.26	94.42 $\pm$ 1.27
win	3	97.61 $\pm$ 0.63	92.70 $\pm$ 4.21	100.0 $\pm$ 0.00	96.08 $\pm$ 3.75	98.31 $\pm$ 1.46	94.40 $\pm$ 5.23
veh	4	62.44 $\pm$ 1.07	58.15 $\pm$ 3.47	74.53 $\pm$ 1.09	66.67 $\pm$ 4.46	73.82 $\pm$ 0.77	66.08 $\pm$ 4.57
cle	5	63.05 $\pm$ 0.73	50.84 $\pm$ 6.14	75.42 $\pm$ 0.76	56.56 $\pm$ 1.83	73.06 $\pm$ 0.69	57.92 $\pm$ 1.96
pag	5	95.67 $\pm$ 0.51	94.53 $\pm$ 0.91	98.17 $\pm$ 0.49	95.62 $\pm$ 1.20	96.49 $\pm$ 0.94	95.62 $\pm$ 1.77
shu	5	95.52 $\pm$ 0.67	95.22 $\pm$ 1.43	98.45 $\pm$ 0.60	97.70 $\pm$ 0.83	95.53 $\pm$ 1.46	95.40 $\pm$ 2.16
gla	6	70.44 $\pm$ 0.37	60.29 $\pm$ 6.30	82.48 $\pm$ 0.72	68.25 $\pm$ 7.34	73.83 $\pm$ 1.48	63.11 $\pm$ 6.28
seg	7	79.82 $\pm$ 1.19	78.70 $\pm$ 2.27	94.96 $\pm$ 0.35	93.29 $\pm$ 1.73	79.26 $\pm$ 3.45	78.23 $\pm$ 2.55
eco	8	79.76 $\pm$ 1.44	76.19 $\pm$ 5.04	92.34 $\pm$ 1.27	81.55 $\pm$ 4.63	84.67 $\pm$ 0.97	76.80 $\pm$ 6.48
pen	10	71.93 $\pm$ 1.19	69.82 $\pm$ 1.84	96.34 $\pm$ 0.67	91.09 $\pm$ 1.14	96.82 $\pm$ 0.35	92.09 $\pm$ 2.05
yea	10	53.84 $\pm$ 1.26	51.22 $\pm$ 4.54	65.40 $\pm$ 0.66	58.90 $\pm$ 1.44	61.12 $\pm$ 0.61	57.14 $\pm$ 1.94
let	26	18.19 $\pm$ 1.64	16.35 $\pm$ 1.58	86.30 $\pm$ 0.31	71.55 $\pm$ 2.18	46.81 $\pm$ 2.11	44.45 $\pm$ 6.27
Mean	X	76.43 $\pm$ 1.04	72.20 $\pm$ 3.41	89.32 $\pm$ 0.70	82.34 $\pm$ 2.72	82.75 $\pm$ 1.34	78.19 $\pm$ 3.42

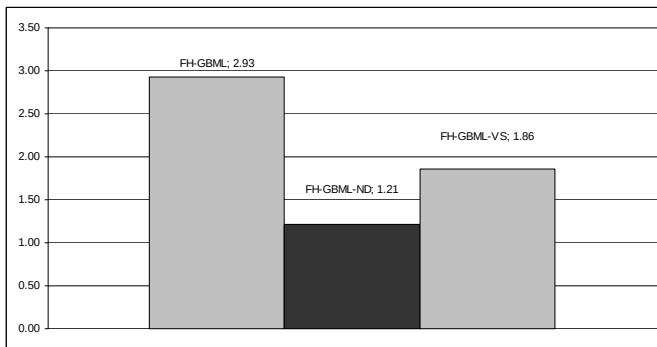


Figura 1: Ranking en media para las diferentes versiones FH-GBML: esquema básico, multi-clasificador con criterio de no dominancia y multi-clasificador con estrategia de voto

Tabla 3: Resultados del test de Iman-Davenport para la comparativa del algoritmo FH-GBML con los enfoques básico y multi-clasificador. El estadístico se calcula con 2 y 26 grados de libertad para $\alpha = 0.05$ y 14 conjuntos de datos

Iman-Davenport	Valor en F_F	p-value
39.0	3.3690	1.49012E-8

El resultado de este test se muestra en la Tabla 4, en la que podemos observar que el algoritmo FH-GBML es superado en rendimiento por la versión de multi-clasificador, tanto para FH-GBML-ND como para FH-GBML-EV. Podemos concluir por tanto que el uso de la propuesta de aprendizaje por parejas proporciona un beneficio en el rendimiento para SCBRDs lingüísticas.

Tabla 4: Test de Holm para los enfoques FH-GBML. FH-GBML es el método de control

i	algoritmo	z	p	α/i	Hipótesis ($\alpha = 0.05$)
2	FH-GBML-ND	4.536	5.745E-6	0.025	Rech. FH-GBML-ND
1	FH-GBML-EV	2.835	0.005	0.05	Rech. FH-GBML-EV

4.2 Análisis del proceso de decisión basado en no dominancia contra una estrategia de voto

El siguiente objetivo de este análisis empírico es estudiar las dos alternativas seleccionadas para el proceso de decisión entre todas las predicciones de los clasificadores difusos. Con esta meta, hemos llevado a cabo un test de Wilcoxon, mostrado en la Tabla 5, en el que comparamos el criterio de no dominancia (ND) contra la estrategia de voto (EV).

Tabla 5: Test de Wilcoxon para comparar la propuesta de no dominancia contra la regla de decisión que usa una estrategia de voto para el algoritmo FH-GBML. R^+ corresponde a la suma de los rangos para el criterio de no dominancia y R^- a la regla de decisión que usa una estrategia de voto

Comparativa	R^+	R^-	p-value	Hipótesis ($\alpha = 0.05$)
FH-GBML-ND vs. FH-GBML-EV	92.5	12.5	0.012	Rechazado para FH-GBML-ND

Debemos destacar que el criterio de no dominancia (con un rendimiento promedio de 82.34) sobrepasa estadísticamente a la alternativa de la regla de decisión basada en la estrategia de voto (con un rendimiento promedio de 78.19), lo que supone una clara ventaja del enfoque propuesto para el proceso de decisión en sistema multi-clasificadores lingüísticos.

Además, de acuerdo a los resultados ordenados por número de clases en la Tabla 2, podemos observar que ambas metodologías son muy similares en rendimiento para conjuntos de datos con un número bajo de clases (hasta 6), pero entonces la calidad de los resultados para la estrategia de voto decrece en contraste con el criterio de no dominancia.

5 Conclusiones

En esta contribución, hemos aplicado una metodología de aprendizaje de parejas para construir un sistema multi-clasificador lingüístico orientado a discriminar entre parejas de clases y para obtener una mejor frontera de decisión en problemas multi-clase.

Para agregar la salida de cada clasificador simple, hemos usado una relación de preferencia difusa, traduciendo el problema de clasificación en un problema de toma de decisión. Para obtener la clase final de salida, hemos propuesto el uso de una regla de decisión basada en el criterio de no dominancia maximal, y hemos contrastado el comportamiento de este modelo con una estrategia de voto.

El estudio experimental mostró dos conclusiones principales: primero, la aplicación de una propuesta de aprendizaje por parejas mejora claramente el rendimiento del método de SCBRD lingüísticas. Segundo, hemos determinado que la mejor alternativa para la clasificación final es el uso de nuestro criterio propuesto de no dominancia en este contexto, especialmente cuando el número de clases del problema es alto.

Agradecimientos

Este trabajo de investigación ha sido posible gracias a la subvención del Ministerio Español de Ciencia y Tecnología bajo los Proyectos TIN2008-06681-C06-01 y TIN2007-65981.

Referencias

- [1] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, and F. Herrera. KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing*, 13(3):307–318, 2009.
- [2] A. Asuncion and D. Newman. UCI machine learning repository, 2007.
- [3] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [4] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, and F. Herrera. Enhancing fuzzy rule based systems in multi-classification using pairwise coupling with preference relations. In *EUROFUSE09 Workshop on Preference Modelling and Decision Analysis*, pages 39–46, 2009.
- [5] J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.
- [6] S. García, A. Fernández, J. Luengo, and F. Herrera. A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Computing*, 13(10):959–977, 2009.
- [7] S. García and F. Herrera. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
- [8] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998.
- [9] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.
- [10] J. C. Hühn and E. Hüllermeier. FR3: A fuzzy rule learner for inducing reliable classifiers. *IEEE Transactions on Fuzzy Systems*, 17(1):138–149, 2009.
- [11] E. Hüllermeier and K. Brinker. Learning valued preference structures for solving classification problems. *Fuzzy Sets and Systems*, 159(18):2337–2352, 2008.
- [12] H. Ishibuchi, T. Nakashima, and M. Nii. *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*. Springer-Verlag, 2004.
- [13] H. Ishibuchi and T. Yamamoto. Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 13:428–435, 2005.
- [14] H. Ishibuchi, T. Yamamoto, and T. Nakashima. Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Transactions on System, Man and Cybernetics B*, 35(2):359–365, 2005.
- [15] J. Luengo and F. Herrera. Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method. *Fuzzy Sets and Systems*, in press, doi:10.1016/j.fss.2009.04.001, 2009.
- [16] S. Orlovsky. Decision-making with a fuzzy preference relation. *Fuzzy Sets and Systems*, 1:155–167, 1978.
- [17] D. Sheskin. *Handbook of parametric and non-parametric statistical procedures*. Chapman & Hall/CRC, second edition, 2006.

UNA MÉTODOLÓGIA PARA GENERAR BASES DE CONOCIMIENTO DIFUSAS BASADA EN AGRUPAMIENTO DIFUSO, SELECCIÓN Y AJUSTE EVOLUTIVOS

Ruth Toscano¹ Javier Aroba² Antonio Peregrín²

¹ Máster en Tecnologías Informáticas Avanzadas
Universidad de Huelva. ruth.toscano@diesia.uhu.es

² Dpto. Tecnologías de la Información.
Universidad de Huelva. {aroba, peregrin}@dti.uhu.es

Resumen

En este trabajo se presenta una metodología para obtener bases de conocimiento lingüísticas de tipo Mamdani para sistemas difusos lingüísticos descriptivos a partir de un conjunto de ejemplos, basado en la combinación de técnicas de agrupamiento difuso, y de selección y ajuste evolutivos. Los resultados obtenidos con la metodología propuesta mejoran tanto en precisión (error) como en complejidad (número de reglas) a algoritmos clásicos y referentes en este campo con los que se ha comparado en un breve primer estudio experimental.

Palabras Clave: Bases de reglas, sistemas difusos lingüísticos, algoritmos genéticos, agrupamiento.

1 INTRODUCCIÓN

En el diseño de Sistemas Basados en Reglas Difusas (SBRDs) de tipo Mamdani, el principal elemento propio del problema es la Base de Conocimiento que contiene fundamentalmente, tanto las reglas difusas que componen la Base de Reglas (BR), como la definición de las funciones de pertenencia de las variables implicadas. Por este motivo, tanto la obtención de la BR [17, 16, 7] como el ajuste de las funciones de pertenencia [1, 14, 6] son problemas que han sido tradicionalmente de gran interés y ampliamente tratados en la literatura [13]. Aprender estos elementos a partir de conjuntos de datos o ejemplos de forma automática permite crear modelos difusos de forma sencilla. Generalmente se pretende que las BRs sean compactas, lo cual favorece su interpretabilidad, y que sean precisas, lo que generalmente produce contrariamente a lo anterior, más cantidad de reglas. Especialmente cuando los conjuntos de datos son grandes, este problema se vuelve de especial interés, pues muchos

métodos con buen nivel de precisión se vuelven poco manejables, bien por la gran cantidad de reglas que utilizan, o bien porque su eficiencia se ve comprometida.

En este ámbito se sitúa éste trabajo, en el que se presenta una metodología basada en combinar agrupamiento difuso y algoritmos evolutivos para la obtención de la RB y ajuste de las funciones de pertenencia, que obtiene buenos resultados en cuanto a número de reglas y un buen nivel de precisión. El agrupamiento difuso, bien conocido también para aplicaciones modelado [3], permite generar reglas que describen individualmente el conjunto de datos, mientras que la selección y el ajuste evolutivos simultáneos, [1, 6], permiten tanto un buen nivel de cooperación entre las reglas como entre éstas y las funciones de pertenencia, que se traduce en buena precisión, y en un conjunto final de reglas compacto.

El trabajo se organiza de la siguiente forma: En la Sección 2 se describe la metodología presentada que denominamos CEST, mientras que en la Sección 3 se desarrolla un breve estudio experimental que muestra las buenas cualidades de la metodología propuesta comparada tanto con un método clásico como con otro de amplio reconocimiento por sus buenos resultados, en dos problemas reales diferentes.

2 METODOLOGÍA CEST

En este trabajo se propone una metodología para generar bases de conocimiento para sistemas difusos descriptivos a partir de ejemplos basada en combinar en el agrupamiento (o *Clustering*) de los ejemplos para generar un conjunto de reglas candidatas, que a continuación son seleccionadas mediante un mecanismo de selección de reglas y ajuste de funciones de pertenencia simultáneos evolutivos (*Simultaneous Evolutionary Selection and Tuning*) CEST. En esta sección se describe el fundamento de los elementos de ésta metodología, la cual podemos considerar que tiene dos etapas secuenciales: la primera, que genera el conjunto de reglas candidatas utilizando una técnica de agrupamiento de los datos, y la segunda que emplea un algoritmo evolutivo para la selección de las reglas candidatas y ajuste de las funciones de pertenencia,

ambas de forma simultánea. Cada etapa se describe en sendas subsecciones.

2.1. GENERACIÓN DE REGLAS CANDIDATAS MEDIANTE UN SISTEMA DE AGRUPAMIENTO DIFUSO

Los algoritmos clásicos de agrupamiento o *clustering* generan una partición de la población de forma que cada caso es asignado exactamente a un *cluster*. El algoritmo Isodata [2] fue modificado por [9] y generalizado en [4] y [5] hasta convertirlo en un algoritmo de agrupamiento difuso de propósito general, denominado algoritmo de la C-medias difusas, y conocido por sus siglas en inglés como FCM (*Fuzzy C-means Method*), basado en la minimización de distancias entre dos puntos y los centroides de los *clusters* (c-medias). Básicamente, este algoritmo intenta clasificar n elementos $x_k \in X$ con $1 \leq k \leq n$, con p características cada uno, es decir, $X \subset \mathbb{R}^p$, en c *clusters* difusos, asignando una función de pertenencia μ_{ik} :

$$\mu_{ik}: X \rightarrow [0,1], 1 \leq i \leq c, 1 \leq k \leq n \quad (1)$$

Para ello, el algoritmo trata de minimizar la siguiente función J_m :

$$J_m(U, P; X) = \sum_{k=1}^n \sum_{i=1}^c (\mu_{ik})^m D_{ik}^2 \quad (2)$$

Donde $U=(\mu_{ik})$ es la matriz de pertenencia de X , $P=[v_1, v_2, \dots, v_c]$ es un vector de centroides de *clusters* que debe ser determinado, y $m \in [1, \infty]$ es un exponente de peso sobre cada pertenencia difusa (en este trabajo se ha considerado $m=2$) y,

$$D_{ik}^2 = \|z_k - v_i\|_A = (z_k - v_i)^T (z_k - v_i) \quad (3)$$

es la norma usada para medir las distancias.

Finalmente, la función J es minimizada para obtener los componentes de U y P , es decir, la matriz de pertenencia y el vector de centroides de los *clusters*. Las condiciones necesarias para minimizar J son:

$$\mu_{ik} = \left[\sum_{j=1}^c \left[\frac{\|x_k - v_i\|_A}{\|x_k - v_j\|_A} \right]^{\frac{2}{m-1}} \right]^{-1} \quad \forall i, k \quad (4)$$

$$v_i = \frac{\sum_{k=1}^n (\mu_{ik})^m x_k}{\sum_{k=1}^n (\mu_{ik})^m} \quad \forall i \quad (5)$$

El objetivo final de esta fase de la metodología es obtener un sistema de reglas difusas a partir de una matriz de

datos cuantitativa multiparamétrica, para lo cual se ha aplicado un algoritmo de agrupamiento difuso basado en la metodología propuesta por Sugeno y Yasukawa en [15], con el propósito de construir un modelo difuso a base de reglas difusas IF-THEN de la forma:

$$R^i: \text{IF } x_i \in A^i \text{ THEN } y \in B^i \quad (6)$$

donde $X=[x_1, x_2, \dots, x_n] \in \mathbb{R}^n$ son parámetros de entrada antecedentes), $A=[A_1, A_2, \dots, A_n]$ son n conjuntos difusos, $y \in \mathbb{R}$ es un parámetro de salida (consecuente), y $B=[B_1, B_2, \dots, B_m]$ son m conjuntos difusos.

La metodología utilizada para obtener el modelo difuso, consiste básicamente en aplicar la partición difusa FCM al parámetro de salida y . Como resultado de este proceso, se obtiene el grado de pertenencia de cada uno de los datos de salida de la base de datos a cada uno de los conjuntos difusos B^i . Una vez obtenida una partición del espacio de salida en *clusters* difusos B^i , se realiza una proyección de estos *clusters* sobre el espacio de entrada obteniendo un conjunto difuso en $\mathbb{R}^n$, que proyectado sobre cada eje asigna a cada parámetro de entrada x_i un conjunto difuso A^i según la ecuación 6, como ilustra la Figura 1.

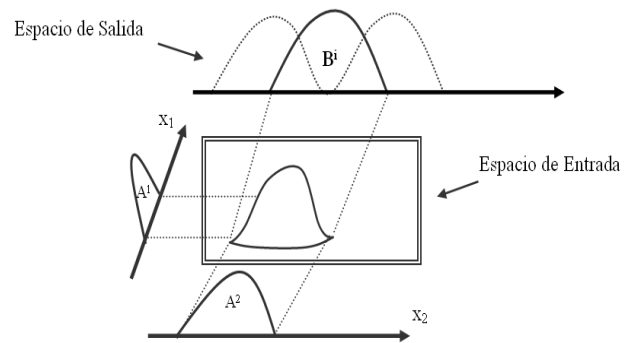


Figura 1: Proyección de *clusters* difusos

Con objeto de obtener un sistema de reglas difusas descriptivo sencillo, en este trabajo se ha optado por asignar etiquetas difusas de tipo triangular a cada uno de los parámetros que conforman tanto el antecedente como el consecuente de las reglas difusas obtenidas. Asimismo, aunque el proceso de clustering genera funciones de pertenencia específicas para cada regla, con objeto de obtener un sistema de tipo Mamdani en el que todas las reglas utilizan el mismo tipo de partición de las variables, se toma como parámetro de entrada el número de particiones difusas de cada variable (granularidad) y se reduce mediante el algoritmo de vecino más cercano el número de funciones de pertenencia que utiliza cada variable reasignándose las etiquetas inicialmente obtenidas. La Figura 2 muestra un ejemplo de una posible BR obtenida mediante este método, donde se puede observar que las etiquetas utilizadas en distintas reglas corresponden a una mismo particionamiento del universo de discurso de las variables. La anchura de las etiquetas es diferente entre cada una de ellas porque se mantiene el

obtenido en el proceso anterior, aunque lógicamente una misma etiqueta en distintas reglas también tiene como corresponde a un sistema Mamdani descriptivo, la misma anchura.

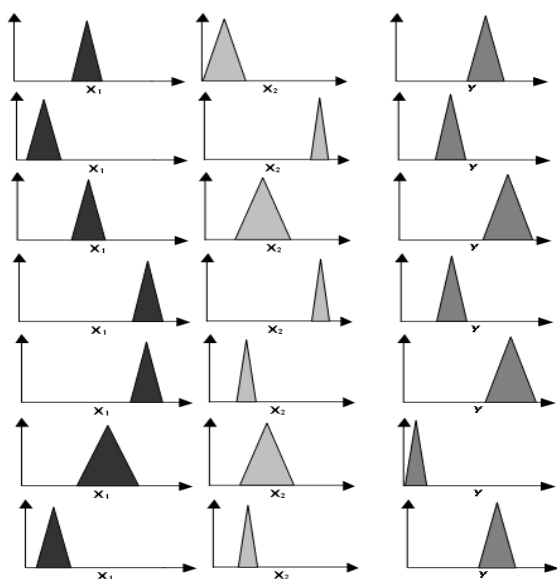


Figura 2: Ejemplo de algunas reglas candidatas

El número de reglas candidatas difusas que se generan por esta metodología se basa en el criterio propuesto por Fukuyama en [12], consistente básicamente en medir la varianza de los datos en un *cluster* y la varianza entre *clusters*. De esta forma, el número de *clusters* óptimo es aquel que minimiza la varianza en cada *cluster* y maximiza la varianza entre ellos. En este trabajo utilizamos un número de reglas igual a dos o cuatro veces el óptimo calculado con este procedimiento, dependiendo del problema, con objeto de partir para la selección evolutiva con un conjunto de reglas mayor y que el algoritmo pueda disponer de una zona de búsqueda más amplia.

2.2. SELECCIÓN Y AJUSTE EVOLUTIVO DEL CONJUNTO DE REGLAS CANDIDATAS

El conjunto de reglas y las funciones de pertenencia de cada variable involucradas que se han generado en el paso anterior, son utilizadas en este segundo paso para encontrar el conjunto final de reglas, y el ajuste las funciones de pertenencia de las mismas.

El método empleado en este trabajo para ello es similar al presentado en [1]. Se trata de utilizar un algoritmo evolutivo en cuyo cromosoma se codifican ambos elementos: las reglas candidatas y las funciones de pertenencia asociadas a cada variable.

Como se ha comentado, utiliza un esquema de codificación doble ($CS_S + CS_T$), donde:

- CS_S codifica las reglas, utilizando para ello una cadena binaria de N genes, cada uno representando una regla candidata proveniente del proceso de agrupamiento. Las reglas seleccionadas tendrán el valor “1” en su gen correspondiente de la cadena, mientras que “0” significa lo contrario. Esta forma de selección de un conjunto de reglas permite obtener subconjuntos de reglas con un mejor nivel de cooperación entre ellas [7], ya que en el conjunto inicial de reglas candidatas no está obtenido bajo esta premisa, sino bajo la de la calidad de cada regla individual aisladamente, lo cual podría conducir a que hubiese algunas reglas redundantes, contradictorias o en general con bajo nivel de cooperación.
- CS_T codifica las funciones de pertenencia para su ajuste, y lo hace concretamente utilizando para ello el llamado ajuste lateral [1], el cual sólo maneja el desplazamiento y la amplitud de las etiquetas. Este método para el ajuste de funciones de pertenencia permite encontrar con mayor facilidad el ajuste de las funciones de pertenencia, al utilizar un menor número de parámetros que el ajuste clásico [14], que si bien a priori supone un inferior número de grados de libertad, cuando se trata con complejas bases de conocimiento constituye una ventaja.

El modelo evolutivo utilizado en este trabajo deriva del modelo CHC [11], el cual presenta un buen equilibrio entre exploración y explotación, constituyendo una buena elección para problemas con espacios de búsqueda complejos. El modelo evolutivo CHC se caracteriza por, en cada generación, utilizar una población de padres de tamaño M para generar una población intermedia de M individuos, los cuales son aleatoriamente emparejados y utilizados para generar M' descendientes. Se mantiene entonces una competición por la supervivencia, donde los mejores M cromosomas de las poblaciones de padres y descendientes son seleccionados para constituir la nueva generación.

En la fase de recombinación no se aplica ningún proceso de mutación. En su lugar, cuando la población converge o el proceso de búsqueda deja de progresar adecuadamente (el umbral de cruce llega a 0 y no se generan nuevos descendientes), la población se reinicia. El cromosoma que representa la mejor solución hasta ese momento se utiliza como patrón para generar la nueva población en un cromosoma, mientras que el resto es aleatorio.

Aunque el algoritmo CHC fue concebido para cromosomas con codificación binaria, existen versiones para su uso con cromosomas con codificación real en parte es empleada en este trabajo, la cual utiliza el BLX- α como operador de cruce ($\alpha=0.5$). El cálculo de la distancia de Hamming en este caso se realiza convirtiendo los genes reales en una representación en cadenas y hallando la distancia de Hamming teniendo en cuenta si difiere o no cada uno de los caracteres. Sólo aquellas cadenas con una distancia mayor del umbral son combinados. El

umbral se inicializa a $L/4$ siendo L la longitud de la cadena o cromosoma. Cuando ningún descendiente es insertado en la nueva población, el umbral se reduce en una unidad.

El aprendizaje de ambos elementos simultáneamente, selección de reglas y funciones de pertenencia, permite obtener no sólo un conjunto de reglas con un buen nivel de cooperación sino también al mismo tiempo el ajuste de las funciones de pertenencia que le corresponde para obtener una buena precisión, es decir, no sólo la cooperación entre las reglas sino entre éstas y las descripciones de las funciones de pertenencia que utilizan.

3 EXPERIMENTACIÓN

Con objeto de mostrar los resultados obtenidos comparativamente del método propuesto, se ha planteado un breve estudio experimental en el que se utilizan dos conjuntos de datos y se utilizan dos métodos como referencia, WM [17] y COR [7]. Consideramos que dado que ambos citados métodos no cuentan con una etapa final de selección a ajuste evolutivos como la que incluye el método aquí propuesto, es necesario utilizar en la comparación este mecanismo también con los dos métodos, que mejora su precisión y complejidad, por el ajuste de las funciones de pertenencia y la selección del subconjunto de reglas con mejor nivel de cooperación, que realiza el evolutivo. Por tanto, los resultados que se van a mostrar en este estudio no son los de WM y COR sino el resultado de la mejora de ambos a los que se les añade el mecanismo descrito en la sección 2.2.

3.1. DESCRIPCIÓN DE LOS PROBLEMAS Y METODOLOGÍA DE COMPARACIÓN

Los problemas considerados para el estudio experimental son los siguientes:

- Un problema de distribución del sistema eléctrico (SE) [8], que consiste en estimar el coste de mantenimiento de las líneas de medio voltaje en una ciudad. (1059 datos, 4 variables de entrada continuas, y una variable de salida).
- El conjunto de datos del tiempo en Ankara (TA), [10] que trata sobre predecir la temperatura media en la ciudad turca de Ankara, (1609 datos, 9 variables de entrada continuas, y una de salida).

Utilizamos validación cruzada de orden 5, es decir, 5 particiones aleatorias de los datos, cada una con el 20% (4 con 211 ejemplos y una con 212 para el caso del problema SE, y 4 con 322 ejemplos y uno con 321 para el problema TA) y la combinación de 4 de ellos (el 80%) como entrenamiento con el quinto para prueba. Se han realizado un total de 30 experimentos para cada modelo ejecutando los métodos de aprendizaje para cada una de

las particiones 6 veces con diferentes semillas para el generador de números aleatorios. Mostraremos los valores promedio del clásico Error Cuadrático Medio (ECM) como media estándar de rendimiento cuya expresión es (7):

$$ECM(S)_B = \frac{\frac{1}{2} \sum_{k=1}^P (y_k - S(x_k))^2}{P}, \quad (7)$$

donde S es el modelo difuso, P es el número de pares del conjunto de datos $Z_k = (x_k, y_k)$, $k=1, \dots, P$, con x_k los valores de las variables de entrada, y con y_k siendo sus valores asociados correspondientes para la variable de salida.

Para cada partición por tanto, se han generado las bases de reglas correspondientes. En los casos de los métodos de WM [17] y de COR [7], éstos métodos generan aproximadamente 65 reglas en el caso del problema SE y 156 en TA. En el caso del método CEST es suficiente generar para el problema SE dos veces el óptimo según el criterio descrito en [8] como indicamos en la Sección 2, es decir, una media de 20 reglas, y para el problema TA cuatro veces dicho óptimo, es decir, una media de 66 reglas.

El tamaño de la población es de 50 individuos aleatoriamente inicializados (dentro de sus intervalos de variación) salvo uno, con la siguiente configuración:

- La parte de la selección de reglas, CS_S , con todos los genes a "1" de modo que todas las reglas provenientes del agrupamiento están activas inicialmente.
- La parte correspondiente al ajuste, CS_T , con la posición inicial de las etiquetas resultante de la etapa anterior.

Para ambos problemas, el número de evaluaciones del algoritmo evolutivo utilizado es de 200.000, determinado empíricamente mediante varias pruebas previas.

La función objetivo es la anteriormente descrita con la expresión (7).

3.2 RESULTADOS Y ANÁLISIS

En esta sección se muestran los resultados obtenidos y el análisis de los mismos. La Tabla 1 muestra los que corresponden al problema SE y la Tabla 2 al problema TA. Las columnas de éstas tablas muestran en primer lugar el método de obtención de la base de conocimiento, la media del número de reglas obtenido por los modelos, y el ECM en entrenamiento y en prueba. Asimismo se ha incluido el resultado de aplicar el test de la t-student (columna t-test) con 95% de confianza al mejor resultado medio de la correspondiente columna comparado 1 a 1 con el resto de resultados medios. La interpretación de esta columna es:

* Indica el resultado con la mejor media.

+ Indica un comportamiento significativamente peor que el mejor.

Como se puede apreciar en ambas tablas, la mejora del método propuesto es claramente visible tanto por la mejora de precisión, reduciendo el ECM en entrenamiento y prueba, así como en complejidad, con menor número de reglas. En el caso del problema SE (Tabla 1), la reducción de reglas es muy importante, habiendo mejorado también el ECM. En el problema TA (Tabla 2), la reducción en el número medio de reglas es importante pero en este caso destaca la reducción del ECM.

Tabla 1: Resultados obtenidos con el problema del SE

Método	#R	ECM _{Entr.}	t-test	ECM _{Prueba}	t-test
WM+EST	35.5	19084.121	+	23256.302	+
COR+EST	41.8	15793.401	+	19781.520	+
CEST	10.8	12274.421	*	13749.437	*

Tabla 2: Resultados obtenidos con el problema del TA

Método	#R	ECM _{Entr.}	t-test	ECM _{Prueba}	t-test
WM+EST	63.8	5.5969318	+	5.8172814	+
COR+EST	57.3	1.8875130	+	2.9646900	+
CEST	41.3	0.0066863	*	0.0072634	*

Debemos incidir que la comparación no se está realizando con el método de WM y el algoritmo COR originales sino tras aplicarles la misma selección evolutiva y ajuste de funciones de pertenencia simultáneos que tiene ya incluido el método propuesto, CEST, en su última etapa. Interpretamos que la ventaja del método propuesto radica en su primera etapa de generación de reglas candidatas basado en agrupamiento difuso, donde se obtiene una buena descripción del conjunto de datos representado por un grupo de posibles reglas individuales de gran calidad, no muy numerosas, y poco compatibles entre sí, que posteriormente formarán un buen subconjunto de reglas con un alto nivel de cooperación gracias a la selección evolutiva, y un bajo nivel de error gracias al ajuste de las funciones de pertenencia, todo ello con una buena integración al realizarse el proceso evolutivo conjuntamente.

4 CONCLUSIONES

En este trabajo se presenta una metodología para obtener bases de conocimiento para sistemas difusos descriptivos lingüísticos a partir de conjuntos de ejemplos cuyas reglas tienen un buen nivel de cooperación, y las funciones de pertenencia se encuentran ya ajustadas y adaptadas a la base de reglas. Empíricamente, en el breve estudio experimental desarrollado, la calidad de los resultados es bastante prometedora, mejorando la precisión y

complejidad de otros métodos ayudados por una de las fases de la propia metodología empleada como postprocesamiento, que es la selección de reglas y ajuste de las funciones de pertenencia. Las mejoras en precisión y complejidad son cuestiones interesantes especialmente para su uso en problemas con conjuntos de datos que mediante otras técnicas obtienen grandes cantidades de reglas o son poco eficientes en tiempo o recursos, y que es en la línea en la que se va a continuar el estudio de ésta propuesta: mayor número de conjuntos de datos con diferentes perfiles (en tamaño y complejidad), razón entre alguna medida del conjunto de datos y el número de reglas a generar en el agrupamiento difuso (para que dicho proceso no sea empírico), estudio de la eficiencia respecto de los otros métodos y estudios experimentales y estadísticos de los resultados con mayor profundidad.

Agradecimientos

Este trabajo está soportado por los proyectos TIN2008-06681-C06-06 del Ministerio de Ciencia e Innovación, y por el proyecto P07-TIC-03179 de la Consejería de Innovación Ciencia y Empresa de la Junta de Andalucía.

Referencias

- [1] R. Alcalá, J. Alcalá-Fdez, F. Herrera. A Proposal for the Genetic Lateral Tuning of Linguistic Fuzzy Systems and its Interaction with Rule Selection. *IEEE Transactions on Fuzzy Systems* 15:4, Pág. 616-635, 2007.
- [2] G.H. Ball, D.J. Hall. ISODATA, a novel method of data analysis and pattern classification. Technical report, Stanford Research Institute, 1965.
- [3] M. Setnes, R. Babuška, H. B. Verbruggen. Rule-Based Modeling: Precision and Transparency. *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews* 28:1, Pág. 165-169, 1998.
- [4] J.C. Bezdek. Pattern Recognition with fuzzy objective function algorithm. *New York: Plenum Press*, 1981.
- [5] J.C. Bezdek, R. Ehrlich, W. Full. FCM: The Fuzzy c-Means Clustering Algorithm. *Computers and Geosciences* 10:2-3, Pág. 191-203, 1984.
- [6] J. Casillas, O. Cordon, M.J. Del Jesus, F. Herrera. Genetic tuning of fuzzy rule deep structures preserving interpretability for linguistic modeling. *IEEE Trans. on Fuzzy Systems* 13:1, Pág. 13-29, 2005.
- [7] J. Casillas, O. Cordon, F. Herrera. COR: A methodology to improve ad hoc data-driven

- linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man and Cybernetics —Part B: Cybernetics*, 32:4, Pág. 526–537, 2002.
- [8] O. Cordon, F. Herrera, L. Sánchez. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence* 10, Pág.5-24, 1999
 - [9] J.C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics* 3, Pág. 32-57, 1973.
 - [10] M. Erdem Kurul, E. Tuzun. Available: www.wunderground.com; <http://www.wunderground.com/global/stations/17128>.
 - [11] L.J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of genetic Algorithms* 1, Pág. 265-283, 1991.
 - [12] Y. Fukuyama, M. Sugeno. A new method of choosing the number of clusters for fuzzy c-means method. *Proceedings of the 5th Fuzzy Systems Symposium*, Pág. 247-250, 1989.
 - [13] F. Herrera. Genetic Fuzzy Systems: Taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1, Pág. 27-46, 2008.
 - [14] C. Karr. Genetic algorithms for fuzzy controllers. *AI Expert* 6:2, Pág. 26-33, 1991.
 - [15] M. Sugeno, A. Yasukawa. A Fuzzy-Logic Based approach to qualitative Modeling. *IEEE Transactions on Fuzzy Systems* 1, Pág. 7-31, 1993.
 - [16] P. Thrift. Fuzzy logic synthesis with genetic algorithms. *Proceedings of the 4th International Conference on Genetic Algorithms* Morgan Kaufmann, Pág. 509-513, 1991.
 - [17] L.X. Wang, J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man, and Cybernetics* 22:6, Pág.1414-1427, 1992.

EVFUZZYSYSTEM: EVOLUCIÓN DE SISTEMAS DIFUSOS PARA PROBLEMAS DE REGRESIÓN MULTI-DIMENSIONALES

M. Martínez-Ballesteros¹ Víctor M. Rivas²

¹ Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, mariamartinez@us.es

² Departamento de Informática, Universidad de Jaén, vrivas@ujaen.es

Resumen

Este trabajo presenta EvFuzzySystem, un método evolutivo que permite el diseño completo de sistemas de lógica difusa, generando de forma simultánea funciones miembro y conjunto de reglas apropiados. EvFuzzySystem representa la extensión del método diseñado inicialmente para la resolución de problemas definidos por dos entradas y una salida. Esta extensión no ha sido trivial desde el punto de vista computacional. Los resultados muestran que puede ser aplicado a problemas de regresión compuestos de cualquier número de entradas y que los resultados obtenidos son comparables a los de métodos ya existentes.

Palabras Clave: Sistemas difusos, Algoritmos genéticos, Algoritmos evolutivos, Métodos híbridos, Función de aproximación.

1 INTRODUCCIÓN

En 2003, Rivas et al. [1] presentó EvFuzzy, un algoritmo evolutivo (AE) diseñado para evolucionar sistemas de lógica difusa (SLD) para problemas de dos dimensiones. El algoritmo evolutivo diseñaba simultáneamente tanto las funciones miembro como el conjunto de reglas necesarias para construir cada SLD. La extensión a problemas multidimensionales se mencionaba ya como una de las propuestas de futuro, siendo no obstante una nueva característica del algoritmo no trivial al representar cambios profundos tanto en la representación usada para los individuos como en los mecanismos para manejar el modelo, principalmente, los operadores genéticos diseñados específicamente para el algoritmo.

En este trabajo se presenta el nuevo algoritmo EvFuzzySystem, el cual puede ser usado para diseñar SLD para problemas de regresión de un número indeterminado de entradas. Desde el punto de vista computacional la extensión realizada no es trivial, por lo que se han debido de usar nuevas estructuras de datos más adecuadas para representar los individuos y, lo que es más importante, se han desarrollado nuevos operadores genéticos para evolucionar dichas estructuras.

2 EL ALGORITMO EVOLUTIVO

Las siguientes subsecciones detallan las arquitecturas tanto del SLD como del AE que lo optimiza, así como la función fitness y los operadores genéticos usados.

2.1 CODIFICACIÓN DE LOS INDIVIDUOS: EL SISTEMA DE LÓGICA DIFUSA

El esquema de representación elegido sigue el enfoque Pittsburg, siendo cada cromosoma un conjunto completo de base de reglas difusas. El cromosoma codifica tanto los antecedentes como los consecuentes, siendo las funciones de pertenencia de los términos lingüísticos funciones triangulares, como muestra la figura 1.

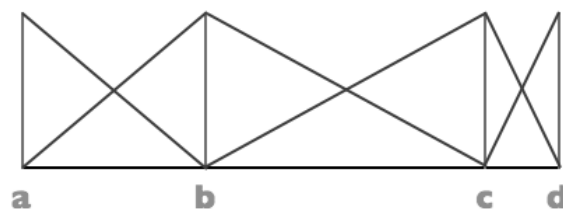


Figura 1: Conjuntos difusos de una variable del algoritmo implementado.

Cada cromosoma se implementa como dos estructuras; la primera representa el conjunto de n variables de entrada, cada una con número m_i de términos (pudiendo cada variable tener un número distinto de términos); la segunda contiene el conjunto de consecuentes, siendo por tanto su tamaño igual al producto de los términos de cada una de las variables de entrada. Cada gen almacena directamente un valor real, no una codificación binaria del mismo.

Para cada variable de entrada se almacenan los centros de las particiones triangulares de las funciones de pertenencia. El extremo de una función de pertenencia, se hace coincidir con el centro de la siguiente. El primer y último centro corresponden al valor mínimo y máximo establecidos respectivamente para dicha variable. Se han fijado en 3 el número mínimo de términos que debe tener cada variable, mientras que el número máximo es uno de los parámetros de entrada del algoritmo evolutivo.

Por su parte, tal y como se hizo en [1], los valores del conjunto de consecuentes son generados aleatoriamente entre un valor mínimo y un máximo dados sin conservar ningún tipo de orden entre los mismos. Para obtener el valor de los consecuentes, no es necesario un método de defuzzificación ya que cada regla tiene un valor real exacto para los consecuentes (valor denominado CRISP o no-difuso).

Para obtener la posición del consecuente correspondiente a una combinación de n valores de entrada, se utiliza la siguiente ecuación:

$$F(v_1, \dots, v_n) = \sum_{i=1}^n (v_i \prod_{j=i+1}^n n_j) \quad (1)$$

El SLD así creado, genera reglas de la siguiente forma:

SI X_1 es $MV[1, i_1]$ y X_2 es $MV[2, i_2]$ y ... y X_n es $MV[n, i_n]$ ENTONCES Z es $MC[F(i_1, i_2, \dots, i_n)]$

Siendo i_n la posición del término correspondiente a la variable X_n .

2.2 PROCESO EVOLUTIVO

El algoritmo evolutivo usado utiliza selección elitista, mantiene el tamaño de población constante y aplica una serie de operadores específicamente diseñados para trabajar con estos cromosomas, y que se explican en los siguientes apartados. He aquí los principales pasos del algoritmo:

1. Crear y evaluar la primera población; constará de p individuos generados aleatoriamente de tamaño aleatorio.

2. Seleccionar una subpoblación con los q mejores individuos. A ellos se aplicarán los operadores evolutivos
3. Repetir $p - q$ veces: seleccionar un individuo de la subpoblación mediante el método de la ruleta; duplicarlo; aplicarle uno de los operadores; evaluar el nuevo individuo.
4. Generar la nueva población uniendo a la subpoblación de q individuos los $p - q$ nuevos creados.
5. Repetir desde el paso 2 hasta alcanzar el número máximo de generaciones deseado.
6. Finalmente, el mejor SLD encontrado es el mejor individuo de la última generación.

2.3 GENERACIÓN DE LA POBLACIÓN INICIAL

La población inicial se genera de forma aleatoria tanto el conjunto de variables de entrada como el conjunto de valores de la variable de salida, así como el número de términos para cada variable de entrada según un máximo dado.

2.4 OPERADORES GENÉTICOS

Se han diseñado operadores genéticos específicos para trabajar con las estructuras utilizadas. Los operadores permiten tanto el intercambio de información entre los individuos como la modificación de los mismos. Además, cada operador genera un SLD válido respetando los valores mínimo y máximo de cada una de las variables de entrada, el número mínimo y máximo de términos lingísticos y el orden en el que están definidos los centroides de las funciones de pertenencia.

2.4.1 Recombinación de consecuentes

Este operador intercambia valores entre un SLD y otro en un solo sentido. Teniendo en cuenta que estos dos SLD no necesariamente tienen la mismas dimensiones, la recombinación no puede ser realizada de manera trivial.

El operador trabaja de la siguiente manera:

Aleatoriamente se eligen un SLD para ser modificado (M_r , o SLD receptor), y otro que proporcionará los genes para ser recombinados (M_d , o SLD donante).

De la estructura de consecuentes correspondiente a M_r , se seleccionan aleatoriamente un bloque aleatorio de celdas adyacentes. Para especificar este bloque basta seleccionar dos celdas que consideraremos los extremos o esquinas del bloque. Sean dichas celdas

$M_r[r_1]$ y $M_r[r_2]$. Imaginando espacialmente la estructura multidimensional que constituyen los consecuentes, las celdas a modificar con valores procedentes de M_d , serán las incluidas en el bloque espacial que forman las dos celdas seleccionadas aleatoriamente, como extremos de bloque, es decir, aquellas celdas del consecuente cuya posición al ser decodificada (aplicándole una función de decodificación para conocer la combinación de posiciones de términos pertenecientes a las variables) contiene las posiciones de los términos de cada variable dentro del rango establecido por las posiciones elegidas aleatoriamente como esquinas del bloque.

Para cada celda $M_r[r_i]$, así decodificada, para saber la posición del término correspondiente en cada variable a la que se refiere la celda en cuestión, $M_r[r_{1i}, r_{2j}, \dots, r_{nm}]$, (donde r_{1i} representa los valores r_{11} a r_{12} , r_{2j} los valores r_{21} a r_{22} y r_{nm} los valores r_{n1} a r_{n2}) el operador hace lo siguiente:

Del conjunto de variables de entrada perteneciente al SLD donante, M_d , de cada variable se selecciona el término cuyo valor se acerca más al término de la misma variable correspondiente al SLD receptor M_r .

Finalmente, se modifica el valor almacenado en $M_r[r_i]$, por el almacenado en la celda perteneciente a M_d que resulta tras codificar las posiciones de los términos de cada variable obtenidos en el punto anterior.

2.4.2 Adición de funciones de pertenencia (Mutador de incremento de tamaño)

Este operador modifica la estructura de un SLD añadiendo una función de pertenencia a las variables de entrada. Una vez seleccionado el individuo a modificar, se selecciona aleatoriamente una de las variables de entrada. A continuación, se elige aleatoriamente la posición del nuevo término dentro de la variable seleccionada y se genera un valor aleatorio para el nuevo término entre los valores de los términos anterior y posterior. Por último, dentro del conjunto de consecuentes se insertarán nuevos valores aleatorios correspondientes a la combinación del nuevo término con todos los términos del resto de variables de entrada.

2.4.3 Eliminación de funciones de pertenencia (Mutador de decremento de tamaño)

Este operador modifica la estructura de un SLD decrementando el número de funciones de pertenencia en las variables de entrada. El operador, una vez seleccionado el SLD a modificar, destruye uno de los términos de una de las variables de entrada, así como todos los consecuentes asociados a dicho término.

2.4.4 Modificación de un valor centroeide (Mutador de precedentes)

Este operador modifica el valor almacenado en un término de una de las variables del conjunto de variables de entrada elegidas al azar. Si el término elegido para cambiar es el i -ésimo, su valor se generará aleatoriamente entre los valores $i - 1$ e $i + 1$.

2.4.5 Modificación de un valor consecuente (Mutador de consecuentes)

Este operador modifica el valor almacenado en un consecuente del conjunto de consecuentes de un determinado SLD. El consecuente es seleccionado al azar y su valor varía dentro del rango delimitado para los consecuentes.

2.5 Función fitness

El fitness asignado al SLD es el valor correspondiente a la inversa del Error Cuadrático Medio Normalizado de las salidas correctas conocidas a las salidas producidas por el SLD, calculadas usando el conjunto de entrenamiento.

$$F_j = \frac{N}{\sum_{i=1}^N ((z_i - z_{ij}) / (z_{max} - z_{min}))^2} \quad (2)$$

Dónde F_j es el fitness para el individuo número j , N es el número de ejemplos del conjunto de entrenamiento (training set), z_i es la salida correcta, y z_{ij} es la salida proporcionada por el SLD. Un caso especial es cuando $z_i = z_{ij}$, $\forall i$ $1 \leq i \leq m$, porque F_j sería igual a $N/0$; en este caso, F_j es asignado con un valor muy alto. Como puede observarse, el tamaño del SLD no interviene en la computación o cálculo del fitness.

Este algoritmo se ejecuta con los siguientes parámetros:

- Tamaño de la población.
- Proporción de individuos que son eliminados en cada nueva generación (non-elite-rate).
- Probabilidad de aplicación de cada operador.
- Máximo número de términos para la primera generación.
- Máximo número de generaciones.

3 EXPERIMENTOS Y RESULTADOS

Los experimentos llevados a cabo pretenden comprobar el funcionamiento del algoritmo desarrollado y

validar el comportamiento de los operadores creados utilizando la misma metodología del algoritmo original publicado en [1], y además se comparan los resultados entre ambos. Las funciones que se ha pretendido aproximar son:

$$F(x, y) = \sin(xy) \quad (3)$$

$$F(x, y) = e^{x \sin(xy)} \quad (4)$$

$$F(x, y) = \frac{40e^{8(x-0.5)^2 + (y-0.5)^2}}{e^{8((x-2)^2 + (y-0.7)^2)} + e^{((x-0.7)^2 + (y-0.2)^2)}} \quad (5)$$

$$F(x, y) = \sin(2\pi\sqrt{x^2 + y^2}) \quad (6)$$

Para cada función se ha generado un fichero con 400 ejemplos, donde los datos de entrada X e Y tienen valores igualmente espaciados. Los valores que toma la salida Z , son los resultantes tras aplicar la fórmula en cada caso.

Para crear los conjuntos de entrenamiento y test de los experimentos, se han generado 5 particiones distintas para cada función, sin usar ningún tipo de particionador específico. Para cada partición, fueron elegidos aleatoriamente 320 puntos para ser usados como conjunto de entrenamiento (80% del total de puntos), mientras que los 80 restantes fueron como conjunto de test (20% del total de puntos) para examinar las aptitudes del mejor SLD obtenido.

Los valores utilizados en los parámetros durante las distintas ejecuciones del algoritmo son: 500 para el tamaño de la población, 300 para el número de generaciones, 40 máximo número de términos, 0.1 probabilidad de recombinación, 0.001 probabilidad del mutador de dimensión, 0.01 probabilidad del mutador de precedentes, 0.01 probabilidad del mutador de consecuentes y 0.35 como porcentaje de individuos que no van a pertenecer a la población élite.

A continuación se muestran gráficamente las salidas de cada función original para los conjuntos de entrenamiento y test, así como la mejor solución obtenida durante las cinco ejecuciones del algoritmo en cada problema.

Una vez representadas las funciones que debe ajustar el algoritmo evolutivo, así como la mejor salida producida durante las cinco ejecuciones y la evolución del fitness y el tamaño, a continuación, en la tabla

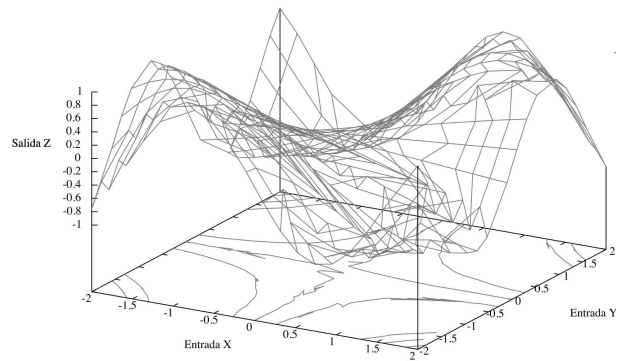


Figura 2: Gráfica Función 1 con salidas reales

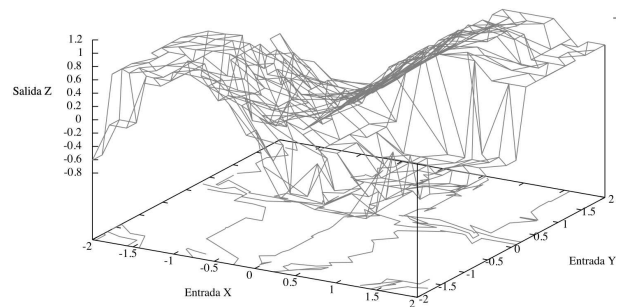


Figura 3: Gráfica Función 1 con salidas obtenidas por EvFuzzySystem

1, se muestra la media y la desviación típica para el fitness final, para el error de generalización y para el tamaño del mejor SLD encontrado para cada ejecución. El error de generalización se calcula como el error cuadrático medio normalizado del mejor individuo obtenido durante la ejecución, en base al conjunto de test. En cuanto al tamaño, o complejidad de las soluciones encontradas, suma de todos los términos de todas las variables más el número de consecuentes

Tabla 1: Resultados del algoritmo implementado para cada función considerada

ID	FITNESS	ECMN	TAMAÑO
1	39.18 (±6.17)	0.029 (±0.0049)	95.4 (±82.97)
2	124.72 (±37.72)	0.013 (±0.0048)	99.4 (±39.17)
3	209.3 (±165.23)	0.008 (±0.0034)	99.8 (±71.32)
4	32.75 (±7.27)	0.056 (±0.0152)	159.2 3 (±95.64)

Por su parte, la tabla 2 presenta los mismos resultados obtenidos en el trabajo original.

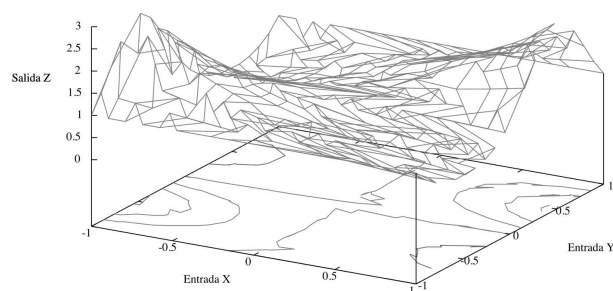


Figura 4: Gráfica Función 2 con salidas reales

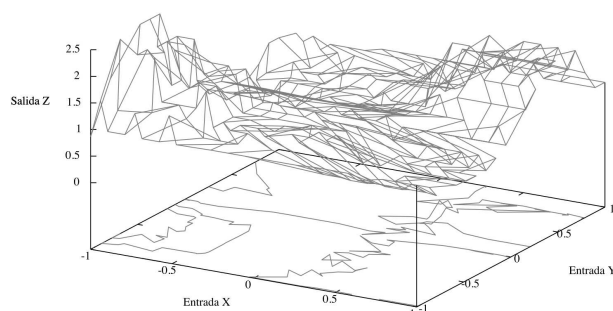


Figura 5: Gráfica Función 2 con salidas obtenidas por EvFuzzySystem

Para cada función de las cuatro con las que hemos trabajado, se ha aplicado el test de t de Student tanto para el fitness obtenido en el algoritmo implementado como en el algoritmo del artículo, así como para el error de generalización y el tamaño de los individuos.

Los resultados han mostrado que, aún cuando los resultados referentes al fitness puedan variar sustancialmente, no hay diferencias significativas en cuanto al error de generalización para las funciones 1 y 4. Por otro lado, el algoritmo original aproxima mejor la función 2, mientras que el nuevo algoritmo descrito ajusta mejor la función 3. En cuanto al tamaño, no existen diferencias significativas para ninguna de las funciones. Como ya se ha mencionado, el tamaño no influye en el cálculo del fitness además de no existir restricción o penalización alguna en cuanto al tamaño para generar los individuos.

Además de las comparaciones entre el algoritmo implementado en este trabajo y el propuesto en [1] para cada función presentada, es necesario comparar con otras técnicas de regresión así como la funcionalidad del mismo para otros conjuntos de datos donde intervienen un número superior a dos variables de entrada, lo cual fue la motivación para la implementación del algoritmo. Para tal fin, se ha utilizado la herramienta de Minería de Datos KEEL [9] que incluye diversos al-

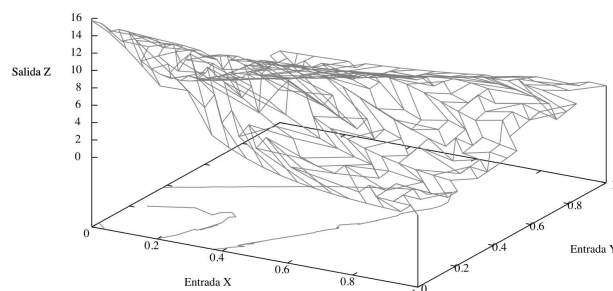


Figura 6: Gráfica Función 3 con salidas reales

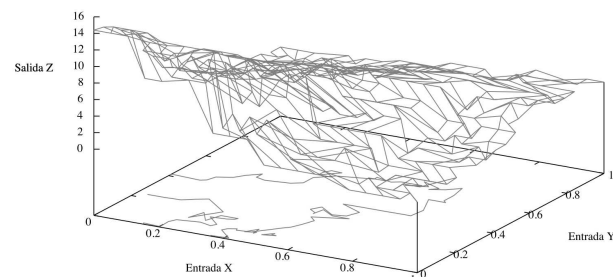


Figura 7: Gráfica Función 3 con salidas obtenidas por EvFuzzySystem

goritmos de regresión así como distintas bases de datos con más de dos variables de entrada.

Los algoritmos seleccionados pertenecen a diferentes familias de las técnicas disponibles de regresión dentro de la herramienta KEEL. Estos son: Fuzzy-Thrift (Aprendizaje de reglas difusas), Regr-NU-SVR (Máquinas de vectores de soporte) y Regr-GAP (Regresión simbólica).

La base de datos utilizada pertenece al problema MachineCPU, donde intervienen 6 variables de entrada y una de salida. La configuración de los parámetros de EvFuzzySystem para el problema MachineCPU es igual que en el experimento anterior, salvo el número de términos máximo que es 8. Los resultados para la media del Fitness y del Error de Generalización se muestran en la tabla 3.

Los resultados muestran que el algoritmo propuesto funciona para más de dos variables de entrada, además de obtener los mejores resultados de los 4 algoritmos ejecutados con la base de datos MachineCPU.

4 CONCLUSIONES

Los experimentos incluidos, han sido realizados para validar el comportamiento de los operadores creados. Los relativos a las funciones del trabajo origi-

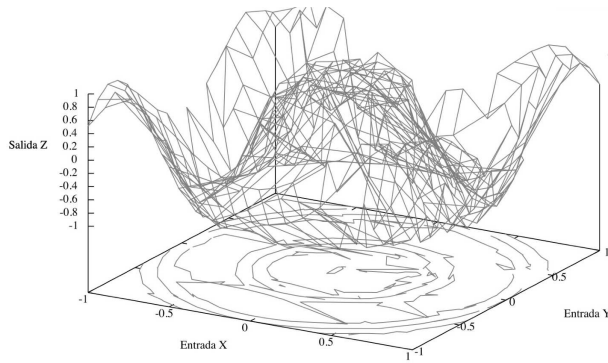


Figura 8: Gráfica Función 4 con salidas reales

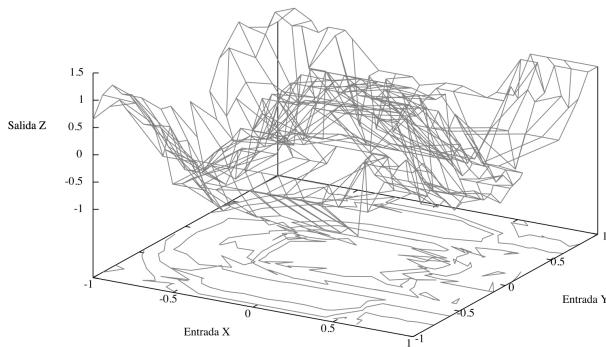


Figura 9: Gráfica Función 4 con salidas obtenidas por EvFuzzySystem

inal [1] demuestran que siguiendo las mismas características del algoritmo propuesto en el artículo, se obtienen resultados con diferencias no significativas en cuanto al tamaño de las soluciones, y de diverso sentido en cuanto al Error de Generalización. También se ha verificado que el algoritmo funciona para más de 2 variables de entrada el cual era el principal objetivo a cumplir en este trabajo.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos CICYT TIN2005-08386-C05-03, CICE PC06-TIC-02025 de la Junta de Andalucía, y el proyecto UJA_08_16_30 de la Universidad de Jaén.

Referencias

- [1] V.M. Rivas, J.J. Merelo, I. Rojas, G. Romero, P.A. Castillo, J. Carpio. Evolving two-dimensional fuzzy systems. *Fuzzy Sets and Systems* 138 Pág. 381-398, 2003.
- [2] Chang, Y.O.; Ayyub, B.M. Fuzzy regression methods a comparative assessment. *Fuzzy Sets*

Tabla 2: Resultados del algoritmo propuesto en el artículo.

ID	FITNESS	ECMN	TAMAÑO
1	54 (± 8)	0.028 (± 0.0007)	34 (± 3)
2	197 (± 35)	0.006 (± 0.003)	41 (± 40)
3	114 (± 8)	0.012 (± 0.001)	18.6 (± 1.2)
4	34 (± 12)	0.046 (± 0.012)	117 (± 36)

Tabla 3: Media de resultados obtenidos para el conjunto de datos MachineCPU.

ALGO-RITMO	FITNESS	ECMN
EvFuzzySystem	181.7 (± 113.9)	0.0197 (± 0.02)
Regr-UN-SVR	49.43 (± 6.02)	0.0204 (± 0.02)
Regr-GAP	164 (± 32.07)	0.007 (± 0.008)
Regr-Thrift	42.84 (± 7.04)	0.023 (± 0.009)

and Systems Pág.119, 197-203, 2003.

- [3] Cordon, O.; Herrera, F.; Hoffmann, F.; and Magdalena, L.; Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases. Advances in Fuzzy Systems: Applications and Theory, World Scientific, 2001.
- [4] Eshelman, L.; Schaffer, J. Real-coded genetic algorithms and interval-schema. In L.D. Whitley (Ed.), Foundation of Genetic Algorithms 2, Pág. 187-202. Morgan Kaufmann.
- [5] Howard, L. M.; D'Angelo, D.J. "The GA-P: A Genetic Algorithm and Genetic Programming Hybrid". IEEE Expert, junio 1995, Pág. 11-15, 1995.
- [6] Karr, C. Applying genetics to fuzzy logic, AI Expert 6. Pág. 26-33, 1991.
- [7] Wright, A. Genetic algorithms for real parameter optimization. In G. J. E. Rawlin (Ed.), Foundations of Genetic Algorithms 1, Pág. 205-218. Morgan Kaufmann.
- [8] ZADEH, L.A. "Fuzzy sets". Information and Control 8 (3): Pág.338-353, 1965.
- [9] KEEL: Knowledge Extraction based on Evolutionary Learning. www.keel.es.

Un algoritmo genético para el ajuste de Sistemas de Clasificación Basados en Reglas Difusas con Conjuntos Intervalo-Valorados Difusos

José Antonio Sanz¹ Alberto Fernández² Humberto Bustince¹ Francisco Herrera²

¹ Dpto. de Automática y Computación (Universidad Pública de Navarra), {joseantonio.sanz,bustince}@unavarra.es

² Dpto. de Ciencias de la Computación e Inteligencia Artificial (Universidad de Granada), {alberto, herrera}@decsai.ugr.es

Resumen

Los Sistemas de Clasificación Basados en Reglas Difusas son una herramienta muy utilizada en Minería de Datos debido a la interpretabilidad proporcionada por el concepto de etiqueta lingüística. Sin embargo, el uso de este tipo de modelos implica un cierto grado de incertidumbre en la definición de las particiones difusas. En este trabajo haremos uso del concepto de conjunto intervalo-valorado difuso para tratar este problema.

El objetivo central de esta contribución es incrementar el rendimiento de los Sistemas de Clasificación Basados en Reglas Difusas utilizando una metodología de ajuste cooperativo tanto de la amplitud del soporte como del desplazamiento lateral (basado en el modelo 2-tuplas) aplicado a las particiones representadas con conjuntos intervalo-valorados difusos. Los resultados empíricos avalan esta propuesta, ya que mejora notablemente el rendimiento del sistema.

Palabras Clave: Sistemas de clasificación basados en reglas difusas, conjuntos intervalo-valorados difusos, ajuste, algoritmos genéticos.

1 Introducción

Una herramienta muy usada para afrontar problemas de clasificación son los Sistemas de Clasificación Basados en Reglas Difusas (SCBRDs) [10]. Sus principales características son que, por un lado, proporcionan un buen rendimiento y, por otro lado, permiten una mayor interpretabilidad de las reglas utilizando etiquetas lingüísticas. Además, ofrecen la posibilidad de

mezclar diferentes tipos de información como la dada por expertos o la obtenida por modelos matemáticos o medidas empíricas.

En los SCBRDs es muy importante la fase de definición de las funciones de pertenencia que utilizamos para representar las etiquetas lingüísticas. A la hora de crear las particiones difusas, podemos hacer uso del conocimiento del experto o definir las etiquetas de manera homogénea sobre el espacio de entrada. En ambos casos, la teoría de Conjuntos Intervalo-Valorados Difusos (CIVDs) [12, 3] nos permite modelar la posible ignorancia inherente a la propia definición de las funciones de pertenencia. Obsérvese que para nosotros el intervalo de pertenencia de un elemento a un conjunto proporciona una cota superior y una inferior para el valor puntual de la pertenencia de dicho elemento al conjunto, valor que nos es desconocido y sólo podemos acotar. En [13], hemos demostrado que el uso de los CIVDs en SCBRDs es útil en el contexto de clasificación con conjuntos de datos no balanceados.

Por otro lado, en ocasiones las particiones difusas no están perfectamente adaptadas al contexto, dado que se mantienen fijas durante el proceso de generación de las reglas difusas. Por tanto, parece necesario realizar una etapa de post-procesamiento en la cual ajustar las etiquetas lingüísticas modeladas en este caso por medio de CIVDs.

El objetivo de este trabajo es mejorar el rendimiento de los SCBRDs. Para ello además de hacer uso del modelo de CIVDs en la fase de inferencia (tras el proceso de obtención de la Base de Reglas), proponemos un modelo de ajuste cooperativo que gestiona tanto la amplitud de las etiquetas lingüísticas como la posición de las mismas mediante un desplazamiento lateral (modelo 2-tuplas [1]). Comprobamos el rendimiento de nuestro modelo con 16 conjuntos de datos seleccionados del repositorio UCI [2].

El trabajo está organizado de la siguiente forma: en la

Sección 2 presentamos el SCBRD en el cual modelamos las etiquetas lingüísticas por medio de CIVDs. Las dos propuestas de ajuste genético junto con el modelo que las combina se introducen en la Sección 3. A continuación, en la Sección 4, mostramos el estudio experimental llevado a cabo y finalizamos el trabajo con las conclusiones principales en la Sección 5.

2 Sistema de Clasificación Basado en Reglas Difusas con Conjuntos Intervalo-Valorados Difusos

En esta sección presentamos el modelo que utiliza CIVDs para representar las etiquetas lingüísticas del SCBRD. El uso de CIVDs permite incrementar el rendimiento del sistema, ya que manejamos la incertidumbre asociada a la construcción ad-hoc de las particiones difusas.

Los CIVDs [3] son una extensión de la teoría de los conjuntos difusos [16] que nacieron en 1975 con el trabajo de Sambuc [12]. En la década de los 80, Gorzalczany y Turksen denominaron por primera vez a estos conjuntos como CIVDs [8, 15].

Denotamos con $L([0, 1])$ al conjunto de todos los subintervalos cerrados del intervalo cerrado $[0, 1]$, es decir:

$$L([0, 1]) = \{\mathbf{x} = [\underline{x}, \bar{x}] | (\underline{x}, \bar{x}) \in [0, 1]^2 \text{ and } \underline{x} \leq \bar{x}\}.$$

Nótese que $L([0, 1])$ es un conjunto parcialmente ordenado con respecto a la relación $\leq_L$ definida de la siguiente forma: dados $\mathbf{x}, \mathbf{y} \in L([0, 1])$:

$$\mathbf{x} \leq_L \mathbf{y} \text{ si y solo si } \underline{x} \leq \underline{y} \text{ y } \bar{x} \leq \bar{y}.$$

Obsérvese que con esta relación de orden el elemento más pequeño es $[0, 0]$ y el más grande es $[1, 1]$.

Definición 1 *Un conjunto intervalo-valorado difuso (CIVD) A en el universo $U \neq \emptyset$ es una función $A : U \rightarrow L([0, 1])$, tal que el grado de pertenencia de $u \in U$ está dado por $A(u) = [\underline{A}(u), \bar{A}(u)] \in L([0, 1])$, donde $\underline{A} : U \rightarrow L([0, 1])$ y $\bar{A} : U \rightarrow L([0, 1])$ son funciones que definen el extremo inferior y el superior del intervalo de pertenencia $A(u)$, respectivamente.*

En este trabajo empleamos el algoritmo de aprendizaje de reglas de Chi y otros [4] para generar la base de conocimiento inicial. Este modelo representa las reglas de la siguiente forma:

Regla R_j : Si x_1 es A_{j1} y ... y x_n es A_{jn}
entonces Clase = C_j con RW_j ,

donde R_j es la etiqueta de la regla j -ésima, $\mathbf{x} = (x_1, \dots, x_n)$ es un vector con n atributos, A_{ji} es un conjunto difuso del antecedente, C_j es la etiqueta de la clase y RW_j es el peso de la regla. Representamos las funciones de pertenencia por medio de funciones triangulares.

A los conjuntos difusos de la base de conocimiento creada por este algoritmo les añadimos el extremo superior para formar los CIVDs. De esta forma estudiamos la influencia de los CIVDs en el proceso de inferencia, no en la fase de generación de las reglas.

Construimos el extremo superior de la siguiente forma: lo centramos en el máximo de la función de pertenencia de la partición difusa y la amplitud de su soporte es un 50% mayor que la del extremo inferior (siendo simétrico a ambos lados).

Además, por trabajar con CIVDs, el peso de la regla estará compuesto por un par (CF_{Lj}, CF_{Uj}) calculado usando el *factor de certeza penalizado* (véase [11]):

$$CF_{Lj} = \frac{\sum_{x_p \in Class C_j} \underline{A_j}(x_p) - \sum_{x_p \notin Class C_j} \underline{A_j}(x_p)}{\sum_{p=1}^m \underline{A_j}(x_p)} \quad (1)$$

$$CF_{Uj} = \frac{\sum_{x_p \in Class C_j} \bar{A_j}(x_p) - \sum_{x_p \notin Class C_j} \bar{A_j}(x_p)}{\sum_{p=1}^m \bar{A_j}(x_p)} \quad (2)$$

El extremo inferior de cada CIVD es el conjunto difuso creado por el algoritmo de aprendizaje de reglas de Chi y otros [4], por tanto CF_{Lj} es igual a RW_j .

Así mismo, hemos de remarcar que el uso de los CIVDs implica dos cambios fundamentales en el proceso de inferencia:

- *Grado de emparejamiento entre el antecedente de la regla y el ejemplo:* Aplicamos la t-norma producto tanto a los extremos inferiores como a los superiores. Por tanto, obtenemos un intervalo.
- *Grado de asociación:* Tomamos la media entre el producto del grado de emparejamiento por el peso de la regla asociados al extremo inferior y el producto del grado de emparejamiento por el peso de la regla asociados al extremo superior.

3 Ajuste de las etiquetas lingüísticas mediante un algoritmo genético

Las funciones de pertenencia, normalmente obtenidas por procesos de normalización o definidas por expertos, permanecen fijas en el proceso de derivación del conjunto de reglas, por lo que las particiones difusas no están adaptadas al contexto de cada variable. Por este motivo es necesario hacer una etapa de post-procesamiento en la cual ajustar las particiones difusas y de esta forma mejorar el rendimiento del SCBRD.

En el resto de la sección vamos a introducir las tres propuestas para realizar dicho ajuste. En primer lu-

gar describimos el modelo de ajuste genético de la amplitud del extremo superior de los CIVDs, a continuación el ajuste genético basado en 2-tuplas para realizar el desplazamiento lateral y por último explicamos el modelo en el que cooperan los dos tipos de ajuste genético anteriores. Por último, describiremos el uso del algoritmo CHC [6] para realizar el proceso de ajuste genético.

3.1 Ajuste de la amplitud de las etiquetas lingüísticas

Como hemos visto en la Sección 2, utilizamos CIVDs, formados añadiendo el extremo superior con una amplitud fija, para representar las etiquetas lingüísticas del SCBRD. Debido a que el experto no tiene la misma incertidumbre definiendo las diferentes funciones de pertenencia, la amplitud del extremo superior no tiene por qué ser igual para todas las etiquetas lingüísticas. Por tanto, realizaremos una etapa de post-procesamiento, por medio de un algoritmo genético, en la que ajustaremos la amplitud del extremo superior de los CIVDs. Es necesario recordar que la amplitud del extremo inferior de cada CIVD permanecerá fija.

La modificación de la amplitud viene dada por un número en el intervalo $[0, 1]$, es decir, desde la situación en la que los extremos están solapados (valor 0) hasta la situación en la que la amplitud del extremo superior es el doble que la del extremo inferior (valor 1). La amplitud del extremo superior variará conforme a los valores intermedios siendo 0,5 la situación inicial del SCBRD, es decir, cuando la amplitud del extremo superior es un 50% mayor que la del extremo inferior.

3.2 Ajuste basado en 2-tuplas para el desplazamiento lateral de las etiquetas lingüísticas

En nuestro modelo inicial hemos definido automáticamente la posición de las diferentes etiquetas, de forma que para cada valor del rango de entrada de cada variable la suma de los grados de pertenencia de las diferentes etiquetas sume 1. Esta distribución de las etiquetas no tiene por qué ser óptima ya que la distribución de los datos no tiene por qué ser uniforme. Por tanto, usamos el ajuste genético basado en 2-tuplas, adaptando una propuesta previa en este tema [1], para realizar desplazamientos laterales de las etiquetas y de esta forma mejorar el rendimiento del SCBRD.

La representación simbólica de un término lingüístico viene dada por un número en el intervalo $[-0.5, 0.5]$ que expresa el rango en el que se puede mover una etiqueta lingüística. Si el número es negativo el desplazamiento

se hará a la izquierda y si es positivo hacia la derecha.

3.3 Ajuste cooperativo para la amplitud y el desplazamiento lateral

Hasta este momento hemos definido dos posibles ajustes de las funciones de pertenencia por separado. Sin embargo, en este trabajo queremos ir más allá y proponer un ajuste en el que cooperen el ajuste de la amplitud y el del desplazamiento lateral. Para ello, basta con definir una representación única dentro del algoritmo genético para ambas posibilidades.

La motivación de esta propuesta reside en que el uso por separado de cada uno de los ajustes anteriormente definidos puede hacer que el sistema obtenga un modelo sub-óptimo. Con esta metodología cooperativa, el motor de búsqueda del algoritmo genético podrá explorar simultáneamente ambas características, lo que nos debe llevar al descubrimiento de soluciones mejor adaptadas al problema y por lo tanto más precisas.

3.4 Algoritmo CHC

Para aplicar el ajuste genético empleamos el algoritmo CHC [6] ya que ofrece una buena relación entre diversidad y convergencia, siendo una buena elección en problemas complejos. Los componentes necesarios para diseñar el proceso son los siguientes:

1. *Codificación:* En todos los modelos empleamos una codificación real, la representación de cada individuo depende del modelo:
 - *Ajuste de amplitud:* Cada gen representa la modificación de la amplitud tal y como hemos explicado en la Subsección 3.1. Por tanto, la longitud del cromosoma será igual al número de etiquetas por el número de variables.
 - *Ajuste lateral:* Cada gen representa el desplazamiento lateral tal y como hemos explicado previamente. Por tanto, la longitud del cromosoma será igual al número de etiquetas por el número de variables.
 - *Ajuste cooperativo:* Cada individuo estará formado por un cromosoma dividido en dos partes, una realizará el ajuste de la amplitud y la otra el ajuste lateral. La representación de cada parte es la misma que hemos explicado para los modelos de ajuste anteriores.
2. *Evaluación:* La función de evaluación será el porcentaje de acierto clásico.
3. *Población inicial:* Según el modelo inicializamos la población de forma diferente:

- *Ajuste de amplitud*: Inicializamos todos los genes del primer individuo con valor ‘0,5’ (el SCBRD inicial). El segundo y el tercer individuo tendrán todos sus genes con valores 0 y 1 respectivamente, mientras que el resto de los individuos serán generados aleatoriamente en el intervalo [0, 1].
 - *Ajuste lateral*: Inicializamos todos los genes del primer individuo con valor ‘0,0’ (el SCBRD inicial), mientras que el resto de los individuos serán generados aleatoriamente en el intervalo [0, 1].
 - *Ajuste cooperativo*: En este modelo inicializamos tres individuos para cubrir las mismas situaciones que el modelo de ajuste de la amplitud, la parte restante de cada cromosoma de estos tres individuos será inicializada como en el modelo de ajuste lateral. El resto de individuos tendrán inicializados todos los genes aleatoriamente en el intervalo [0, 1].
4. *Operador de cruce*: Para todos los modelos usamos el operador *Parent Centric BLX (PCBLX)*, que está basado en el $BLX-\alpha$. Usamos un mecanismo de prevención de incesto, calculando la distancia Hamming entre los padres, que debe ser mayor que un umbral, que inicialmente vale:

$$Umbral = (\#Genes \cdot BITSGEN)/4.0 \quad (3)$$

donde $\#Genes$ es el número de genes del cromosoma y $BITSGEN$ es el número de bits empleados por gen para aplicar la codificación *gray*, dado que usamos una representación real.

Para trabajar con el modelo de ajuste cooperativo, realizamos el cruce de la siguiente forma: cruzamos entre sí las partes del cromosoma que representen al mismo tipo de ajuste. En este proceso se generan cuatro descendientes, de los que seleccionamos los dos mejores.

5. *Reinicialización*: Cuando el valor del umbral sea menor que cero, todos los cromosomas serán regenerados aleatoriamente en el intervalo [0, 1]. Además, incluiremos en la población la mejor solución obtenida hasta ese momento para incrementar la convergencia del algoritmo.

4 Estudio experimental

En este estudio, nuestro objetivo es observar la mejora obtenida en los SCBRDs al realizar la etapa de post-procesamiento por medio de los ajustes genéticos. Además, queremos comprobar que el comportamiento de la sinergia entre el ajuste de la amplitud y el ajuste lateral mejore al de las propuestas individuales.

En el resto de la sección, primero describimos el marco experimental junto con los parámetros empleados en el estudio, a continuación introducimos los test estadísticos empleados y para finalizar mostramos los resultados empíricos obtenidos.

4.1 Marco experimental

Hemos analizado el rendimiento de las diferentes propuestas en 16 conjuntos de datos seleccionados del repositorio UCI [2]. En la Tabla 1 podemos ver las siguientes características de los *data-sets*: número de ejemplos ($\#Ej.$), número de atributos ($\#Atr.$) y número de clases ($\#Clas.$).

Tabla 1: Descripción de los conjuntos de datos empleados en el estudio.

Conjunto de datos	#Ej.	#Atr.	#Clas.
Balance	625	4	3
Bupa	345	6	2
Cleveland	297	13	5
Ecoli	336	7	8
Glass	214	9	6
Haberman	306	3	2
Iris	150	4	3
Magic	1902	10	2
New-Thyroid	215	5	3
Page-blocks	548	10	5
Pensabed	1099	16	10
Pima	768	8	2
Ring	768	8	2
Vehicle	846	18	4
Wine	178	13	3
Wisconsin	683	9	2

Para realizar la experimentación hemos empleado el modelo *5-folder cross-validation*, es decir, 5 particiones con el 20% de los datos cada una, combinando 4 de ellas para el entrenamiento y usando la restante para test. Como resultado de cada conjunto de datos empleamos la media de las 5 particiones.

La configuración del SCBRD es la siguiente: usamos 3 etiquetas por partición difusa, como operador de conjunción usamos la t-norma producto, para calcular el peso de las reglas usamos el *factor de certeza penalizado* y como método de razonamiento difuso empleamos el de la regla ganadora.

Todos los modelos de ajuste genético explicados en la Sección 3 emplean poblaciones formadas por 50 individuos, 30 bits por gen para la codificación *gray* y el número de evaluaciones es $5.000 \cdot \text{número de variables}$.

4.2 Tests estadísticos para la comparativa del rendimiento

En esta contribución, usamos las técnicas de comprobación de hipótesis para proporcionar un soporte estadístico al análisis de los resultados. Concretamente, usaremos tests no paramétricos, dado que las condiciones iniciales que garantizan la fiabilidad de los tests paramétricos pueden no satisfacerse, haciendo que el análisis estadístico pierda credibilidad con estos tipos

Tabla 2: Resultados en Entrenamiento (Entr.) y Test (Test) de los diferentes métodos.

Conjunto de datos	Chi		Chi.Lat		Chi.CIVD		Chi.CIVD.Amp		Chi.CIVD.Lat		Chi.CIVD.Coop	
	Entr.	Test	Entr.	Test	Entr.	Test	Entr.	Test	Entr.	Test	Entr.	Test
Balance	91,62	89,92	92,18	89,76	91,10	90,24	92,18	90,88	92,30	91,52	92,30	90,08
Bupa	60,73	57,68	77,45	58,84	59,71	58,26	61,09	57,68	68,87	59,42	75,85	61,16
Cleveland	92,22	36,01	94,76	40,75	90,02	53,21	92,81	50,50	94,00	51,51	95,86	53,21
Ecoli	76,18	72,64	71,26	64,84	69,98	67,58	77,37	71,45	68,35	61,28	88,35	82,16
Glass	66,28	57,95	75,09	60,30	55,94	51,87	69,21	59,86	77,21	64,50	78,73	63,57
Haberman	74,57	72,88	78,42	72,56	73,83	73,53	74,73	72,88	75,88	72,88	78,51	73,22
Iris	92,94	92,67	98,32	94,67	93,61	92,00	97,31	94,67	97,98	96,00	98,49	94,00
Magic	75,98	74,87	83,64	78,86	73,31	72,66	77,06	75,08	81,63	78,92	83,37	79,13
New-Thyroid	86,32	84,65	98,25	94,88	81,75	80,93	86,43	85,12	90,29	86,05	99,18	94,88
Pageblocks	92,73	91,42	94,38	93,06	91,50	90,88	92,87	91,42	93,14	91,24	94,42	93,61
Penbased	98,66	94,27	99,50	93,09	95,31	92,27	98,95	94,36	99,11	95,55	99,59	95,09
Pima	75,45	72,53	81,94	73,69	69,71	67,97	76,30	72,39	78,55	74,08	81,55	74,22
Ring	59,53	52,70	96,92	85,81	51,17	50,41	59,66	52,43	93,81	85,81	96,14	88,78
Vehicle	65,85	60,88	78,78	66,55	54,87	51,55	68,99	60,05	74,08	63,24	77,89	66,08
Wine	98,73	92,67	100,00	93,81	98,02	94,90	99,86	94,33	100,00	94,38	100,00	94,92
Wisconsin	98,17	90,49	99,27	93,71	97,65	96,05	98,39	96,05	98,68	96,19	99,08	95,75
Media	81,62	74,64	88,76	78,45	77,97	74,02	82,70	76,20	86,49	78,91	89,96	81,24

de tests [5, 7]. Concretamente, usaremos el test de rangos de Wilcoxon [14] como procedimiento estadístico no paramétrico para realizar comparaciones entre dos algoritmos; Usamos el test de Iman-Davenport [14] para detectar diferencias estadísticas entre un grupo de resultados y el test post-hoc de Holm [9] para encontrar qué algoritmos rechazan la hipótesis de igualdad con respecto a un método de control seleccionado.

4.3 Resultados empíricos

Como hemos dicho anteriormente, queremos comprobar la importancia de la etapa de post-procesamiento para mejorar el rendimiento de los SCBRDs. Además, el principal objetivo del estudio es comprobar que el comportamiento de la propuesta de cooperación entre los ajustes genéticos de la amplitud y el lateral sea el mejor.

La Tabla 2 muestra los resultados obtenidos tanto en entrenamiento como en test por las diferentes propuestas en cada conjunto de datos. En primer lugar aparecen los resultados obtenidos por el modelo de Chi et al. básico, después los de esta propuesta con ajuste lateral, en tercer lugar los resultados del SCBRD con CIVD y a continuación podemos ver, en este orden, los resultados de los modelos de ajuste de amplitud, lateral y cooperativo aplicados al SCBRD con CIVDs.

En los modelos de ajuste es usual encontrar un comportamiento de sobreaprendizaje, especialmente cuando dicho ajuste se guía específicamente mediante el porcentaje de acierto. Sin embargo, tal y como se puede apreciar en la Tabla 2, nuestro modelo de ajuste cooperativo tiene un buen equilibrio entre el rendimiento de entrenamiento y test, lo cual es una propiedad muy deseable en este tipo de metodología.

Para llevar a cabo nuestro estudio, primero mostramos en la Figura 1 el ranking de los distintos métodos utilizados asignando las posiciones de estos algoritmos en función del rendimiento para cada conjunto de datos

y obteniendo a continuación el valor medio. Podemos observar que claramente la mejor posición la obtiene nuestra propuesta de ajuste cooperativo, seguida del ajuste lateral realizada sobre el SCBRD con CIVDs y el ajuste lateral realizado sobre el SCBRD clásico.

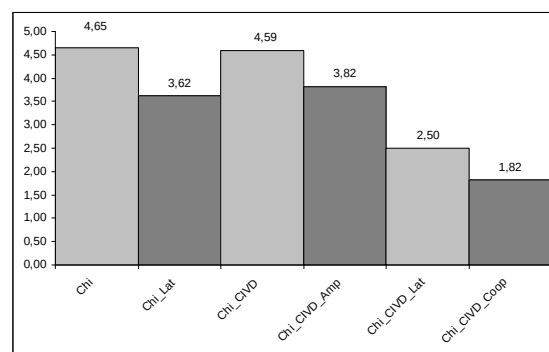


Figura 1: Ranking de las diferentes propuestas.

A continuación, realizamos un test de Iman-Davenport para detectar diferencias significativas entre los resultados de estas propuestas. El “p-value” asociado es cercano a cero, lo que implica que hay diferencias significativas entre los resultados y así, debemos aplicar un test post-hoc, en este caso el test de Holm (Tabla 3). Queremos comparar nuestra metodología de ajuste contra el resto de enfoques de ajuste, así como para los SCBRDs obtenidos por el modelo de Chi, a nivel básico e incluyendo CIVDs.

Tabla 3: Test de Holm para comparar todas las metodologías. La propuesta que realiza el ajuste cooperativo es utilizada como método de control.

<i>i</i>	Algoritmo	<i>z</i>	<i>p</i>	α/i	Hipótesis ($\alpha = 0,05$)
5	Chi	4,400	1,082E-5	0,01	Rechazado para Chi.CIVD.Coop
4	Chi.CIVD	4,308	1,644E-5	0,0125	Rechazado para Chi.CIVD.Coop
3	Chi.CIVD.Amp	3,117	0,002	0,017	Rechazado para Chi.CIVD.Coop
2	Chi.Lat	2,796	0,005	0,025	Rechazado para Chi.CIVD.Coop
1	Chi.CIVD.Lat-	1,054	0,292	0,05	No Rechazado

A partir de los resultados del test de Holm se demuestra que nuestro modelo de ajuste cooperativo es estadísticamente mejor que el resto de propuestas, si

bien hemos de aplicar un test de Wilcoxon (Tabla 4) para observar diferencias significativas frente al modelo de ajuste lateral aplicado al SCBRD con CIVDs.

Tabla 4: Test de Wilcoxon para comparar la propuesta de ajuste cooperativo contra el SCBRD con CIVDs con ajuste lateral.

Comparación	R^+	R^-	Hipótesis ($\alpha = 0.05$)	p-value
Chi_CIVD_Coop vs. Chi_CIVD_Lat	118,0	35,0	Rechazado para Chi_CIVD_Coop	0,049

En resumen, hemos mostrado la buena sinergia de ambos modelos de ajuste en una metodología cooperativa que explota los beneficios que provee cada propuesta de ajuste realizada individualmente.

5 Conclusiones

En este trabajo hemos propuesto un modelo de post-procesamiento genético sobre SCBRDs con CIVDs que realiza simultáneamente un ajuste de la amplitud del extremo superior de los CIVDs y el ajuste basado en 2-tuplas para el desplazamiento de las particiones difusas con el objetivo de mejorar el rendimiento global del sistema difuso frente a las metodologías de ajuste realizadas individualmente.

Los resultados empíricos obtenidos demuestran que nuestro método de ajuste cooperativo es estadísticamente superior al resto de modelos utilizados en el estudio. De este modo, no solo hacemos un uso correcto del manejo de la incertidumbre mediante los CIVDs, si no que nuestro ajuste cooperativo permite adaptar eficazmente las funciones de pertenencia al contexto de cada problema para obtener de este modo una clasificación más precisa que aplicando este tipo de ajustes por separado.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Educación y Ciencia con los proyectos TIN2008-06681-C06-01 y TIN2007-65981.

Referencias

- [1] R. Alcalá, J. Alcalá-Fdez y F. Herrera: A proposal for the Genetic Lateral Tuning of Linguistic Fuzzy Systems and its Interaction with Rule Selection IEEE Transactions on Fuzzy Systems **15**(4) (2007) 616–635
- [2] Asuncion, A., Newman, D.: UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences. <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [3] Bustince, H., Montero, J., Barrenechea, E., Gomez, D.: A survey of Interval-Valued Fuzzy Sets. In: Handbook of Granular Computing. Addison Wesley (2008) 491 – 516
- [4] Chi, Z., Yan, H., Pham, T.: Fuzzy algorithms with applications to image processing and pattern recognition. World Scientific (1996)
- [5] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [6] Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Foundations of Genetic Algorithms. Morgan Kaufman (1991) 265–283
- [7] S. García and F. Herrera. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
- [8] Gorlaczany, M. B., 1987. A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets and Systems* 21, 1–17.
- [9] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.
- [10] Ishibuchi, H., Nakashima, T., Nii, M.: Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining. Springer-Verlag (2004)
- [11] Ishibuchi, H., Yamamoto, T.: Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems* **13** (2005) 428–435
- [12] Sambuc, R., 1975. Function ϕ -fous, application a l’aide au diagnostic en pathologie thyroïdienne. Ph.D. thesis, University of Marseille.
- [13] Sanz, J., Fernandez, A., Bustince, H., Herrera, F.: A First Study on the Use of Interval-valued Fuzzy Sets with Genetic Tuning for Classification with Imbalanced Data-sets. *Lecture Notes on Computer Science* **5572** (2009) 581 – 588
- [14] D. Sheskin. *Handbook of parametric and non-parametric statistical procedures*. Chapman & Hall/CRC, second edition, 2006.
- [15] Turksen, I. B., 1995. Fuzzy normal forms. *Fuzzy Sets and Systems* 69, 319–346.
- [16] Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning–I–III. *Information Sciences* **8**(3) (1975) 199–249

FUZZYCN2: UN ALGORITMO DE EXTRACCIÓN DE LISTAS DE REGLAS DIFUSAS

Pablo Martín Muñoz¹ Francisco José Moreno Velo²

¹Unidad para la Dirección Estratégica, Universidad de Huelva, pablo.martin@sc.uhu.es

²Departamento de Tecnologías de la Información, Universidad de Huelva, francisco.moreno@dti.uhu.es

Resumen

La mayoría de los algoritmos de extracción de reglas de clasificación difusas producen reglas con antecedentes conjuntivos en los que aparecen todos los atributos del sistema. Con este tipo de antecedentes, el número de reglas crece exponencialmente en función del número de atributos. Este trabajo presenta un nuevo algoritmo de extracción de reglas de clasificación difusas llamado FuzzyCN2. Se trata de una versión difusa del conocido algoritmo CN2 y genera una lista ordenada de reglas. El algoritmo produce reglas donde no todos los atributos están presentes y permite, además, utilizar modificadores lingüísticos en estas reglas. Esto permite reducir el número de reglas con respecto a los resultados de los algoritmos convencionales.

Palabras Clave: Clasificadores Difusos, Extracción de Reglas Difusas, Listas Ordenadas de Reglas Difusas, Modificadores Lingüísticos.

1 INTRODUCCIÓN

Los sistemas basados en reglas difusas han demostrado ser una opción muy adecuada en problemas de clasificación. Por un lado y frente a alternativas como las redes neuronales, las máquinas de vectores soporte o los vecinos más cercanos, el uso de reglas difusas permite la inducción de modelos interpretables. Por otro lado y frente a los sistemas expertos clásicos, la salida de un clasificador difuso no se limita a la clase seleccionada, sino que ofrece el grado de activación de cada clase. Esto resulta especialmente útil en problemas con atributos continuos. En estos casos los sistemas expertos clásicos se basan en consultas del tipo *atributo* $\geq$ *valor* [9], por lo que estos sistemas presentan cambios abruptos alrededor de estos valores. Por su parte, el comportamiento de los clasificadores difusos presenta cambios suaves en el grado de activación de cada clase, lo que permite detectar situaciones en las que la

decisión es más arriesgada al involucrar a clases con grados de activación muy próximos.

El algoritmo de extracción de reglas difusas más conocido es el propuesto por Wang y Mendel [13]. Este algoritmo se basa, en primer lugar, en definir las etiquetas lingüísticas asociadas a cada atributo y, a continuación, generar la regla difusa conjuntiva más activa para cada uno de las instancias del conjunto de datos. Cada una de las reglas generadas en este algoritmo contiene un término para cada atributo del sistema. Esto provoca un crecimiento exponencial en el número de reglas generadas en función del número de atributos del sistema. Existen muchas propuestas para reducir el número de reglas generadas, pero la mayoría mantiene la forma de las reglas, es decir, utilizan reglas conjuntivas que contienen a todos los atributos (por ejemplo [7][10]).

Este enfoque contrasta con el seguido habitualmente en los algoritmos clásicos de extracción de reglas. El objetivo de estos algoritmos es minimizar el número de reglas generadas. Para ello tratan de inducir las reglas más generales posible, esto es, reglas en las que no aparezcan todos los atributos. Entre estos algoritmos los más conocidos son ID3 [8], que genera árboles de decisión, y la familia de algoritmos AQ [5], que genera conjuntos de reglas de clasificación. Tomando ideas de estos algoritmos, Clark y Niblett propusieron en 1989 el algoritmo CN2 [4], que induce listas ordenadas de reglas de clasificación. Posteriormente Clark y Boswell plantearon algunas mejoras a este algoritmo [3].

Algunas propuestas para inducir reglas de clasificación difusas se basan en versiones difusas de los algoritmos anteriores. Existen numerosas adaptaciones del algoritmo ID3 para generar árboles de decisión difusos [2][11][12][14]. El algoritmo CN2 también ha sido adaptado al caso difuso [16], pero en este caso se trata de la versión del algoritmo que induce conjuntos desordenados de reglas.

Este trabajo presenta una adaptación diferente del algoritmo CN2 para extraer listas ordenadas de reglas difusas.

La propuesta contiene no sólo un modo de adaptar los diferentes procedimientos y métricas de CN2 al caso difuso, sino que también añade la capacidad de utilizar modificadores lingüísticos como parte de las reglas inducidas, lo que permite extraer reglas más compactas y expresivas que las inducidas en la mayoría de algoritmos de clasificación difusos.

2 EL ALGORITMO CN2

El algoritmo CN2 fue propuesto sobre la base de los algoritmos AQ tratando de introducir la capacidad de tratamiento de ruido de los algoritmos TDIDT (Top Down Induction of Decision Trees). El algoritmo CN2 toma de los algoritmos AQ la idea de encontrar el mejor conjunto de reglas por medio de un conjunto de búsquedas en haz en paralelo (lo que se conoce como una búsqueda en estrella) y de los algoritmos TDIDT la idea de finalizar la búsqueda cuando las reglas encontradas no superan cierto umbral de significancia estadística (técnicas de poda).

El algoritmo se basa en un bucle externo en el que, dado un conjunto de ejemplos de clasificación, se encuentra la mejor regla para dicho conjunto y se eliminan los ejemplos cubiertos por dicha regla. El bucle finaliza cuando el conjunto de ejemplos queda vacío o cuando no se encuentra ninguna regla con el mínimo nivel de significancia exigido. Puesto que las reglas se generan sobre conjuntos de ejemplos en los que se han eliminado aquellos que han sido cubierto por reglas anteriores, el resultado debe interpretarse como una lista ordenada de reglas, es decir que una regla sólo debe ser considerada si ninguna de las reglas precedentes se encuentra activa.

La búsqueda de la mejor regla asociada a un conjunto de ejemplos se basa en una búsqueda en estrella. En cada iteración las reglas candidatas son especializadas añadiéndoles un nuevo selector (una consulta que compara un atributo con uno de sus valores). En los algoritmos AQ, los términos a añadir se obtienen a partir de uno de los ejemplos positivos del conjunto (*semilla*). El algoritmo CN2 amplía este conjunto a todos los posibles selectores. Para estudiar la bondad de las reglas el algoritmo utiliza dos heurísticas. Por un lado, la entropía de información se utiliza para medir la calidad de las reglas candidatas. Por otro lado, a las reglas se les exige un mínimo de significancia estadística que se mide por medio de una heurística conocida como *likelihood ratio*. En un trabajo posterior, presentado en 1991, se propuso la utilización de la estimación del error laplaciano como heurística de calidad, en sustitución de la entropía de información. Este trabajo contiene también una versión del algoritmo que genera listas de reglas desordenadas.

El algoritmo FuzzyCN2 presentado en este trabajo se basa en una versión difusa del algoritmo CN2 original, que

genera listas de reglas ordenadas. La heurística utilizada como medida de calidad es una versión difusa de la estimación del error laplaciano y la medida de la significancia estadística se basa en un valor mínimo de la cobertura de las reglas sobre el conjunto de ejemplos.

3 LISTAS ORDENADAS DE REGLAS DIFUSAS

El algoritmo FuzzyCN2 genera una lista ordenada de reglas difusas en lugar de un conjunto desordenado de reglas. Ambas representaciones tienen sus respectivas ventajas y desventajas en cuanto a interpretabilidad. En una lista ordenada de reglas la interpretación de cada regla depende de cuáles son las reglas que le preceden, ya que dicha regla sólo se activa en la medida en que las predecesoras no lo hacen. Por su parte, los conjuntos de reglas desordenadas pueden contener reglas que cubren el mismo espacio de entrada, es decir, reglas que se activan al mismo tiempo. En este caso es necesario algún tipo de mecanismo adicional que decide cuál de las reglas se impone cuando surgen estos conflictos, de manera que las reglas no pueden considerarse estrictamente independientes de las demás si permiten la existencia de estos conflictos.

La tabla 1 muestra estos dos tipos de listas de reglas difusas. El mecanismo de inferencia en una lista de reglas desordenadas considera el grado de activación del consecuente de cada regla, $\mu(C_i)$, como el grado de activación de su antecedente, $\mu(A_i)$. En el caso de las listas ordenadas de reglas el mecanismo de inferencia es diferente. En este caso hay una condición *else* implícita respecto a las reglas precedentes por lo que el grado de activación del consecuente de la regla, $\mu(C_i)$, depende tanto del grado de activación de su antecedente, $\mu(A_i)$, como del grado de activación de sus predecesoras, $\prod(1-\mu(A_j))$.

Tabla 1: Diferentes tipos de conjuntos de reglas

Conjunto	Lista ordenada
if(A1) then C1 if(A2) then C2 if(A3) then C3	if(A1) then C1 elseif(A2) then C2 elseif(A3) then C3
$\mu(C1) = \mu(A1)$ $\mu(C2) = \mu(A2)$ $\mu(C3) = \mu(A3)$	$\mu(C1) = \mu(A1)$ $\mu(C2) = (1-\mu(A1)) \cdot \mu(A2)$ $\mu(C3) = (1-\mu(A1)) \cdot (1-\mu(A2)) \cdot \mu(A3)$

4 EL ALGORITMO FUZZYCN2

El algoritmo FuzzyCN2 consiste en una versión difusa del algoritmo CN2 clásico. Para realizar esta traducción es necesario resolver algunas cuestiones:

- a) ¿Cómo se eliminan los ejemplos cubiertos por una regla difusa? Puesto que las reglas difusas sólo cubren los ejemplos en un cierto grado, estos ejemplos deberían ser eliminados sólo parcialmente.
- b) ¿Cómo se representan conjuntos de ejemplos que han sido eliminados parcialmente?
- c) ¿Cómo se calcula el grado de cubrimiento de una regla difusa sobre un ejemplo eliminado parcialmente?
- d) ¿Cómo se traducen las heurísticas de búsqueda (estimación del error laplaciano y test de significancia estadística) al caso difuso?

Las siguientes definiciones permiten resolver estas cuestiones:

Definición 1. Un selector difuso es un término difuso que relaciona un atributo y una de sus etiquetas lingüísticas. Esta relación puede ser la igualdad, es decir, *atributo igual a etiqueta*, o puede utilizar modificadores lingüísticos, como *atributo mayor que etiqueta*. La Figura 1 muestra los diferentes selectores considerados en FuzzyCN2.

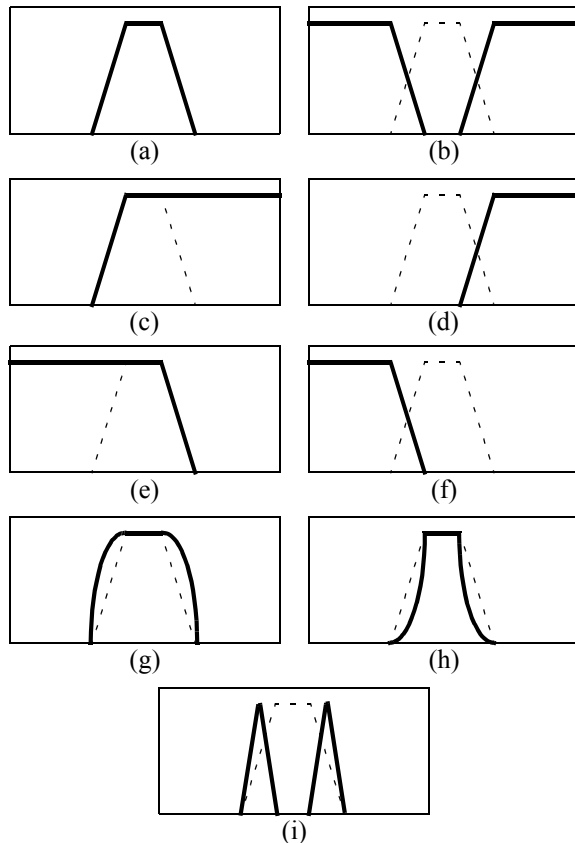


Figura 1: Selectores utilizados en FuzzyCN2 por cada atributo (*attr*) y etiqueta (*lb*). (a) *attr es igual a lb*; (b) *attr es distinto de lb*; (c) *attr es mayor o igual que lb*; (d) *attr es mayor que lb*; (e) *attr es menor o igual que lb*; (f) *attr es menor que lb*; (g) *attr es aproximadamente igual a lb*; (h) *attr es fuertemente igual a lb*; (i) *attr es casi igual a lb*.

Definición 2. Un complejo es una conjunción de selectores difusos. Se utilizan como antecedentes en las reglas inducidas por CN2.

Definición 3. Un conjunto difuso de instancias es un conjunto de ejemplos de clasificación en el que cada ejemplo tiene asociado un grado de activación. Un conjunto concreto (*crisp*) de instancias puede interpretarse como un conjunto difuso de instancias en el que el grado de activación de todos los ejemplos es la unidad. Los conjuntos difusos de instancias pueden ser truncados, eliminando aquellos ejemplos cuyo grado de activación no alcance un cierto nivel de corte.

Definición 4. Dado un complejo C y un ejemplo e con grado de activación $\mu(e)$, el soporte de C sobre e se define como el producto entre el grado de activación del complejo al aplicarse sobre el ejemplo y el grado de activación del ejemplo, es decir,

$$\text{soporte}(C, e) = \mu(C, e) \cdot \mu(e) \quad (1)$$

Definición 5. El soporte de un complejo C sobre un conjunto difuso de instancias S se define como la suma del soporte de C sobre cada uno de los ejemplos de S .

$$\text{soporte}(C, S) = \sum_{e \in S} \text{soporte}(C, e) \quad (2)$$

Definición 6. El soporte de un complejo C sobre una clase c y un conjunto difuso de instancias S se define como la suma del soporte de C sobre aquellos ejemplos de S que pertenecen a la clase c .

$$\text{soporte}(C, S, c) = \sum_{e \in S, \text{clase}(e) = c} \text{soporte}(C, e) \quad (3)$$

Definición 7. El soporte relativo de un complejo C sobre un conjunto difuso de ejemplos S se define como

$$\text{soporte-relativo}(C, S) = \frac{\text{soporte}(C, S)}{\sum_{e \in S} \mu(e)} \quad (4)$$

Definición 8. La desactivación de un conjunto difuso de ejemplos S con respecto a un complejo C consiste en actualizar el grado de activación de cada ejemplo e de la siguiente forma:

$$\mu_{\text{desactivado}}(e) = \mu(e) \cdot (1 - \mu(C, e)) \quad (5)$$

Definición 9. Se dice que un complejo C es significativo con respecto a un conjunto difuso de ejemplos S si el soporte relativo de C sobre S es mayor que un cierto valor

(*soporte relativo mínimo*) y el soporte de C sobre S es mayor que un cierto valor (*soporte absoluto mínimo*).

Definición 10. La estimación difusa del error laplaciano de un complejo C es una medida de la calidad de dicho complejo definida como

$$\text{Laplaciano}(C, S) = \frac{\text{soporte}(C, S, c) + 1}{\text{soporte}(C, S) + k} \quad (6)$$

donde c es la clase de mayor soporte y k es el número de clases.

La siguiente tabla muestra el pseudocódigo del algoritmo FuzzyCN2. El algoritmo toma como entrada un conjunto de ejemplos de clasificación y genera una lista ordenada de reglas difusas.

Tabla 2: El algoritmo FuzzyCN2

```

procedure FuzzyCN2(InstanceSet)
returns RuleList
  RuleList  $\leftarrow \emptyset$ 
  FuzzyInstSet  $\leftarrow$  AddActivationDegree(InstanceSet)
  Complex  $\leftarrow$  FindBestComplex(FuzzyInstSet)
  while IsNotNil(Complex) and IsNotEmpty(FuzzyInstSet)
    Conseq  $\leftarrow$  MostCoveredClass(FuzzyInstSet, Complex)
    NewRule  $\leftarrow$  "if( Complex ) then class = Conseq"
    RuleList  $\leftarrow$  Add(RuleList, NewRule)
    FuzzyInstSet  $\leftarrow$  Deactivate(FuzzyInstSet, Complex)
    FuzzyInstSet  $\leftarrow$  Truncate(FuzzyInstSet)
    Complex  $\leftarrow$  FindBestComplex(FuzzyInstSet)
  endwhile
  return RuleList
end

procedure FindBestComplex(FuzzyInstSet)
returns BestComplex
  BestComplex  $\leftarrow$  nil
  Selectors  $\leftarrow$  CreateAllSelectors()
  Star  $\leftarrow$  InitializeStar()
  while IsNotEmpty(Star)
    NewStar  $\leftarrow$  SpecializeStar(Star, Selectors)
    NewStar  $\leftarrow$  SignificanceTest(NewStar, FuzzyInstSet)
    NewStar  $\leftarrow$  QualityTest(NewStar, FuzzyInstSet)
    BestComplex  $\leftarrow$  BetterQuality(NewStar, BestComplex)
    Star  $\leftarrow$  NewStar
  endwhile
  return BestComplex
end

```

El primer paso del algoritmo FuzzyCN2 consiste en transformar el conjunto de ejemplos de clasificación en un con-

junto de ejemplos difusos añadiendo a cada ejemplo un grado de activación de la unidad (método *AddActivationDegree*). Al igual que el algoritmo CN2, el algoritmo FuzzyCN2 está dividido en dos bucles. El bucle externo consiste en buscar una nueva regla difusa (el método *FindBestComplex* genera el antecedente de la nueva regla y el método *MostCoveredClass* el consecuente), añadirla a la lista ordenada (método *Add*), eliminar parcialmente los ejemplos cubiertos por la regla (método *Deactivate*) y truncar el conjunto difuso de ejemplos para eliminar aquellos que estén poco activos (método *Truncate*). El bucle termina cuando no quedan ejemplos por cubrir o no se encuentra ninguna regla con los niveles de calidad y de significancia exigida.

El bucle interno se desarrolla en el método *FindBestComplex*. Este método toma un conjunto difuso de ejemplos y busca el mejor complejo en términos de significancia y calidad. El procedimiento considera un conjunto de selectores entre los que se incluyen los modificadores lingüísticos escogidos por el usuario (método *CreateAllSelectors*). La estrella contiene un conjunto de complejos candidatos. Inicialmente la estrella contiene un complejo vacío. En cada iteración se estudian todos los complejos obtenidos añadiendo uno de los selectores a cada uno de los complejos de la estrella (método *SpecializeStar*). De estos complejos se eliminan aquellos en los que algún atributo aparece más de dos veces o en los que se repite algún selector. También se eliminan aquellos en los que no se alcanza el nivel mínimo de significancia (Definición 9). Tras esto se calcula la heurística de calidad sobre los complejos (Definición 10) y se genera una nueva estrella del tamaño fijado (método *QualityTest*). El bucle termina cuando la estrella queda vacía, devolviéndose el complejo de mayor calidad encontrado en la búsqueda.

5 ESTUDIO EXPERIMENTAL

A continuación se presenta un estudio comparativo del funcionamiento del algoritmo FuzzyCN2 frente a otros algoritmos de clasificación, tanto difusos como clásicos. Para ello se han escogido 11 conjuntos de datos de clasificación del repositorio UCI [1] con atributos continuos. En el caso de los algoritmos difusos, los atributos han sido descritos por medio de tres etiquetas lingüísticas, tal y como muestra la figura 2b. Con respecto a los algoritmos

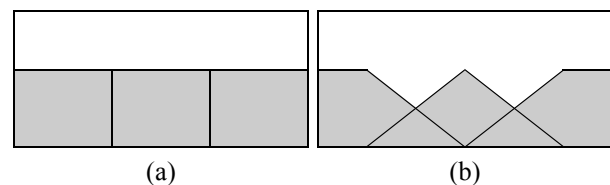


Figura 2: Discretización empleada en el estudio experimental (a) en los algoritmos clásicos; (b) en los algoritmos difusos.

Tabla 3: Número de reglas promedio generadas por los distintos algoritmos

	ID3	C4.5	Ripper	W-M	FuzzyID3	FuzzyCN2 v1	FuzzyCN2 v2	FuzzyCN2 v3
haberman	17.6	3.4	2.1	17.6	15.3	10.1	11.5	17.7
iris	8.9	4.8	3.3	20.4	5.8	5.0	5.8	6.4
bupa	40.4	8.8	2.1	52.8	16.9	14.7	16.3	22.1
ecoli	51.0	19.6	8.6	71.4	20.7	12.1	11.7	11.9
pima	115.7	16.0	3.3	152.6	30.0	24.6	28.3	34.4
glass	41.4	22.2	7.2	55.2	17.5	9.4	10.3	12.6
wbc	54.0	10.6	5.9	231.4	19.3	15.4	14.8	15.4
wine	25.7	12.2	6.0	152.1	18.4	8.7	9.0	8.5
cleveland	143.4	44.6	3.8	249.1	60.2	22.5	24.8	25.4
vehicle	283.7	90.4	12.5	462.9	38.2	28.6	33.3	36.1
ion	56.4	23.4	7.7	239.4	28.2	18.7	18.6	18.6
media	76.2	23.3	5.7	155.0	24.6	15.4	16.8	19.0

Tabla 4: Porcentaje de error promedio sobre el conjunto de entrenamiento

	ID3	C4.5	Ripper	W-M	FuzzyID3	FuzzyCN2 v1	FuzzyCN2 v2	FuzzyCN2 v3
haberman	24.1%	25.4%	25.4%	24.8%	24.7%	24.8%	24.2%	21.5%
iris	2.1%	2.1%	2.1%	2.7%	3.2%	4.4%	2.4%	2.6%
bupa	33.7%	37.9%	39.0%	36.9%	41.3%	37.0%	33.3%	29.7%
ecoli	12.1%	17.2%	17.5%	15.1%	19.5%	20.7%	12.9%	12.3%
pima	18.6%	23.2%	25.4%	22.2%	27.7%	21.5%	19.3%	16.4%
glass	22.9%	29.3%	31.2%	26.0%	48.2%	37.7%	31.7%	21.8%
wbc	0.4%	3.1%	3.0%	1.4%	2.6%	2.6%	1.6%	1.1%
wine	0.1%	5.2%	6.5%	0.0%	4.6%	3.5%	1.4%	1.2%
cleveland	1.4%	26.5%	40.9%	4.1%	29.9%	27.1%	22.5%	18.4%
vehicle	13.2%	25.7%	38.4%	23.9%	40.1%	30.3%	25.7%	20.4%
ion	0.6%	5.2%	7.0%	1.4%	6.9%	3.2%	1.6%	1.3%
media	11.7%	18.3%	21.5%	14.4%	22.6%	19.3%	16.1%	13.3%

Tabla 5: Porcentaje de error promedio sobre el conjunto de test

	ID3	C4.5	Ripper	W-M	FuzzyID3	FuzzyCN2 v1	FuzzyCN2 v2	FuzzyCN2 v3
haberman	28.4%	26.1%	26.5%	26.1%	29.4%	28.1%	28.1%	30.4%
iris	13.3%	12.0%	12.0%	5.3%	3.3%	4.7%	4.7%	2.7%
bupa	52.4%	44.0%	39.8%	42.6%	44.6%	41.3%	38.9%	37.4%
ecoli	32.1%	27.7%	29.7%	19.3%	24.5%	23.8%	15.2%	15.1%
pima	43.2%	27.8%	28.6%	27.2%	29.5%	27.5%	27.1%	25.5%
glass	56.8%	52.6%	50.3%	34.5%	61.1%	41.4%	39.7%	30.6%
wbc	5.0%	4.2%	4.4%	5.4%	3.5%	4.5%	4.5%	4.1%
wine	10.2%	12.4%	13.5%	10.7%	9.5%	6.7%	3.9%	4.5%
cleveland	57.5%	44.0%	46.1%	60.0%	46.4%	46.7%	46.3%	48.0%
vehicle	46.4%	44.3%	50.6%	38.2%	42.3%	35.4%	31.7%	32.0%
ion	12.9%	13.6%	13.7%	7.1%	12.2%	12.5%	11.3%	8.5%
media	32.6%	28.1%	28.7%	25.1%	27.8%	24.8%	22.9%	21.7%

clásicos, los datos han sido discretizados en tres intervalos del mismo tamaño con un significado equivalente al caso difuso, tal y como muestra la figura 2a. Los resultados del estudio utilizando una validación cruzada en 10 trozos se muestran en las tablas 3, 4 y 5. En el caso de los algoritmos clásicos, se han estudiado los algoritmos ID3, C4.5 y

RIPPER utilizando para ello la herramienta Weka [15] y la configuración por defecto de estos algoritmos. En cuanto a los algoritmos difusos, se han estudiado los algoritmos de Wang y Mendel y FuzzyID3 realizando el estudio con el entorno Xfuzzy 3 [6].

Las últimas columnas de las tablas muestran el resultado del estudio para tres configuraciones del algoritmo FuzzyCN2. En la primera de ellas (v1) no se han utilizado modificadores lingüísticos en los selectores estudiados. En la segunda configuración (v2) se han incluido los modificadores lingüísticos mayor, menor, mayor o igual y menor o igual. En la tercera configuración (v3) se han considerado todos los modificadores lingüísticos presentados en la Figura 1. En todos los casos, los valores de los parámetros de control del algoritmo han sido de 0.5 para el corte alfa, 5 para el tamaño de la estrella, 5 para el soporte absoluto mínimo y 0.05 para el soporte relativo mínimo.

En cuanto al número de reglas generadas, las tres configuraciones muestran un resultado parecido aunque la versión v3 suele dar un número de reglas algo mayor. En los tres casos el número de reglas resulta significativamente menor que los obtenidos por el resto de algoritmos estudiados salvo RIPPER y C4.5. En general, el algoritmo RIPPER genera un número de reglas muy pequeño a costa de un porcentaje de error mayor que el del resto de algoritmos. Con respecto a C4.5, el número de reglas parece menor en conjuntos de datos con muchos atributos y más alto en conjuntos de datos con menos atributos.

En cuanto al error promedio sobre los conjuntos de test, los resultados para FuzzyCN2 resultan mejores que el del resto de los algoritmos estudiados, particularmente para la configuración v3. En este último caso y a pesar de que el número de conjuntos de datos estudiado no es demasiado grande, el test no paramétrico de Wilcoxon (Wilcoxon's Signed Rank Test) permite afirmar que el error promedio del algoritmo FuzzyCN2 es menor que el del resto de los algoritmos estudiados con un nivel de confianza del 95%.

Agradecimientos

Este trabajo ha sido financiado parcialmente por los Proyectos de Investigación TEC2008-04920/TEC y TIN2008-06681-C06-06 de la CICYT y los Proyectos de Excelencia P07-TIC-03179 y P08-TIC-03674 de la Junta de Andalucía.

Referencias

- [1] A. Asuncion, D.J. Newman,. UCI Machine Learning Repository [http://www.ics.uci.edu/~mllearn/MLRepository.html]. University of California, School of Information and Computer Science.
- [2] K.J. Cios, L.M. Sztandera. Continuous ID3 algorithm with fuzzy entropy measures. *Proc. IEEE Int. Conference on Fuzzy Systems*, Pág. 469-476, 1992.
- [3] P. Clark, R. Boswell. Rule induction with CN2: some recent improvements. *Proc. 5th European Working Session on Learning (EWSL-91)*, Pág. 151-163, 1991.
- [4] P. Clark, T. Niblett. The CN2 Induction Algorithm. *Machine Learning*, 3, Pág. 261-283, 1989.
- [5] R.S. Michalski, I. Mozetic, J. Hong, N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proc. 5th National Conference on Artificial Intelligence (AAAI86)*, Pág. 1041-1047, 1986.
- [6] F.J. Moreno-Velo, I. Baturone, S. Sánchez-Solano, A. Barriga. Nuevos algoritmos de clasificación integrados en Xfuzzy. *Actas del XIV Congreso Español de Tecnologías y Lógica Fuzzy (ESTYLF 2008)*, Pág. 277-284, 2008.
- [7] D. Nauck, R. Kruse. NEFCLASS - A Neuro-Fuzzy Approach for the Classification of Data. *Proc. 1995 ACM Symposium on Applied Computing*, Pág. 461-465, 1995.
- [8] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1, Pág. 81-106, 1986.
- [9] J.R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4, Pág. 77-90, 1996.
- [10] R. Senhadji, S. Sánchez-solano, A. Barriga, I. Baturone, F.J. Moreno-Velo. NORFREA: An algorithm for non-redundant fuzzy rule extraction. *Proc. IEEE International Conference on Systems, Man and Cybernetics*, Pág. 604-608, 2002.
- [11] T. Tani, M. Sakoda, K. Tanaka. Fuzzy Modeling by ID3 algorithm and its application to prediction, *Proc. IEEE International Conference on Fuzzy Systems*, Pág. 923-930, 1992.
- [12] M. Umano, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, J. Kinoshita. Fuzzy Decisión Trees by Fuzzy ID3 Algorithm and Its Application to Diagnosis Systems. *Proc. IEEE International Conference on Fuzzy Systems*, Pág. 2113-2118, 1994.
- [13] L.Wang, J.M. Mendel. Generation of Rules by Learning from Examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22, Pág. 1414-1427, 1992.
- [14] R. Weber. Fuzzy ID3: a class of methods for automatic knowledge acquisition. *Proc. 2nd International Conference on Fuzzy Logic and Neural Networks*, Pág. 265-268, 1992.
- [15] I.H. Witten, E. Frank. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
- [16] J. van Zyl, I. Cloete. An inductive algorithm for learning conjunctive fuzzy rules. *Proc. 3rd IEEE International Conference on Machine Learning and Cybernetics*, Pág. 4181-4187, 2004.

REFLEXIVIDAD Y T-TRANSITIVIDAD APROXIMADAS

Dionís Boixader J. Recasens

Universitat Politècnica de Catalunya.
E.T.S. Arquitectura del Vallès C. Pere Serra 1-15
Sant Cugat del Vallès – 08190
{dionis.boixader,j.recasens}@upc.edu

Resumen

Los preórdenes y las equivalencias son probablemente las relaciones más estudiadas dentro de la Lógica Difusa. Sin embargo, su definición generalmente aceptada es crisp, ya que dada una relación, sólo puede cumplir, o no cumplir, con los axiomas de preorden o equivalencia difusos. En este artículo proponemos una fuzzificación de dichos conceptos basada en dos procedimientos: primero, a base de introducir grados en el cumplimiento de los axiomas y, segundo, comparando relaciones por medio de una similitud. El estudio posterior muestra que ambas vías están estrechamente relacionadas.

Palabras Clave: preorden difuso, equivalencia difusa, t -norma, implicación residuada, reflexividad, simetría, T -transitividad.

1 INTRODUCCIÓN

Los preórdenes y las equivalencias difusas son, probablemente, las relaciones que han merecido más atención por parte de los investigadores en el campo de la Lógica Difusa.

Aunque aparecen bajo una gran variedad de nombres en la literatura, rige un consenso casi general sobre cómo sus homólogos crisp deben generalizarse a contexto fuzzy, a saber, a través de la fuzzificación de las propiedades de reflexividad, simetría y transitividad que definen estas relaciones en la matemática tradicional.

Dicha fuzzificación apareció por primera vez en Zadeh [7] y desde entonces pocas modificaciones ha sufrido, a excepción del uso de una t -norma general en la

modelización de la transitividad difusa, una función restringida originalmente a $T=\text{MIN}$.

Definición 1.1: Una operación T del intervalo unidad $[0,1]$ es una t -norma si es asociativa, conmutativa, monótona y satisface las condiciones de contorno $T(x,0)=T(0,x)=0$ y $T(x,1)=T(1,x)=x$ para todo x de X .

Estrechamente relacionadas con las t -normas aparecen las cuasi-inversas o implicaciones residuadas:

Definición 1.2: Dada una t -norma T , su cuasi-inversa es la operación del intervalo unidad definida mediante:

$$\bar{T}(x|y) = \sup\{a \in [0,1] / T(a,x) \leq y\} \text{ para todos } x, y \text{ de } [0,1].$$

Simetrizando la cuasi-inversa de una t -norma se obtiene una nueva operación E_T .

Definición 1.3: Dada una t -norma T , la *biimplicación* o *equivalencia natural* asociada se define como:

$$\bar{T} = \min\{\bar{T}(x|y), \bar{T}(y|x)\} \text{ para todos } x, y \text{ de } [0,1].$$

En el marco de las relaciones difusas, ambas operaciones se interpretan como relaciones en el intervalo unidad, y se utilizan las notaciones I_T y E_T para la implicación residuada y la equivalencia natural, respectivamente.

Definición 1.4: Una relación difusa R en un conjunto X , o lo que es lo mismo, una aplicación $R: X \times X \rightarrow [0,1]$ es:

Reflexiva, si $R(x,x)=1$ para todo x de X .

Simétrica, si $R(x,y)=R(y,x)$ para todos x, y de X .

T-Transitiva, si $T(R(x,y), R(y,z)) \leq R(x,z)$ para todos x, y, z de X .

Definición 1.5: Un *Preorden Difuso* es una relación difusa reflexiva y T -transitiva.

El ejemplo fundamental de preorden difuso lo proporciona la propia implicación residuada I_T , mientras

que E_T constituye el ejemplo básico de equivalencia difusa.

Definición 1.6: Una *Equivalencia Difusa* es una relación difusa reflexiva, simétrica y T -transitiva.

Otras denominaciones para las equivalencias difusas son *Similitud*, *Semejanza*, *Relación de Parecido*, *T-indistinguibilidad* y *Igualdad Difusa*. Todos estos nombres sugieren que las equivalencias difusas proporcionan una medida de hasta qué punto dos objetos pueden verse como próximos o intercambiables en el marco de una teoría.

Resulta claro que una relación difusa dada R puede satisfacer, o no satisfacer, cada una de las propiedades listadas más arriba. Y que esto se produce de una forma estrictamente crisp.

Así, el conjunto de las relaciones difusas en X se divide entre las que son reflexivas y las que no lo son. Entre las que son simétricas, y las que no lo son, etcétera. Y, por conjunción de dichas propiedades, entre las que son preórdenes difusos, o equivalencias difusas, y las que no.

Resulta imposible comparar dos relaciones R y S y afirmar, por ejemplo, que R se comporta más como un preorden difuso que S , o que S es ligeramente más simétrica que R . Ser un preorden, o ser simétrica, no es en modo alguno una cuestión cuantificable o graduable.

De esta forma, grado y vaguedad quedan totalmente excluidos de la propia teoría. Una teoría que, no lo perdamos de vista, está orientada precisamente a modelizar los conceptos vagos y graduales por naturaleza. Objeciones de este estilo, así como vías de solución basadas en la gradación de los axiomas pueden encontrarse en [1,3,5].

Más allá de las consideraciones teóricas que se quieran invocar, existen también aspectos prácticos que justifican la necesidad de ampliar la clase de las relaciones estudiadas con nuevos elementos que no sean, en sentido estricto, preórdenes o equivalencias difusas.

Allá donde la información provenga de la cuantificación subjetiva de un agente expresando la proximidad entre objetos, no es razonable esperar que los datos se comporten de forma estrictamente T -transitiva, por ejemplo. O también en aquellos casos en que la información esté vinculada a medidas experimentales, y por tanto sujeta a errores de naturaleza probabilística o sistemática inherentes al proceso de medida. Sin embargo, cabe esperar que las relaciones difusas obtenidas por dichos procedimientos se comporten, aproximadamente,

como preórdenes o equivalencias difusas, si ésta es la hipótesis subyacente al modelo teórico.

Con la finalidad de poder asumir que cualquier relación difusa es, hasta cierto punto, un preorden o una equivalencia difusa, proponemos dos caminos distintos.

De una parte, se trata de redefinir los axiomas de reflexividad, simetría y T -transitividad a base de introducir grados de satisfacción en las definiciones originales. Esta vía respondería probablemente a las motivaciones de orden teórico que se han expuesto.

Por otro lado, y dado que los errores experimentales o subjetivos deforman en cierta medida el conjunto de valores de la relación en su conjunto, propondremos también medidas de semejanza entre relaciones. La idea es considerar una relación S más o menos como un preorden o una equivalencia difusa, de acuerdo con la medida de semejanza entre S y R , donde R es un preorden o equivalencia difusa en el sentido tradicional. En este sentido, hablaremos de *candidatos* (la relación S) y de *prototipos* o *referencias* (la relación R).

Quizás, la parte más interesante de la investigación que presentamos se encuentre en que ambos tratamientos del problema están, de hecho, muy relacionados.

2 INTRODUCCIÓN DE GRADOS EN LOS AXIOMAS

Empecemos por la modificación de los axiomas, o propiedades definitorias de las relaciones estudiadas.

Definición 2.1: Una relación difusa R en X se dirá que es:

a-Reflexiva si $R(x, x) \geq a$, para cierto $a \in [0,1]$ y para todo x de X .

a-Simétrica respecto a una t -norma T si $T(a, R(x, y)) \leq R(y, x)$, para cierto $a \in [0,1]$ y para todos x, y de X .

a-Transitiva respecto a una t -norma T , o *a-T-transitiva*, si $T(a, T(R(x, y), R(y, z))) \leq R(x, z)$, para cierto $a \in [0,1]$ y para todos x, y, z de X .

Dos simples consecuencias se desprenden de la definición anterior. La primera, que para $a=1$ se recuperan reflexividad, simetría y T -transitividad en el sentido habitual. La segunda, que toda relación a -reflexiva, a -simétrica o a - T -transitiva será también b -reflexiva, b -

simétrica o b - T -transitiva para todo $b \leq a$. Estas observaciones dan sentido a la próxima definición.

Definición 2.2: Dada una t -norma T y una relación difusa R en un conjunto X :

Su grado de reflexividad es $a_r = \inf_{x \in X} R(x, x)$

Su grado de simetría respecto a T es

$$a_s = \inf_{x, y \in X} \left\{ \sup \{a \in [0, 1] \mid T(a, R(x, y)) \leq R(y, x)\} \right\}$$

Su grado de transitividad con respecto a T es

$$a_t = \inf_{x, y \in X} \left\{ \sup \{a \in [0, 1] \mid T(a, T(R(x, y), R(y, z))) \leq R(x, z)\} \right\}$$

De esta manera, una relación difusa R es un preorden difuso si, y sólo si $a_r = a_t = 1$, y es una equivalencia difusa si, y sólo si $a_r = a_s = a_t = 1$.

Los anteriores niveles se pueden expresar utilizando la casi-inversa $\vec{T}$, o implicación residuada. El grado de simetría de R respecto a T es

$$a_s = \inf_{x, y \in X} \left\{ \vec{T}(R(x, y) \mid R(y, x)) \right\}$$

mientras que el grado de transitividad es

$$a_t = \inf_{x, y \in X} \left\{ \vec{T}(T(R(x, y), R(y, z)) \mid R(x, z)) \right\}$$

Una posible alternativa consiste en reescribir el nivel de simetría a_s mediante la equivalencia natural del intervalo unidad E_T como sigue:

$$a_s = \inf_{x, y \in X} E_T(R(x, y), R(y, x))$$

Definición 2.3: Una relación difusa R es un a -Preorden si sus niveles de reflexividad y transitividad son ambos mayores que a .

Definición 2.4: Una relación difusa R es una a -Equivalencia si sus niveles de reflexividad, simetría y transitividad son todos mayores que a .

3 RELACIONES DE REFERENCIA

A continuación, procederemos a comparar relaciones difusas con la intención de transferir grados de reflexividad, simetría y transitividad entre ellas.

Dos relaciones difusas R y S definidas sobre un mismo conjunto X pueden compararse de muchas maneras. En este trabajo nos restringiremos a la proximidad o semejanza punto a punto. Nos referiremos a esta particular manera de evaluar la semejanza entre relaciones

como a su *equivalencia natural*, dado que extiende de forma obvia la Definición 1.4 para el intervalo unidad.

Definición 3.1: Dadas dos relaciones difusas sobre X , la *equivalencia natural* entre R y S se define como:

$$E_T(R, S) = \inf_{x, y \in X} E_T(R(x, y), S(x, y))$$

Por esta misma razón seguiremos usando la notación E_T . De hecho, se trata únicamente de una extensión punto a punto, o mejor dicho tratándose de relaciones, par a par.

Obsérvese que E_T define una equivalencia difusa en el conjunto de todas las relaciones difusas sobre X , estrictamente en el sentido que marca la Definición 1.6, según se prueba en [5]. Por lo tanto, E_T constituye una medida de proximidad entre R y S , o lo que es lo mismo, de hasta qué punto ambas relaciones son intercambiables entre sí.

Para transferir propiedades de R (relación de referencia) a S (relación candidato), otro concepto importante es el *orden puntual*.

Definición 3.2: Una relación difusa R es menor o igual que otra relación difusa S punto a punto, $R \leq_x S$, si $R(x, y) \leq S(x, y)$ para cualesquiera x e y de X .

En este caso, es habitual decir que R es *más específica* que S .

En adelante, supondremos que la relación de referencia es un preorden o una equivalencia difusos, mientras que S , el candidato, puede ser cualquier relación, con la única condición de que sea similar a R cuando ambas se comparen mediante E_T .

Proposición 3.3: Sean R y S dos relaciones difusas sobre X . Si R es reflexiva y $E_T(R, S) \geq a$ entonces S es a -reflexiva.

Este resultado se obtiene como consecuencia de los dos lemas siguientes.

Lema 3.4: Sean R y S relaciones difusas sobre X tales que $S \leq_x R$. Si R es reflexiva y $E_T(R, S) \geq a$ entonces S es a -reflexiva.

Para probar la Proposición 3.3 podemos dividir S en dos relaciones $S_1 = \min\{S, R\}$ y $S_2 = \max\{S, R\}$ tales que $S_1 \leq_x R$ y $R \leq_x S_2$. Los grados de reflexividad de S_1 y S_2 , a y 1 respectivamente, se agregarán mediante la t -norma T y así se obtiene el grado de reflexividad estimado para S , que vale $T(a, 1) = a$.

Proposición 3.5: Sean R y S dos relaciones difusas sobre X . Si R es simétrica y $E_T(R, S) \geq a$ entonces S es b -simétrica para $b = T(a, a)$.

Como antes, el nivel estimado de simetría de S puede ser mejorado si se supone que las relaciones que intervienen están ordenadas puntualmente. La Proposición 3.5 es también en este caso una consecuencia natural del lema siguiente.

Lema 3.6: Sean R y S relaciones difusas en X tales que $R \leq_x S$ o $S \leq_x R$. Si R es simétrica y $E_T(R, S) \geq a$ entonces S es a -simétrica.

Descomponiendo S en S_1 y S_2 de la forma anterior, se obtiene que $b = T(a, a)$ es el grado de simetría estimado para S .

Finalmente, tenemos también un resultado parecido para la transitividad.

Proposición 3.7: Sean R y S dos relaciones difusas en X . Si R es T -transitiva y $E_T(R, S) \geq a$ entonces S es b - T -transitiva para $b = T(T(a, a), a)$.

También aquí el orden puntual de las relaciones permite mejorar la estimación sobre el grado de transitividad del candidato S , a la vez que proporciona una demostración de la Proposición 3.7.

Lema 3.8: Sean R y S relaciones difusas en X tales que $R \leq_x S$. Si R es T -transitiva y $E_T(R, S) \geq a$ entonces S es b - T -transitiva con $b = T(a, a)$.

Lema 3.9: Sean R y S relaciones difusas en X tales que $S \leq_x R$. Si R es T -transitiva y $E_T(R, S) \geq a$ entonces S es a - T -transitiva.

Resumiendo, y puesto que la transitividad establece la cota más baja entre las tres propiedades consideradas, cuando R es un preorden o una equivalencia difusos, se tiene:

Proposición 3.10: Si R es un preorden o una equivalencia difusos sobre X , y S es una relación difusa tal que $E_T(R, S) \geq a$, entonces S será un b -preorden o una b -equivalencia con $b = T(T(a, a), a)$.

4 RELACIÓN ENTRE TRANSITIVIDAD APROXIMADA Y REFLEXIVIDAD.

Aunque en general transitividad y reflexividad se presentan como axiomas independientes, no lo son del todo. Cierta grado de reflexividad está garantizado cuando una relación es T -transitiva.

En efecto, si R es T -transitiva a cada elemento x de X se le puede asignar un valor a_x que sea una cota inferior del grado de reflexividad de la relación en x , $R(x, x)$. Tal valor a_x se calcula mediante la *clausura transitiva* (véase, por ejemplo, [6]), que en el caso que nos ocupa se escribirá como $a_x = \sup_{u \in X} T(R(x, u), R(u, x))$

Aunque cada cota inferior a_x esté relacionada con un elemento x específico, se puede calcular una cota inferior conjunta de la manera habitual.

Proposición 4.1: Toda relación T -transitiva R en X es también b -reflexiva, con $b = \inf_{x \in X} a_x$.

Este resultado se generaliza sin dificultades al caso en que la relación R sea tan solo a - T -transitiva.

Proposición 4.2: Toda relación a - T -transitiva R en X es también b -reflexiva, con $b = \inf_{x \in X} a_x$, donde cada a_x se calcula como $a_x = \sup_{u \in X} T(T(R(x, u), R(u, x)), a)$.

Y si el grado de transitividad determina una cota inferior para el de reflexividad, éste último determina una cota superior para el primero.

Proposición 4.3: Sea R una relación b -reflexiva sobre X . Entonces el grado de transitividad de R es menor o igual que $c = \bar{T}(\inf_{x \in X} a_x | b)$

Nótese que los niveles de reflexividad por encima de a_x no tienen efecto alguno sobre los de transitividad.

Proposición 4.4: Sea R una relación a - T -transitiva sobre X . Si S es otra relación tal que $S(x, y) = R(x, y)$ para todos $x \neq y$, pero $S(x, x) \geq R(x, x)$, entonces S es también a - T -transitiva.

Así que, vista la particular relación que existe entre transitividad y reflexividad aproximadas, por un lado, y la completa independencia que ambas propiedades presentan respecto a la simetría, parece razonable afirmar que la reflexividad es una propiedad superflua en el contexto que nos ocupa. Un punto de vista parecido puede encontrarse en [4].

5 BÚSQUEDA DE RELACIONES DE REFERENCIA PRÓXIMAS.

Vamos a abordar ahora el problema de encontrar, para una relación dada S , una relación de referencia R parecida. Supondremos que S satisface hasta cierto grado algunos de los axiomas en el sentido que establece la Definición 2.2. Como consecuencia, calcularemos cotas para los grados de similitud entre S y alguna relación de referencia R que satisfaga los mismos axiomas que S , pero en el sentido habitual (Definición 1.4). De hecho, veremos que a mayor grado de satisfacción de los axiomas por parte de S , más próxima se encuentra la relación de referencia R .

Lema 5.1: Dada una relación difusa S en X , la relación $S_a(x, y) = T(a, S(x, y))$ (T una t -norma) satisface:

$$5.1.1 \quad S_a \leq_x S$$

$$5.1.2 \quad E_T(S_a, S) \geq a$$

Lema 5.2: Si S es una relación reflexiva y b -transitiva en X , entonces $S_a(x, y) = T(a, S(x, y))$ (T una t -norma) satisface:

$$5.2.1 \quad S_a \text{ es } c\text{-}T\text{-transitiva con } c = \bar{T}(b | a).$$

$$5.2.2 \quad S_a \text{ es } a\text{-reflexiva.}$$

Como consecuencia, $c = 1$ y por tanto S_a es T -transitiva cuando $b \leq a$, lo que prueba el corolario siguiente.

Proposición 5.3: Dado un a -preorden S en X , la relación difusa $S_a(x, y) = T(a, S(x, y))$ (T una t -norma) es b -reflexiva con $b = T(a, a)$, es T -transitiva, satisface que $S_a \leq_x S$ y que $E_T(S_a, S) \geq a$.

Por lo tanto $R = S_a$ es una relación de referencia para S , y su proximidad viene determinada esencialmente por el grado de transitividad de la relación candidato S .

Además, teniendo en cuenta la Proposición 4.4 se puede transformar S_a en una nueva relación reflexiva R , incrementando los valores de la diagonal:

$$R(x, y) = S_a(x, y) \text{ para todos } x \neq y, y$$

$$R(x, x) = 1 \text{ para todo } x.$$

Una tal relación R constituye un preorden de referencia para S . Sin embargo, ni $R \leq_x S$ ni $S \leq_x R$, y además $E_T(R, S) \geq T(a, a)$, es decir, que R se aleja de S .

Volviendo a la Proposición 5.3, lo que allí se ha encontrado es una aproximación inferior para una relación dada S . ¿Existe alguna aproximación superior? Una posible opción es tomar la clausura transitiva de S . La clausura transitiva de una relación se define como la menor de todas las relaciones transitivas que son mayores

o iguales que ella según el orden puntual. Así pues, si R' denota la clausura transitiva de S , se tiene que $S \leq_x R'$. Sin embargo, es difícil establecer una cota para $E_T(R', S)$ y, de momento, este problema permanece abierto.

Finalmente, consideremos también la simetría. El procedimiento estándar para simetrizar una relación dada S consiste en tomar el mínimo entre los valores $S(x, y)$ y $S(y, x)$ para cada par (x, y) .

Lema 5.4: Dada una relación a -simétrica S sobre X , la relación difusa $R_{SYM}(x, y) = \min\{S(x, y), S(y, x)\}$ cumple:

$$5.4.1 \quad R_{SYM} \text{ es simétrica.}$$

$$5.4.2 \quad R_{SYM} \leq_x S$$

$$5.4.3 \quad E_T(R_{SYM}, S) \geq a$$

Por lo tanto, R_{SYM} proporciona una relación de referencia para S en lo que a simetría se refiere. Además, el procedimiento preserva tanto la reflexividad como la T -transitividad.

Lema 5.5: Para toda relación S a -reflexiva, la relación $R_{SYM}(x, y) = \min\{S(x, y), S(y, x)\}$ es también a -reflexiva.

Lema 5.6: Para toda relación S a - T -transitiva, la relación $R_{SYM}(x, y) = \min\{S(x, y), S(y, x)\}$ es también a - T -transitiva

Proposición 5.7: Dada una a -equivalencia S sobre X , la relación difusa $S_a(x, y) = T(a, R_{SYM}(x, y))$ es a -reflexiva, simétrica y T -transitiva.

Como antes, se puede transformar S_a incrementando su diagonal y obtener así una relación reflexiva:

$$R(x, y) = S_a(x, y) \text{ para todos } x \neq y, y$$

$$R(x, x) = 1 \text{ para todo } x.$$

También en este caso, aunque R constituye una relación de referencia para S , se pierde el orden puntual y la cota de proximidad empeora convirtiéndose en

$$E_T(R, S) \geq T(T(a, a), a).$$

6 CONCLUSIONES.

Se ha mostrado cómo el enfoque tradicional en la fuzzificación de preórdenes y equivalencias puede relajarse, de manera que incluya un mayor espectro de relaciones.

De hecho, cualquier relación difusa puede ser considerada como un preorden o una equivalencia, hasta cierto punto.

El grado con que pueda serlo se ha introducido de dos maneras distintas. Primero, mediante la relajación de los axiomas de reflexividad, simetría y transitividad. Después, comparando la semejanza o proximidad entre la relación arbitraria S y una relación de referencia R . Finalmente, se han establecido relaciones entre ambos puntos de vista, que resultan ser esencialmente equivalentes.

Referencias

- [1] L. A. Zadeh, "Similarity relations and fuzzy orderings," *Information Sciences*, 3, 1971, pp. 177–200.
- [2] G. J. Klir and Bo Yuan, *Fuzzy sets and fuzzy logic. Theory and applications*. New Jersey: Prentice Hall, 1995, ch. 3.
- [3] D. Boixader, J. Jacas and J. Recasens, "Fuzzy equivalence relations: advanced material" in *Fundamentals of Fuzzy Sets*, D. Dubois and H. Prade eds, Kluwer 2000, pp.261-290.
- [4] S. Gottwald, "Fuzzified fuzzy relations. *Proc. of the 4th IFSA congress*, vol. Mathematics, Brussels 1991, pp.82-86.
- [5] J. Jacas and J. Recasens, "Fuzzified properties of fuzzy relations" *Proc. IPMU 10th International Conference*, 2002, pp.157–161.
- [6] L. Behoumek, "Extensionality in graded properties of fuzzy relations" *Proc. IPMU Eleventh 10th International Conference*, vol.II, 2006, pp.1604–1611.
- [7] U. Hohle, "Fuzzy sets and sheaves. Part 1: basic concepts." *Fuzzy Sets and Systems* 158, 2007, pp.1143-1174.

ON THE RELATION BETWEEN FUZZY PREORDERS, FUZZY CLOSING MORPHOLOGICAL OPERATORS AND FUZZY CONSEQUENCE OPERATORS

J. Elorza¹ R. Fuentes-González² J. Bragard¹ P. Burillo²

¹ Dept. of Physics and Applied Mathematics. University of Navarra, {jelorza,jbragard}@unav.es

² Dept. of Automation and Computer Science. Public University of Navarra, {rfuentes,pburillo}@unavarra.es

Abstract

In a previous paper [10] we explored the notion of coherent fuzzy consequence operator. It is well-known that the operator induced by a fuzzy preorder through Zadeh's compositional rule is always a coherent fuzzy consequence operator. It is also known that the relation induced by a fuzzy consequence operator is a fuzzy preorder if such operator is coherent [7]. Fuzzy closing operators of mathematical morphology can be considered as fuzzy consequence operators. In [12] we showed that they are coherent operators. The aim of this paper is to analyze the relations between both classes of operators and the class of all fuzzy preorders in order to translate well known properties from Approximate Reasoning to the one of Image Processing.

Keywords: Fuzzy Preorder, Fuzzy Consequence Operator, Fuzzy Morphological Closing Operator, Coherent Operators, Approximate Reasoning, Image Processing.

1 INTRODUCTION

Consequence Operators in classical logic were introduced by A. Tarski in 1930 [20]. According to Tarski, a logic is just a set of propositions with a consequence operator. The relationship between consequence operators and preorders is well known [5].

Concepts of Fuzzy Preorder and Fuzzy Consequence Operator (FCO for short) are essential on fuzzy logic. These notions have been defined as a natural generalization of the classical ones:

Given a non-empty universal set X which will represent a set of propositions and a t-norm $*$, a fuzzy (binary) relation R on X (fuzzy subset of $X \times X$) is called a fuzzy $*$ -preorder if it satisfies:

$$(R1) \ R(x, x) = 1 \ \forall x \in X \quad (\text{reflexivity})$$

$$(R2) \ R(x, z) \geq R(x, y) * R(y, z) \ \forall x, y, z \in X \quad (*\text{-transitivity}).$$

If R is also symmetric, this is, $R(x, y) = R(y, x)$ for all $x, y \in X$ then it is called a fuzzy $*$ -similarity, fuzzy $*$ -indistinguishability or, equivalently, fuzzy $*$ -equivalence. If R is only reflexive and symmetric, we will say that it is a fuzzy tolerance.

Fix a complete lattice L which will be the range of the memberships of the fuzzy subsets of X , J. Pavelka introduced in 1979 the concept of FCO on X in fuzzy logic [18] extending the concept of consequence operator in Tarski's sense in a natural way.

A function $C : L^X \longrightarrow L^X$ is a FCO on X if it satisfies:

$$(C1) \ \mu \subset C(\mu) \text{ for all fuzzy subset } \mu \in L^X \quad (\text{inclusion})$$

$$(C2) \ \mu_1 \subset \mu_2 \implies C(\mu_1) \subset C(\mu_2) \quad \text{for all } \mu_1, \mu_2 \in L^X \quad (\text{monotony})$$

$$(C3) \ C(C(\mu)) \subset C(\mu) \text{ for all } \mu \in L^X \quad (\text{idempotence})$$

Notice that, under the inclusion axiom, (C3) may be written equivalently as (C3') $C(C(\mu)) = C(\mu) \ \forall \mu \in L^X$.

These operators, also called closure operators, have been studied extensively [1],[2],[8],[12],[13],[24]. During the last and current decades, these operators have been also studied in the context of the fuzzy logic taking the chain $L = [0, 1]$ as a special case [6],[7],[10],[11],[15].

In 1991, J.L. Castro and E. Trillas proved [7] the following result: If R is a fuzzy $*$ -preorder then the

operator C_R^* between fuzzy subsets of X given by the max-* Zadeh's compositional rule

$$C_R^*(\mu) = \mu \circ^* R \quad (1)$$

is a fuzzy consequence operator (induced by R), where

$$(\mu \circ^* R)(x) = \sup_{w \in X} \{\mu(w) * R(w, x)\} \quad (2)$$

We proved [10] that the converse of the previous result is also true: R is a *-preorder if and only if C_R^* is a FCO. Thus the only possibility to obtain FCOs from relations using Zadeh's max-* compositional rule is to work with *-preorders.

For the inverse process (inducing fuzzy preorders from FCOs), J.L. Castro and E. Trillas added the coherence axiom to the FCO concept [6]. A FCO C is called *-coherent if

$$\mu(a) * C(\varphi_a)(x) \leq C(\mu)(x) \quad (3)$$

for all $\mu \in [0, 1]^X$ and for all $(a, x) \in X \times X$, where $\varphi_a(t) = \varphi_{\{a\}}(t)$ is the crisp membership of the singleton $\{a\}$.

It is proven [7] that if C is a *-coherent FCO then the relation defined by

$$R_C(x, y) = C(\varphi_x)(y) \quad (4)$$

is a fuzzy *-preorder (induced by C). It is also proven that the operator C_R^* is *-coherent.

It is easy to extend the notion of coherence to operators $C : L^X \rightarrow L^X$ between fuzzy subsets in L^X . This is: a FCO C is called *-coherent if $\mu(a) * C(\varphi_a)(x) \leq C(\mu)(x)$ for all $\mu \in L^X$ and for all $(a, x) \in X \times X$, where $\varphi_a(t) = \begin{cases} 1 & \text{if } t = a \\ 0 & \text{if } t \neq a \end{cases}$, being in this case 1 and 0 the greatest and least element of the lattice L , respectively. Then, all the previous results hold.

On the other hand, the classical morphological operators of image processing have been introduced by G. Matheron and J. Serra in [16],[17],[21]. These operators have been studied recently, see for instance [12],[14],[23]. In [14] they are studied in a new context: obtaining relevant information in fuzzy relational systems.

In this context, the most used morphological operators are erosion, dilation, opening and closing that are given respectively by $\varepsilon_R(\mu) = R^{op} \triangleleft \mu$, $\delta_R(\mu) = R \circ \mu$, $\alpha_R(\mu) = R \circ (R^{op} \triangleleft \mu)$ and $\beta_R(\mu) = R \triangleleft (R^{op} \circ \mu)$, where R is the relation acting as structurant element

and μ is in the class of all fuzzy subsets of the universal set X or $X \times X$. R^{op} denotes the inverse relation of R , $R^{op}(x, y) = R(y, x)$ and the operators $\circ$, $\triangleleft$ are defined as usual:

$$(R \circ \mu)(x) = \sup_{w \in X} \{R(x, w) * \mu(w)\}, \quad (R \triangleleft \mu)(x) = \inf_{w \in X} \{R(x, w) \rightsquigarrow \mu(w)\} \text{ for some t-norm } * \text{ and for some implication operator } \rightsquigarrow.$$

Notice that the operator $\circ$ and possibly the operators $\triangleleft$ and $\rightsquigarrow$ depend on the t-norm $*$. However, when there is no confusion, in order to alleviate the notation we omit the symbol $*$.

From now on, $*$ will be a left-continuous t-norm, that is, $\alpha * (\sup M) = \sup(\alpha * M)$ for all $\alpha \in L$, $M \subset L$ and $\rightsquigarrow_*$ will denote the residuated implication induced by the t-norm $*$: $\alpha \rightsquigarrow_* \beta = \sup\{w \in L / \alpha * w \leq \beta\}$. It is also called the pseudoinverse application of $*$.

Both in the crisp and in the fuzzy cases, opening and closing operators with an isotropic structuring element are used in image processing to eliminate specific image details smaller than the structuring element. The global shape of the objects is not distorted.

In particular, a closing operator connects objects that are close to each other, fills up small holes and smooths the object outline by filling up narrow gulfs. Meanings of *near*, *small* and *narrow* are related to the size and the shape of the structuring element.

Figure 1 ([22]) shows an example where a set is transformed by a closing operator through a disk as structuring element.

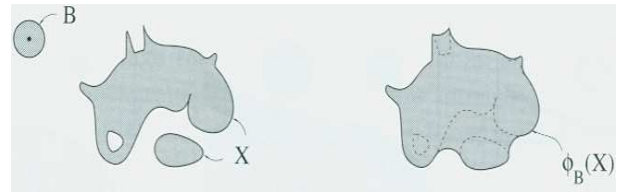


Figure 1: Effect of a closing operator on a non-smooth object

In this way, the concept of fuzzy closing operator φ_R^* induced by a fuzzy relation is relevant in fuzzy mathematical morphology, defined in [12] as follows. For any fuzzy relation R on X ,

$$\varphi_R^*(\mu) = R^{op} \triangleleft_{\rightsquigarrow_*} (R \circ^* \mu) \quad (5)$$

where $R \circ^* \mu$ is given by

$$(R \circ^* \mu)(x) = \sup_{w \in X} \{R(x, w) * \mu(w)\} \text{ and}$$

$$(R \triangleleft_{\rightsquigarrow_*} \mu)(x) = \inf_{w \in X} \{R(x, w) \rightsquigarrow_* \mu(w)\}.$$

It is well-known that φ_R^* is a closure operator, that is, a fuzzy consequence operator.

In Section 2, we recall some properties which are proven in [12].

In Section 3, we show some results about the relationship between fuzzy preorders and their induced consequence and closing morphological fuzzy operators.

2 FUZZY CLOSING MORPHOLOGICAL OPERATORS AS FUZZY CONSEQUENCE OPERATORS

In [12] we proved that, for all t-norm $*$, the fuzzy closing operator induced by a relation is always a $*$ -coherent operator acting as a fuzzy consequence operator. More precisely, we proved the following result.

Theorem 1.- ([12]) *Let R be any fuzzy relation on X . Then the fuzzy closing operator $\varphi_R^* : L^X \rightarrow L^X$ induced by R by means of (5) is a $*$ -coherent fuzzy consequence operator.*

Since the relation induced by a coherent fuzzy consequence operator is a preorder, it immediately follows that the relation induced by the fuzzy closing operator φ_R^* is a preorder:

Corollary 1.- ([12]) *Let R be a fuzzy relation on X . Then the fuzzy relation induced by the operator φ_R^* by means of (4) is a $*$ -preorder.*

We also proved the following characterization of such induced relation from which other properties can be easily deduced.

Lemma 1.- ([12]) *Let R be any fuzzy relation on X and put $C = \varphi_R^*$. Then the $*$ -preorder induced by C by means of (4), R_C is such that*

$$R_C = (R^{op} \triangleleft_{\rightsquigarrow_*} R)^{op} \quad (6)$$

Recall that the $\triangleleft_{\rightsquigarrow_*}$ composition for two relations R and S on X is defined by $(R \triangleleft_{\rightsquigarrow_*} S)(x, y) = \inf_{w \in X} \{R(x, w) \rightsquigarrow_* S(w, y)\}$.

From the previous lemma it follows immediately that $(R^{op} \triangleleft_{\rightsquigarrow_*} R)^{op}$ is always a $*$ -transitive relation, then $R^{op} \triangleleft_{\rightsquigarrow_*} R$ is also $*$ -transitive. This result about the

transitivity of the relation R does not involve the composition operator $\circ$, even though transitivity is defined using composition.

The following characterization about the transitivity of a relation R does not involve composition either.

Theorem 2.- ([12]) *Let R be any fuzzy relation on X . Then R is $*$ -transitive if and only if $R \leq R^{op} \triangleleft_{\rightsquigarrow_*} R$.*

Accordingly, we have the following characterization of fuzzy preorders without the operator $\circ$:

Corollary 2.- ([12]) *Let R be any fuzzy relation on X . Then R is an $*$ -preorder if and only if $R = R^{op} \triangleleft_{\rightsquigarrow_*} R$, this is, R $*$ -preorder if and only if $R_C = R^{op}$.*

3 FUZZY PREORDERS AND THEIR INDUCED CONSEQUENCE AND CLOSING MORPHOLOGICAL FUZZY OPERATORS

From now on, we make use of the following notation introduced in [11].

Γ^* will represent the class of all fuzzy preorders, Ω will represent the class of all fuzzy consequence operators, $\tilde{\Omega}^*$ the subclass of all $*$ -coherent fuzzy consequence operators and Ω_p^* the subclass of all fuzzy consequence operators induced by a $*$ -preorder by means of (1), this is, $\Omega_p^* = \{C \in \Omega \mid \exists R \in \Gamma^*, C = C_R^*\}$.

$\theta^* : \Gamma^* \rightarrow \Omega$ is the function such that for each $*$ -preorder R , $\theta^*(R) = C_R^*$ given by means of (1) and $\tilde{\theta} : \tilde{\Omega}^* \rightarrow \Gamma^*$ is the function such that for each $*$ -coherent fuzzy consequence operator C , $\tilde{\theta}(C) = R_C$ given by means of (4). Notice that function $\tilde{\theta}$ does not depend of the t-norm $*$.

Here are some elementary properties of these families and functions.

It is easy to prove that θ^* is one to one. Moreover for each fuzzy relation R , $R_{C_R^*}$ is exactly the relation R , in other words, $\tilde{\theta} \circ \theta^*$ is the identity mapping on Γ^* .

In consequence, if $C \in \Omega_p^*$, there exists a unique $R \in \Gamma^*$ such that $\theta^*(R) = C$, then $\tilde{\theta} \circ \theta^*(R) = R$ and $\theta^* \circ \tilde{\theta}(C) = (\theta^* \circ \tilde{\theta}) \circ \theta^*(R) = \theta^* \circ (\tilde{\theta} \circ \theta^*)(R) = \theta^*(R) = C$. Conversely, if $\theta^* \circ \tilde{\theta}(C) = C$, C is the image of the fuzzy relation $\tilde{\theta}(C)$ by the mapping θ^* and $C \in \Omega_p^*$. Therefore, $\Omega_p^* = \{C \in \Omega \mid \theta^* \circ \tilde{\theta}(C) = C\}$.

Now we will prove the following lemma in order to ob-

tain a characterization of $*$ -preorders which involves fuzzy closing morphological operators that are induced by a $*$ -preorder as fuzzy consequence operators in Approximate Reasoning by means of (1).

Lemma 2.- *Let C be any $*$ -coherent operator (satisfying (3), not necessarily fuzzy consequence operator). Then $\theta^* \circ \tilde{\theta}(C) \subset C$.*

Proof.- If C satisfies (3):

$$\begin{aligned}\theta^* \circ \tilde{\theta}(C)(\mu)(x) &= C_{\tilde{\theta}(C)}^*(\mu)(x) = \\ \sup_{w \in X} \{ \mu(w) * \tilde{\theta}(C)(w, x) \} &= \\ \sup_{w \in X} \{ \mu(w) * R_C(w, x) \} &= \\ \sup_{w \in X} \{ \mu(w) * C(\varphi_w)(x) \} &\leq \\ \sup_{w \in X} \{ C(\mu)(x) \} &= C(\mu)(x)\end{aligned}$$

Therefore $\theta^* \circ \tilde{\theta}(C) \subset C$. $\square$

Theorem 3.- *Let R be any fuzzy relation. If R is a $*$ -preorder then $\varphi_R^* \in \Omega_p^*$, this is, φ_R^* is induced by a $*$ -preorder by means of (1).*

Proof.- Assume that R is $*$ -preorder and prove that $\theta^* \circ \tilde{\theta}(\varphi_R^*) = \varphi_R^*$. From Theorem 1, φ_R^* is a $*$ -coherent operator, then $\theta^* \circ \tilde{\theta}(\varphi_R^*) \subset \varphi_R^*$ by Lemma 2.

On the other hand,

$$\begin{aligned}\varphi_R^*(\mu)(x) &= \inf_{w \in X} \{ R(w, x) \rightsquigarrow_* C_{R^{op}}^*(\mu)(w) \} \leq \\ R(x, x) \rightsquigarrow_* C_{R^{op}}^*(\mu)(x) &= 1 \rightsquigarrow_* C_{R^{op}}^*(\mu)(x) = \\ C_{R^{op}}^*(\mu)(x)\end{aligned}$$

From Corollary 2, if R $*$ -preorder then

$$R_C = \tilde{\theta}(\varphi_R^*) = R^{op}$$

Thus

$$\begin{aligned}\varphi_R^* \subset C_{R^{op}}^* &= \theta^*(R^{op}) = \\ \theta^*(\tilde{\theta}(\varphi_R^*)) &= \theta^* \circ \tilde{\theta}(\varphi_R^*)\end{aligned} \quad \square$$

Notice that if R is a $*$ -preorder then $\varphi_R^* \in \Omega_p^*$ and there exists a $*$ -preorder S such that $\varphi_R^* = C_S^*$. Thus

$$\tilde{\theta}(\varphi_R^*) = \tilde{\theta}(C_S^*) = \tilde{\theta} \circ \theta^*(S) = S$$

and

$$\theta^* \circ \tilde{\theta}(\varphi_R^*) = \theta^*(S)$$

As θ^* is one to one

$$\tilde{\theta}(\varphi_R^*) = S$$

that is

$$R_C = S$$

From Corollary 2, $R_C = R^{op}$ and $S = R^{op}$.

From this, the following corollary is straightforward.

Corollary 3.- *Let R be any fuzzy relation. If R is a $*$ -preorder then $\varphi_R^* = C_{R^{op}}^*$.*

Notice that, for an equivalence relation R , it is enough to consider in Corollary 3, $R^{op} = R$ by symmetry to obtain the following result.

Corollary 4.- *Let R be any fuzzy relation. If R is a $*$ -equivalence then $\varphi_R^* = C_R^*$.*

Finally, we show two properties for relations which are not transitive.

If R is only reflexive:

Theorem 4.- *Let R be any fuzzy relation. If R is reflexive then $\varphi_R^* \leq C_{R^{op}}^*$.*

Proof.- If R is reflexive then

$$\begin{aligned}\varphi_R^*(\mu)(x) &= \inf_{t \in X} \{ R(t, x) \rightsquigarrow_* \sup_{w \in X} \{ R(t, w) * \mu(w) \} \} \leq \\ R(x, x) \rightsquigarrow_* \sup_{w \in X} \{ R(x, w) * \mu(w) \}\end{aligned}$$

By reflexivity

$$\begin{aligned}\varphi_R^*(\mu)(x) &\leq 1 \rightsquigarrow_* \sup_{w \in X} \{ R(x, w) * \mu(w) \} = \\ \sup_{w \in X} \{ \mu(w) * R(x, w) \} &= C_{R^{op}}^*(\mu)(x)\end{aligned} \quad \square$$

If R is a tolerance:

Corollary 5.- *Let R be any fuzzy relation. If R is reflexive and symmetric then $\varphi_R^* \leq C_R^*$.*

4 CONCLUSIONS AND FUTURE WORKS

In [12], we have connected the class of fuzzy consequence operators in the context of approximate reasoning with the class of fuzzy closing operators used in image processing acting as filters.

We do not know of any previous relationship between both classes in the literature apart that both are closure operators. Nevertheless, an interesting relationship between several fuzzy mathematical morphologies, approximate reasoning and spatial reasoning is established [3],[4].

In this paper, we have proved some properties that connects both structures with the class of all fuzzy preorders.

More precisely, in Theorem 3 we have shown that if R is a preorder then the fuzzy closing morphological operator induced by R by means of (5) is also induced by R as fuzzy consequence operator by means of (1). In fact, we have proved that if R is $*$ preorder then $\varphi_R^* = C_{R^{op}}^*$.

From this, we have shown in Corollary 4 that for equivalence relations, the induced fuzzy closing morphological operator coincides with induced fuzzy consequence operator, this is, $\varphi_R^* = C_R^*$.

In future works we will use the connection established between both classes of operators and the class of preorders to translate well known properties from the field of approximate reasoning to the one of fuzzy relational systems. In particular we will explore its applications to image processing.

Acknowledgements

This research is partially supported by MICINN (Spain) under the project FIS2008-06335-C02-02 and Gobierno de Navarra (Spain) under the project 2008/2301.

We thank both anonymous referees for their valuable suggestions.

References

- [1] R. Bělohlávek. Fuzzy closure operators. *Journal of Mathematical Analysis and Appl.* Vol 262, pp. 473-489, 2001.
- [2] R. Bělohlávek. Fuzzy closure operators II. *Soft Computing* Vol 7. No 1, pp. 53-64, 2002.
- [3] I. Bloch. Fuzzy and Bipolar Mathematical Morphology, Applications in Spatial Reasoning. *Lecture Notes in Computer Science* Vol 5590, pp. 1-13. Springer. 2009
- [4] I. Bloch, H. Maître. Fuzzy Mathematical Morphologies: a comparative study. *Pattern Recognition* Vol 28. No 9, pp. 1341-1387, 1995.
- [5] J.L. Castro. Fuzzy logics as families of bivaluated logics. *Fuzzy sets and systems* 64, pp. 321-332, 1994.
- [6] J.L. Castro, M. Delgado, E. Trillas. Inducing Implication Relations. *International Journal of Approximate Reasoning*. Vol.10, pp. pp. 235-250, 1994.
- [7] J.L. Castro, E. Trillas. Tarski's fuzzy consequences. *Proc. of the Internat. Fuzzy Engineering Symp '91*. Vol.1, pp. 70-81, 1991.
- [8] D. Dikranjan, W. Tholen. Categorical Structure of Closure Operators. *Kluwer Academic Publishers*. 1995
- [9] A. Di Nola, A. Lettieri. Relation equations in residuated lattices. *Rendiconti del Circolo matematico di Palermo*, s.II, XXXVIII, pp. 246-256, 1989.
- [10] J. Elorza, P. Burillo. On the relation between fuzzy preorders and fuzzy consequence operators. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. Vol.7. No 3, pp. 219-234, 1999.
- [11] J. Elorza, P. Burillo. Connecting fuzzy preorders, fuzzy consequence operators and fuzzy closure and co-closure systems. *Fuzzy sets and systems*. Vol.139. No 3, pp. 601-613, 2003.
- [12] J. Elorza, R. Fuentes-González, J. Bragard, P. Burillo. Fuzzy Closing Operators and their coherence as Fuzzy Consequence Operators *Proc. of the 14th Spanish Conference on Fuzzy Logic and Technologies (ESTYLF 2008)*, pp. 221-227, 2008.
- [13] F. Esteva, P. García, L. Godo, R.O Rodríguez. Fuzzy Approximation Relations, Modal Structures and Possibilistic Logic. *Mathware & Soft Computing*, Vol. V, n 2-3, pp. 151-166, 1998.
- [14] R. Fuentes-González, P. Burillo, N. Frago. Técnicas de Morfología Matemática en el tratamiento de Sistemas Relacionales Difusos *Actas del congreso Estylf 2002 sobre tecnologías y lógica fuzzy*, pp. 321-326, 2002.
- [15] J. Jacas, J. Recasens. Preórdenes Duales, Operadores de Clausura y Contextos Borrosos *Actas del congreso Estylf 2004 sobre tecnologías y lógica fuzzy*, pp. 125-128, 2004.
- [16] G. Matheron. Eléments pour une théorie des milieux poreux. *Masson*. Paris. 1967
- [17] G. Matheron. Random Sets and Integral Geometry. *Wiley*. NY. 1975
- [18] J. Pavelka. On Fuzzy Logic I. *Zeitschr. f. Math. Logik und Grundlagen d. Math.* Bd.25, pp. 45-52, 1979.
- [19] K. Peeva, Y. Kyosev. Algorithm for solving max-product fuzzy relational equations. *Soft Computing* Vol 11. No 7, pp. 593-605, 2007.

- [20] A. Tarski. Logic, semantics, metamathematics. Clarendon Press, Oxford, 1956.
- [21] J. Serra. Image Analysis and Mathematical Morphology. *Academic Press*. Vol I (fourth printing, 1990) and Vol II (second printing, 1992).
- [22] P. Soille. Morphological Image Analysis. Principles and Applications. *Springer*. 2nd edition, 2003.
- [23] M.E. Valle, P. Sussner. A general framework for fuzzy morphological associative memories. *Fuzzy sets and systems*. Vol.159. No 7, pp. 747-768, 2008.
- [24] M. Ward. The closure operators of a lattice. *Annals of Mathematics*. Vol.43. N.2, pp. 191-196, 1940.

AN EXPLICIT EXPRESSION FOR k -NEGATIONS

E. de Amo¹ J. Fernández-Sánchez

University of Almería, Spain

¹edeamo@ual.es

Abstract

We give a new proof for the existence and the uniqueness of k -negations. With the aid of a generalised dyadic representation system is possible to give an explicit expression for k -negations and it is possible to showing that there exists a set of measure 1 where k -negations have derivatives with zero values.

Keywords: Negation, k -negation, generalised dyadic system, singular function

1 Introduction

Negation functions are very well known objects from Set Theory and Fuzzy Logic: they are mappings $n : [0, 1] \rightarrow [0, 1]$ satisfying a) $n(0) = 1$, $n(1) = 0$; b) n is not increasing. In case n satisfies c) $n^2(x) = x$, we say that n is a strong negation.

In a classic paper by Trillas [14], we can find the following characterization for strong negations:

Theorem 1 *The function n is a strong negation if, and only if, there exists an increasing bijection $f : [0, 1] \rightarrow [0, 1]$, verifying $n(x) = f^{-1}(1 - f(x))$.*

A particular case of great interest is that of negations which are related with the duality of aggregation functions and the idempotency of some binary functions. They are called k -negations and are denoted by n_k . In fact, they are strong negations satisfying this additional property (for $k \in]0, 1[$):

$$n_k(kx) = k + (1 - k)n_k(x),$$

and they have been studied in [5], [6], and [7], among others. The main fact is that for each k , there exists one, and only one, n_k .

The solution for the system of functional equations

$$\begin{cases} h(\frac{x}{2}) = kh(x) \\ h(\frac{1+x}{2}) = k + (1 - k)h(x) \end{cases}$$

with x in $[0, 1]$, is a function which inverse satisfies the relation of Trillas for k -negations. It has been deeply studied as we can see in [2], [3], [4], [9], [10], [11], [12], and [13]. We will denote it with the capital letter R .

The function R is (for $k \neq 1/2$) singular in the sense of Measure Theory (i.e., it is a continuous and monotone function having null derivatives on a set of measure 1).

The relation

$$n_k(x) = R(1 - R^{-1}(x))$$

does not yields to an explicit expression for $n_k(x)$. In this sense Fraile et al. [5], gave a recurrence expression on a dense and denumerable subset of $[0, 1]$ which is generalized in [8].

The target of this paper is double: on the one hand, we give a explicit expression for k -negations, and, on the other hand, we show this new property for n_k : it has null derivatives on a set of measure 1 (i.e., the same property that was true for R), and moreover, this is the only value that the derivatives can get.

The main tools we will use to obtain these results are a generalised dyadic system developed by the authors in [1], and the system of functional equations characterizing k -negations.

2 The existence of k -negations

Lemma 2 *The system of functional equations*

$$\begin{cases} h(kx) = kh(x) \\ h(k + (1 - k)x) = k + (1 - k)h(x) \end{cases}$$

has one, and only one, solution in the set $\mathcal{B}([0, 1], \mathbb{R})$ of the real bounded functions defined on the unit interval; and it is the identity map.

Proof. Clearly, the identity map satisfies the system of equations.

For short, let us denote $\mathcal{B}([0, 1])$ the Banach space of real bounded functions defined on the unit interval equipped with the sup-norm, and the functional operator introduced by the formula

$$F : \mathcal{B}([0, 1]) \longrightarrow \mathcal{B}([0, 1]); \quad g \rightarrow F(g),$$

where $F(g)$ acts in the following way:

$$F(g)(y) := \begin{cases} kg\left(\frac{y}{k}\right), & 0 \leq y \leq k \\ k + (1-k)g\left(\frac{y-k}{1-k}\right), & k < y \leq 1. \end{cases}$$

Hence, F is contractive of ratio $b := \max\{k, 1-k\}$. For this, let us denote $h_i := F(g_i)$ for given g_1 and g_2 in $\mathcal{B}([0, 1])$. Easy computations show that

$$h_2(y) - h_1(y) = k \left[g_2\left(\frac{y}{k}\right) - g_1\left(\frac{y}{k}\right) \right] \text{ if } 0 \leq y \leq k, \text{ and} \\ h_2(y) - h_1(y) = (1-k) \left[g_2\left(\frac{y-k}{1-k}\right) - g_1\left(\frac{y-k}{1-k}\right) \right] \text{ if } k < y \leq 1,$$

$$\text{implies } \|h_2 - h_1\| \leq b \|g_2 - g_1\|.$$

The contractive mapping theorem of Banach gives that there exist one and only one fix point for F : the desired uniqueness of h is just fulfilled, and the proof is finished. ■

Lemma 3 *One element h in $\mathcal{B}([0, 1])$ is a k -negation if, and only if, it satisfies the equations*

$$\begin{cases} h(kx) = k + (1-k)h(x) \\ h(k + (1-k)x) = kh(x) \end{cases} \quad (a)$$

Proof. A k -negation satisfies

$$\begin{cases} n_k(kx) = k + (1-k)n_k(x) \\ n_k^2(x) = x \end{cases}.$$

Because $n_k(n_k(kx)) = kx$, we can write

$$n_k(k + (1-k)n_k(x)) = kx; \text{ and doing } n_k(x) = y \text{ we have } n_k(k + (1-k)y) = kn_k^{-1}(y).$$

But $n_k^{-1} = n_k$, and hence, we obtain the second of the equations in (a).

For the reverse implication it is enough to show that $n_k^2(x) = x$. From (a), we obtain

$$\begin{cases} n_k^2(kx) = n_k(k + (1-k)n_k(x)) = kn_k^2(x) \\ n_k^2(k + (1-k)x) = n_k(kn_k(x)) = k + (1-k)n_k^2(x) \end{cases}$$

These equalities and the lemma above ensure that $n_k^2(x) = x$. Moreover, one can check that n_k is decreasing. ■

3 An explicit expression for k -negations

3.1 Generalised dyadic number system

In this section, we refer to [1, §.3] for a wider study in detail.

The authors introduce a new representation system for numbers in $]0, 1[$ via series expansion combining the numbers k and $1-k$ ($k \in]0, 1[$); precisely, in the form $x = \sum_{n=1}^{+\infty} (1-k)^n k^{m_n}$. This situation is unique, very similar to that of dyadic expansions, but a denumerable set of numbers for which there are exactly two representations: one finite and the other infinite. They are obtained from a dynamical system which is ergodic and λ -preserving measure.

Definition 4 *Let $k \in]0, 1[$. For each $x \in]0, 1[$, there exists a non negative integer n_0 such that*

$$k^{n_0+1} \leq x \leq k^{n_0}.$$

Hence, $x = k^{n_0+1} + y_1$, with $0 \leq y_1 \leq k^{n_0}(1-k)$; and we can write

$$x = k^{n_0+1} + k^{n_0}(1-k)x_1,$$

where $x_1 \in]0, 1[$. Reasoning on x_1 , we obtain

$$x = k^{n_0+1} + k^{n_0+n_1+1}(1-k) + k^{n_0+n_1}(1-k)^2 x_2;$$

and, by induction, we have this formal equality:

$$x = \sum_{d=0}^{+\infty} (1-k)^d k^{1+\sum_{j=0}^d n_j},$$

which will be called the generalized dyadic representation for the corresponding number x .

These series converge to x , because

$$\left| x - \sum_{d=0}^m (1-k)^d k^{1+\sum_{j=0}^d n_j} \right| < (1-k)^{m+1},$$

and the majorization M-test of Weierstrass is applied. We summarize this situation:

Proposition 5 *Let $k \in]0, 1[$. If $x \in]0, 1[$, then there exists an increasing sequence of naturals $1 \leq m_0 \leq m_1 \leq \dots \leq m_d \leq \dots$, such that $x = \sum_{d=0}^{+\infty} (1-k)^d k^{m_d}$.*

Proposition 6 *The expansion in the above proposition is unique but it would be finite or stationary (i.e., $m_d = m_j$ if $d \geq j$).*

Proof. Because

$$1 = k + k(1-k) + k(1-k)^2 + k(1-k)^3 + \dots,$$

in the finite or stationary cases double expansions appear:

$$\sum_{d=0}^n (1-k)^d k^{m_d} = \sum_{d=0}^{n-1} (1-k)^d k^{m_d} + \sum_{d=n}^{+\infty} (1-k)^d k^{m_{d+1}}.$$

By the other hand, if the sequence (m_d) is not bounded (the expansion for x is not finite neither stationary), let us consider infinite expansions for x and y :

$$\begin{aligned} x &= \sum_{d=0}^{+\infty} (1-k)^d k^{m_d} \\ y &= \sum_{d=0}^{+\infty} (1-k)^d k^{m'_d} \end{aligned}$$

where $0 \leq d \leq n-1$, implies $m_d = m'_d$, and if $d \geq n$ then $m_d < m'_d$. Hence:

$$\begin{aligned} y &\leq \sum_{d=0}^{n-1} (1-k)^d k^{m_d} + k^{m'_n} \sum_{d=n}^{+\infty} (1-k)^d = \\ &= \sum_{d=0}^{n-1} (1-k)^d k^{m_d} + k^{m'_n-1} (1-k)^n < \\ &< \sum_{d=0}^{n-1} (1-k)^d k^{m_d} + \sum_{d=n}^{+\infty} (1-k)^d k^{m_d} = x. \end{aligned}$$

As a consequence, for x with non bounded (m_d) , x differs from y being stationary or non stationary. ■

3.2 A singular function with the generalised dyadic system

Definition 7 For each $k \in]0, 1[\setminus \{\frac{1}{2}\}$ let us define the function $f_k : [0, 1] \rightarrow [0, 1]$ given in the following way: each x with non stationary infinite expansion (i.e., there exist $1 \leq t_0 < t_1 < \dots < t_d < \dots$, such that)

$$\begin{aligned} x &= k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \\ &\quad + k^{t_1} (1-k)^{s_0+1} + \dots + k^{t_1} (1-k)^{s_1} + \dots \\ &\quad + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d} + \dots \end{aligned}$$

is mapped to

$$\begin{aligned} f_k(x) &:= k + k(1-k) + \dots + k(1-k)^{t_0-2} + \\ &\quad + k^{s_0+2} (1-k)^{t_0-1} + \dots + k^{s_0+2} (1-k)^{t_1-2} + \\ &\quad + k^{s_1+2} (1-k)^{t_1-1} + \dots + k^{s_1+2} (1-k)^{t_2-2} + \dots \\ &\quad + k^{s_{d-1}+2} (1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2} (1-k)^{t_d-2} + \dots \end{aligned}$$

If $t_0 := 1$, then $k + k(1-k) + \dots + k(1-k)^{t_0-2}$ does not exist. In the stationary case, i.e., when x has finite expansion

$$\begin{aligned} x &= k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \dots \\ &\quad + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d}, \end{aligned}$$

then

$$\begin{aligned} f_k(x) &: = k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots \\ &\quad + k^{s_{d-1}+2} (1-k)^{t_{d-1}-1} + \dots \\ &\quad + k^{s_{d-1}+2} (1-k)^{t_d-2} + k^{s_d+1} (1-k)^{t_d-1}. \end{aligned}$$

Remark 8 If we work with x given by an infinite number of chains in the form

$$k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d},$$

the image will be an infinite sum of chains in the form

$$k^{s_{d-1}+2} (1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2} (1-k)^{t_d-2}.$$

If the number of chains is finite, the result is analogous, but adding

$$k^{s_d+1} (1-k)^{t_d-1}.$$

First, we give some properties concerning f_k .

Proposition 9 The function f_k is a strictly decreasing function.

Proof. Let us consider four different cases.

a. Let be $y < x$, where

$$\begin{aligned} x &= k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \\ &\quad + k^{t_1} (1-k)^{s_0+1} + \dots + k^{t_1} (1-k)^{s_1} + \dots \\ &\quad + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d} + \\ &\quad + k^{t_{d+1}} (1-k)^{s_d+1} + \dots + k^{t_{d+1}} (1-k)^{s'_{d+1}} + \dots \\ &\quad + k^{t_{d+1}} (1-k)^{s_{d+1}} + \dots \end{aligned}$$

and

$$\begin{aligned} y &= k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \\ &\quad + k^{t_1} (1-k)^{s_0+1} + \dots + k^{t_1} (1-k)^{s_1} + \dots \\ &\quad + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d} + \\ &\quad + k^{t_{d+1}} (1-k)^{s_d+1} + \dots + k^{t_{d+1}} (1-k)^{s'_{d+1}} + \\ &\quad + k^{t'_{d+2}} (1-k)^{s'_{d+1}+1} + \dots \end{aligned}$$

(note that $s'_{d+1} < s_{d+1}$), thereof

$$\begin{aligned} f_k(x) &:= k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots \\ &\quad + k^{s_{d-1}+2} (1-k)^{t_{d-1}-1} + \dots \end{aligned}$$

$$\begin{aligned}
& +k^{s_{d-1}+2}(1-k)^{t_{d-2}} + k^{s_d+2}(1-k)^{t_{d-1}} + \dots \\
& +k^{s_d+2}(1-k)^{t_{d+1}-2} + k^{s_{d+1}+2}(1-k)^{t_{d+1}-1} + \dots
\end{aligned}$$

and

$$\begin{aligned}
f_k(y) &:= k + k(1-k) + \dots \\
& +k(1-k)^{t_0-2} + \dots + k^{s_{d-1}+2}(1-k)^{t_{d-1}-1} + \dots \\
& +k^{s_{d-1}+2}(1-k)^{t_{d-2}} + k^{s_d+2}(1-k)^{t_{d-1}} + \dots \\
& +k^{s_d+2}(1-k)^{t_{d+1}-2} + k^{s'_{d+1}+2}(1-k)^{t_{d+1}-1} + \dots;
\end{aligned}$$

which yields to $f_k(x) < f_k(y)$.

b. Now, let us consider x and y , $y < x$, in the form:

$$\begin{aligned}
x &= k^{t_0} + \dots + k^{t_0}(1-k)^{s_0} + \\
& +k^{t_1}(1-k)^{s_0+1} + \dots + k^{t_1}(1-k)^{s_1} + \dots \\
& +k^{t_d}(1-k)^{s_{d-1}+1} + \dots \\
& +k^{t_d}(1-k)^{s_d} + \dots + k^{t_{d+1}}(1-k)^{s_{d+1}} + \dots
\end{aligned}$$

and

$$\begin{aligned}
y &= k^{t_0} + \dots + k^{t_0}(1-k)^{s_0} + \\
& +k^{t_1}(1-k)^{s_0+1} + \dots + k^{t_1}(1-k)^{s_1} + \dots \\
& +k^{t_d}(1-k)^{s_{d-1}+1} + \dots + k^{t'_{d+1}}(1-k)^{s_{d+1}} + \dots \\
& +k^{t_{d+1}}(1-k)^{s'_{d+1}+1} + k^{t'_{d+1}}(1-k)^{s'_{d+1}+1} + \dots
\end{aligned}$$

(Note here that $t_{d+1} < t'_{d+1}$.) Hence,

$$\begin{aligned}
f_k(x) &:= k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots \\
& +k^{s_{d-1}+2}(1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2}(1-k)^{t_{d-2}} + \\
& +k^{s_d+2}(1-k)^{t_{d-1}} + \dots + k^{s_d+2}(1-k)^{t_{d+1}-2} + \dots
\end{aligned}$$

and

$$\begin{aligned}
f_k(y) &:= k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots \\
& +k^{s_{d-1}+2}(1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2}(1-k)^{t_{d-2}} + \\
& +k^{s_d+2}(1-k)^{t_{d-1}} + \dots + k^{s_d+2}(1-k)^{t'_{d+1}-2} + \dots;
\end{aligned}$$

which, again, implies $f_k(x) < f_k(y)$.

c. x has finite expansion and y has infinite expansion, with $x < y$.

d. x and y have both finite expansion.

Cases *c.* and *d.* fulfil the rest of possibilities and the proofs run as in *a.* and *b.* ■

Proposition 10 *The function f_k is a continuous function.*

Proof. Let $x \in [0, 1]$, and we consider the following two cases:

a. x has not finite expansion:

$$\begin{aligned}
x &= k^{t_0} + \dots + k^{t_0}(1-k)^{s_0} + \dots \\
& +k^{t_d}(1-k)^{s_{d-1}+1} + \dots + k^{t_d}(1-k)^{s_d} + \dots
\end{aligned}$$

If a sequence (x_n) converges to x , then there exists m such that $n \geq m$ implies

$$\begin{aligned}
x_n &= k^{t_0} + \dots + k^{t_0}(1-k)^{s_0} + \dots \\
& +k^{t_d}(1-k)^{s_{d-1}+1} + \dots + k^{t_d}(1-k)^{s_d} + \\
& +k^{t'_{d+1}}(1-k)^{s_{d+1}} + \dots + k^{t'_{d+1}}(1-k)^{s'_{d+1}+1} + \dots,
\end{aligned}$$

i.e., x_n and x coincide on the first d blocks if $n \geq m$. Then $f(x)$ and $f(x_n)$ have expansions that coincide in the form

$$\begin{aligned}
& k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots \\
& +k^{s_{d-1}+2}(1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2}(1-k)^{t_{d-2}} + \\
& +k^{s_d+2}(1-k)^{t_{d-1}} + \dots
\end{aligned}$$

which implies that

$$\begin{aligned}
|f_k(x) - f_k(x_n)| &\leq 2k^{s_d+2} \left[(1-k)^{t_{d-1}} + (1-k)^{t_d} + \dots \right] = \\
& = 2k^{s_d+1}(1-k)^{t_{d-1}}.
\end{aligned}$$

But, if $n \rightarrow +\infty$ then, d, t_d and $s_d \rightarrow +\infty$. Hence, $f_k(x_n) \rightarrow f_k(x)$.

b. If x has finite expansion, then we will consider sequences (x_n) one side converging to x . Firstly, let $x_n \searrow x$. Because

$$\begin{aligned}
x &= k^{t_0} + \dots + k^{t_0}(1-k)^{s_0} + \dots \\
& +k^{t_d}(1-k)^{s_{d-1}+1} + \dots + k^{t_d}(1-k)^{s_d},
\end{aligned}$$

then

$$\begin{aligned}
x_n &= k^{t_0} + \dots + k^{t_0}(1-k)^{s_0} + \dots + k^{t_d}(1-k)^{s_{d-1}+1} + \dots \\
& +k^{t_d}(1-k)^{s_d} + k^{t'_{d+1}}(1-k)^{s_{d+1}} + \dots
\end{aligned}$$

(where $t'_{d+j} = t'_{d+j}(n)$ for all j 's and $t'_{d+j} \rightarrow +\infty$ if n does). Applying f_k on them:

$$\begin{aligned}
f_k(x) &= k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots \\
& +k^{s_{d-1}+2}(1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2}(1-k)^{t_{d-2}} + \\
& +k^{s_d+2}(1-k)^{t_{d-1}};
\end{aligned}$$

and

$$\begin{aligned}
f_k(x_n) &= k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots \\
& +k^{s_{d-1}+2}(1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2}(1-k)^{t_{d-2}} + \\
& +k^{s_d+2}(1-k)^{t_{d-1}} + \dots + k^{s_d+2}(1-k)^{t'_{d+1}-2} + \dots
\end{aligned}$$

But, if $t'_{d+j} \rightarrow +\infty$, then $k^{s_d+2}(1-k)^{t_{d-1}} + \dots + k^{s_d+2}(1-k)^{t'_{d+1}-2} + \dots = k^{s_d+1}(1-k)^{t_{d-1}}$; i.e., $f_k(x_n) \rightarrow f_k(x)$.

By the other hand, if $x_n \nearrow x$, things run analogously.

Both cases *a.* and *b.* together give the considerable convergence $f_k(x_n) \rightarrow f_k(x)$, if $n \rightarrow +\infty$. ■

The next lemma will be useful in that follows.

Lemma 11 ([1, Th.13]) *The set of points $x = \sum_{n=1}^{+\infty} (1-k)^n k^{m_n}$ for which $\lim_n \frac{m_n}{n} = \frac{k}{1-k}$ is a set of λ -measure 1.*

Definition 12 *A point x is said normal in the basis k (or k -normal) if it verifies the property in the lemma above.*

Theorem 13 *There exists a set of measure 1 where the function f_k has null derivatives on each point.*

Proof. Let us consider x a normal number in this representation system. It will necessarily have an infinite expansion:

$$x = k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \dots + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d} + \dots;$$

and we define sequences (x_n) and (y_n) :

$$x_n = k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \dots + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d},$$

and

$$y_n = k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \dots + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d} + k^{t_d} (1-k)^{s_{d+1}},$$

verifying $x_n \leq x \leq y_n$ for all $n \in \mathbb{N}$. The function f_k acts respectively as follows on them:

$$f_k(x_n) = k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots + k^{s_{d-1}+2} (1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2} (1-k)^{t_d-2} + k^{s_d+1} (1-k)^{t_d-1};$$

and

$$f_k(y_n) = k + k(1-k) + \dots + k(1-k)^{t_0-2} + \dots + k^{s_{d-1}+2} (1-k)^{t_{d-1}-1} + \dots + k^{s_{d-1}+2} (1-k)^{t_d-2} + k^{s_d+2} (1-k)^{t_d-1}.$$

Then, we have

$$\begin{aligned} \frac{f_k(y_n) - f_k(x_n)}{y_n - x_n} &= \frac{k^{s_d+2} (1-k)^{t_d-1} - k^{s_d+2} (1-k)^{t_d-1}}{k^{t_d} (1-k)^{s_d+1}} \\ &= \frac{-k^{s_d+1} (1-k)^{t_d}}{k^{t_d} (1-k)^{s_d+1}} = - \left(\frac{k}{1-k} \right)^{\frac{1-2k}{1-k} s_d + o(s_d)}, \end{aligned}$$

where we have taken into account that $t_d \simeq \frac{k}{1-k} s_d$ for normal numbers.

Now, on the one hand, if $k > 1/2$, then $\frac{1-2k}{1-k} < 0$, and

$$\lim \frac{f_k(y_n) - f_k(x_n)}{y_n - x_n} = 0.$$

On the other hand, if $k < 1/2$, then $\frac{1-2k}{1-k} > 0$, and, analogously,

$$\lim \frac{f_k(y_n) - f_k(x_n)}{y_n - x_n} = 0.$$

Hence, the existence of $f'_k(x)$ implies it is null.

But monotonicity of f means its derivability on a set of measure 1 (see [11]). Hence, $f'_k(x) = 0$ on a set of measure 1. ■

Proposition 14 *The function f_k does not admit non-zero derivatives.*

Proof. Let us consider x with finite expansion:

$$x = k^{t_0} + \dots + k^{t_d} (1-k)^{s_d},$$

and

$$x_n = k^{t_0} + \dots + k^{t_d} (1-k)^{s_d} + k^n (1-k)^{s_{d+1}}.$$

For each n :

$$\begin{aligned} \frac{f_k(x_n) - f_k(x)}{x_n - x} &= \frac{-k^{s_{d+1}} (1-k)^m}{k^m (1-k)^{s_{d+1}}} \\ &= - \left(\frac{k}{1-k} \right)^{s_{d+1}-m}, \end{aligned}$$

and when $n \rightarrow +\infty$, this sequence goes to 0 or to $-\infty$, depending on the value of $\frac{k}{1-k}$. Hence, in case the limit existing, it must be zero.

On the other hand, if x has not finite expansion, then let us consider sequences (x_n) and (y_n) , as in the above theorem. In addition, let

$$\begin{aligned} y'_n &= k^{t_0} + \dots + k^{t_0} (1-k)^{s_0} + \dots + k^{t_d} (1-k)^{s_{d-1}+1} + \dots + k^{t_d} (1-k)^{s_d} + \\ &\quad + k^{t_d} (1-k)^{s_{d+1}} + k^{t_d} (1-k)^{s_{d+2}}. \end{aligned}$$

For these numbers:

$$\frac{f_k(y'_d) - f_k(x_d)}{y'_d - x_d} = - \left(\frac{k}{1-k} \right)^{-s_{d+1}-t_d},$$

and

$$\frac{f_k(y'_d) - f_k(x_d)}{y'_d - x_d} = - \frac{1+k}{2-k} \left(\frac{k}{1-k} \right)^{-s_{d+1}-t_d}.$$

In case there exists derivative, the quotient of both limits $-\frac{1+k}{2-k}$ must be equal to 1; but this is not the case when $k \neq 1/2$. Hence, because the derivative exists, it must be equal to 0. ■

Theorem 15 *The function f_k is the unique solution for the system of functional equations given by*

$$\begin{cases} f(kx) = k + (1-k)f(x) \\ f(k + (1-k)x) = kf(x) \end{cases}$$

Proof. Firstly, we will show that f_k verifies this system of functional equations. Let us consider the case when x has infinite expansion; the finite case runs analogously. If

$$x = k^{t_0} + \dots + k^{t_0}(1-k)^{s_0} + k^{t_1}(1-k)^{s_0+1} + \dots + k^{t_d}(1-k)^{s_d} + \dots,$$

then

$$\begin{aligned} f_k(x) &:= k + k(1-k) + \dots + k(1-k)^{t_0-2} + \\ &+ k^{s_0+2}(1-k)^{t_0-1} + \dots + k^{s_0+2}(1-k)^{t_1-2} + \\ &+ k^{s_1+2}(1-k)^{t_1-1} + \dots + k^{s_1+2}(1-k)^{t_2-2} + \dots \end{aligned}$$

For these formulae, writing kx and $f_k(kx)$, the first of the two equation holds.

For the second equation the reasoning is that follows: if $t_0 \neq 1$, then

$$k + (1-k)x = k + k^{t_0}(1-k) + \dots + k^{t_0}(1-k)^{s_0+1} + k^{t_1}(1-k)^{s_0+2} + \dots + k^{t_1}(1-k)^{s_1+1} + \dots$$

and

$$\begin{aligned} f_k(k + (1-k)x) &= k^2 + k^2(1-k) + \dots \\ &+ k^2(1-k)^{t_0-2} + k^{s_0+3}(1-k)^{t_0-1} + \dots \\ &+ k^{s_0+3}(1-k)^{t_1-2} + \dots \end{aligned}$$

and if $t_0 = 1$, then

$$f_k(x) = k^{s_0+2}(1-k)^{t_0-1} + \dots + k^{s_0+2}(1-k)^{t_1-2} + \dots$$

and

$$\begin{aligned} f_k(k + (1-k)x) &= k^{s_0+3}(1-k)^{t_0-1} + \dots \\ &+ k^{s_0+3}(1-k)^{t_1-2} + \dots \end{aligned}$$

For both cases, the second of the equations is fulfilled. ■

Corollary 16 *The functions n_k and f_k are, in fact, the same function.*

In [2], the authors show two important properties concerning k -negations. Here λ denotes de Lebesgue measure on the reals.

Corollary 17 *The k -negation n_k maps a set of λ -measure 0 on a set of λ -measure 1. The initial set has Hausdorff dimension equal to $\frac{\ln[k^k(1-k)^{1-k}]}{\ln[k^{1-k}(1-k)^k]}$.*

Corollary 18 *The k -negation n_k maps a set of λ -measure 1 on a set of λ -measure 0 with fractal dimension equal to $\frac{\ln[k^k(1-k)^{1-k}]}{\ln[k^{1-k}(1-k)^k]}$.*

References

- [1] E. de Amo, and J. Fernández-Sánchez: *A Generalised Dyadic Representation System. Int. J. Pure Appl. Math.* **52** (1) (2009) 49-66
- [2] E. de Amo, and J. Fernández-Sánchez: *Singular Functions with Applications to Fractals and Generalised Takagi's Functions*. Submitted to press.
- [3] L. Berg, and M. Kruppel: *De Rham's singular function and related functions. Zeitschrift für Analysis und ihre Anwendungen* **19** (1) (2000) 227-237
- [4] B. Bernstein, T. Erber, and M. Karamolengos: *Sneaky plasticity and mesoscopic fractals. Chaos, Solitons & Fractals* **3** (1993) 269-277
- [5] A. Fraile, G. Mayor, and M. Monserrat: *k-negations*. Actes del IV Congrés Català de Lògica, Barcelona (1985) 67-68
- [6] P. García, and L. Valverde: *Sobre una classe de ternes de De Morgan*. Actes del IV Congrés Català de Lògica, Barcelona (1985) 73-76.
- [7] G. Mayor: *On a family of quasi-arithmetic means. Aequationes Math.* **35** (1988) 97-98
- [8] G. Mayor, and J. Torrens: *Duality for binary operations. Fuzzy Sets and Systems* **47** (1992) 77-80
- [9] J. Paradís, P. Viader, and L. Bibiloni: *Riesz-Nagy singular functions revisited. J. Math. Anal. Appl.* **329** (2007) 592-602
- [10] G. de Rham: *Sur quelques courbes definies par des equations fonctionnelles. Rend. Sere. Mat. dell'Univ, e del Politecnico di Torino* **16** (1956) 101-113
- [11] F. Riesz, and B. Sz-Nagy: *Leçons d'analyse fonctionnelle*. Gauthier-Villars, Paris, 1968
- [12] R. Salem: *On singular monotone function., Trans. Amer. Math. Soc.* **53** (1943) 427-439
- [13] L. Takacs: *An increasing continuous singular function. Amer. Math. Mon.* **85** (1978) 35-36
- [14] E. Trillas: *Sobre funciones de negación en la teoría de conjuntos difusos. Stochastica* **3** (1) (1979) 47-59

FIRST ORDER SMTL LOGIC AND QUASI-WITNESSED MODELS

Marco Cerami Francesc Esteva

Institut d'Investigació en Intel·ligència Artificial (IIIA - CSIC)
Campus U.A.B., 08193 Bellaterra, Catalunya {cerami,esteva}@iiia.csic.es

Abstract

In this paper we prove strong completeness of axiomatic extensions of First Order SMTL logic adding the so-called quasi-witnessed axioms with respect to quasi-witnessed Models. In order to achieve this result, we make use of methods that are typical of Classical Predicate Logic, and have been later generalized by P. Hájek to cope with Predicate Fuzzy Logic. At the end of the paper, we obtain, as a particular case, the result of strong completeness, already proven by M.C. Laskowski and S. Malekpour, for Product Predicate Logic with respect to quasi-witnessed Models.

Keywords: Foundations of Fuzzy Logic, Mathematical Fuzzy Logic, First Order Monoidal t-norm based logic, First Order Product Logic, witnessed and quasi-witnessed models.

1 INTRODUCTION

In recent times there has been a growing interest on First Order Fuzzy Logic due, among other facts, to its role in the construction of Description Logics (DLs) for the Semantic Web (see [1]). Since its birth in the eighties, Classical DLs enjoyed of an already existing wide amount of results in Classical First Order Logic. Also, during the last twenty years there have been attempts to develop DLs based on Fuzzy Logic, in order to cope with vague information but, in this field, the main results obtained are limited to the case of the so-called Zadeh logic¹ and, in the last time, after Hájek's paper [5], some results on FDL over Gödel

¹This is a logic whose connectives are related to max, min, the negation function $\neg x = 1 - x$ and the impli-

cation function $x \rightarrow y = \max(\neg x, y)$, all of them defined over the real unit interval. Even though this is not a residuated logic, it is definable in Łukasiewicz logic. and Łukasiewicz Logic (see [2]). A basic result on FDL over Łukasiewicz are based on an important property shared by Classical and Łukasiewicz First Order Logic: the so-called *witnessed model property* defined in [5]. In [4], generalizing the classical case, the value of an universally (existentially) quantified formula is the infimum (supremum) of the values of the results of replacing the quantified variable by the evaluation of a term of the language in $\mathbf{M}$. In the context of Classical Logic, as well as every finitely valued logic, the infimum and supremum turn out to be a minimum and a maximum, respectively, but, when we move to infinitely valued logics, we can find sets of formulas which have values whose infimum (resp. supremum) is strictly different from each element of the set, i.e., the quantified formula has no *witness*. A *witnessed model* is then a model in which each quantified formula has a witness and it is an important property because Hájek proves that it implies a limited form of finite model property for Description Logics based on Łukasiewicz logic. Moreover, he introduces the so-called witnessing axioms that are satisfied in Łukasiewicz Logic and, adding them to any axiomatic extension of $\text{MTL}\forall$, we obtain a logic complete with respect to witnessed models. The $\mathcal{ALC}$ description language over these logics extended by means of witnessing axioms, enjoys the finite model property. In particular, using such results, in [5], Hájek proves that assertion satisfiability and validity, for fuzzy $\mathcal{ALC}$ based on Łukasiewicz t-norm are decidable problems.

In [8] it is proven that Łukasiewicz First Order Logic is complete with respect to witnessed models (or, in other words, it has the witnessed model property), but also that neither Gödel, nor Product First Order Logic share this property. In fact no other first order logic of a continuous t-norm enjoys this property, since it is related to continuity of the truth functions that it is known that only Łukasiewicz logic has. However, in [9]

cation function $x \rightarrow y = \max(\neg x, y)$, all of them defined over the real unit interval. Even though this is not a residuated logic, it is definable in Łukasiewicz logic.

is proven that Product Predicate Logic enjoys a weaker property, what we call *quasi-witnessed model property*. Quasi-witnessed models² are models in which whenever the value of an universally quantified formula is strictly greater than 0, then it has a witness, while existentially quantified formulas are always witnessed. In this paper we prove, following the style of [8] that there is an axiomatic extension of SMTL that enjoys quasi-witnessed model property³; the result of [9] about the completeness of Product First Order Logic with respect to quasi-witnessed models, will follow as a corollary from our main result.

2 PRELIMINARIES

The logic SMTL is defined in the literature as the axiomatic extension of the Monoidal t-norm Logic MTL by the axiom:

$$(PC) \quad \varphi \wedge \neg\varphi \rightarrow \perp$$

At the semantic level it is the logic of left-continuous t-norms whose associated negation is the so called *Gödel negation*, i.e. the negation whose truth function, in the standard semantics, is defined by:

$$n(x) = \begin{cases} 0, & \text{if } x > 0 \\ 1, & \text{if } x = 0 \end{cases}$$

In this paper we deal with an axiomatic extension of predicate SMTL, denoted by $SMTL^{\forall qw}$, which we define below; following [4], we previously define first order SMTL logic, (denoted by $SMTL^{\forall}$).

Definition 2.1 $SMTL^{\forall}$ is the expansion of propositional SMTL adding the two "classical" quantifiers $\forall$ and $\exists$ and the following set of axiom schemata:

(P) the axioms resulting from the axioms of SMTL after the substitution of propositional variables by formulas of the new predicate language.

$$(\forall 1) \quad (\forall x)\varphi(x) \rightarrow \varphi(t), \text{ where } t \text{ is substitutable for } x \text{ in } \varphi.$$

$$(\exists 1) \quad \varphi(t) \rightarrow (\exists x)\varphi(x), \text{ where } t \text{ is substitutable for } x \text{ in } \varphi.$$

²This models are called closed models in [9] but we decided, after some discussions with colleagues, to use the more informative name of quasi-witnessed models. We take into account the fact that the name closed is used in mathematics and logic in many different context with different meanings and could induce some confusions.

³In what follows we implicitly maintain that we are considering always *safe* models: a safe model, as defined in [4], is a model in which, for each first order formula of a given language, the value (the required infima and suprema) is always defined.

$$(\forall 2) \quad (\forall x)(\chi \rightarrow \varphi) \rightarrow (\chi \rightarrow (\forall x)\varphi(x)), \text{ where } x \text{ is not free in } \chi.$$

$$(\exists 2) \quad (\forall x)(\varphi \rightarrow \chi) \rightarrow ((\exists x)\varphi(x) \rightarrow \chi), \text{ where } x \text{ is not free in } \chi.$$

$$(\forall 3) \quad (\forall x)(\chi \vee \varphi) \rightarrow (\chi \vee (\forall x)\varphi(x)), \text{ where } x \text{ is not free in } \chi.$$

and which has, as rules of inference, *Modus Ponens* (MP) and *generalization* (G): From φ infer $(\forall x)\varphi(x)$.

Definition 2.2 Let $\mathcal{L}^{\forall}$ be an axiomatic extension of $SMTL^{\forall}$. We denote by $\mathcal{L}^{\forall qw}$ the axiomatic extension of $\mathcal{L}^{\forall}$ by the following axiom schemata called, from now on, "quasi-witnessed axioms":

$$(C\exists) \quad (\exists y)((\exists x)\varphi(x) \rightarrow \varphi(y)),$$

$$(\Pi C\forall) \quad \neg\neg(\forall x)\varphi(x) \rightarrow ((\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x))).$$

These quasi-witnessed axioms are a modification of witnessed axioms. The first one, $(C\exists)$, is a witnessed axiom and the second one says that the witnessed axiom $(C\forall) (\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x))$ is valid only when $(\forall x)\varphi(x)$ is different from 0, i.e., when $\neg\neg(\forall x)\varphi(x) = 1$.

Definition 2.3 The logic $\Pi\forall$ (resp. $\Pi\forall^{qw}$) is the axiomatic extension of $SMTL^{\forall}$ (resp. $SMTL^{\forall qw}$) by the following axioms⁴:

$$(C) \quad \varphi \wedge \psi \leftrightarrow \varphi \odot (\varphi \rightarrow \psi),$$

$$(\Pi) \quad \neg\neg\chi \rightarrow (((\varphi \odot \chi) \rightarrow (\psi \odot \chi)) \rightarrow (\varphi \rightarrow \psi)).$$

The following definitions are required to prove the main results given in Section 3. They are typical within the framework of Classical First Order Logic. Their presentation in our context, slightly different from the classical one, follows the generalization, due to [8], and necessary to adapt them to a many-valued framework.

Definition 2.4 We say that a theory T' in a predicate language Γ' is an expansion of a theory T in a predicate language Γ , if $\Gamma \subseteq \Gamma'$ and, each formula provable in T is provable in T' . We say that T' is a conservative expansion of T if T' is an expansion of T and each formula in the language of T , provable in T' , is provable in T .

Definition 2.5 A theory T is linear if, for each pair of sentences φ, ψ , we have $T \vdash \varphi \rightarrow \psi$ or $T \vdash \psi \rightarrow \varphi$.

⁴Alternatively, $\Pi\forall$ (resp. $\Pi\forall^{qw}$) can be defined as the expansion of the well known Propositional Product Logic Π by classical quantifiers and axioms in Definition 2.2 (resp. 2.1).

Definition 2.6 Let Γ and Γ' be predicate languages such that $\Gamma \subseteq \Gamma'$ and T a Γ' -theory. We say that T is $\forall$ - Γ -Henkin if, for each Γ -sentence $\varphi = (\forall x)\psi(x)$ such that $T \not\models \varphi$, there is a constant c in Γ' such that $T \not\models \psi(c)$.

We say that T is $\exists$ - Γ -Henkin if, for each Γ -sentence $\varphi = (\exists x)\psi(x)$ such that $T \vdash \varphi$, there is a constant c in Γ' such that $T \vdash \psi(c)$.

A theory is called Γ -Henkin if it is both $\forall$ - Γ -Henkin and $\exists$ - Γ -Henkin.

If $\Gamma = \Gamma'$, we say that T is $\forall$ -Henkin ($\exists$ -Henkin, Henkin).

From a semantic point of view, before defining what a first order model is, we need a definition of SMTL-algebra.

Definition 2.7 An SMTL-algebra $\mathbf{A} = \langle \mathbf{A}, \cap, \cup, *, \Rightarrow, 0, 1 \rangle$ is a bounded commutative integral residuated lattice which satisfies the following equations:

1. $(x \Rightarrow y) \cup (y \Rightarrow x) = 1$,
2. $x \cap (x \Rightarrow 0) = 0$.

Moreover, if it is linearly ordered, we say that it is a SMTL-chain.

For each axiomatic extension $\mathcal{L}$ of SMTL, a $\mathcal{L}$ -algebra is a SMTL-algebra satisfying the equations corresponding to added axioms.

Let Γ be a predicate language without function symbols and $\mathbf{A}$ a SMTL-chain. An $\mathbf{A}$ -structure for a given predicate language Γ is a structure $\mathbf{M} = (M, (P_{\mathbf{M}})_{P \in \Gamma}, (c_{\mathbf{M}})_{c \in \Gamma})$, where $M \neq \emptyset$, each $P_{\mathbf{M}}$ is an n -ary $\mathbf{A}$ -fuzzy relation on M and each $c_{\mathbf{M}}$ is an element of M . The truth value $\|\varphi\|^{\mathbf{A}, \mathbf{M}}$ of a predicate formula φ is defined in the usual way and so are the concepts of *satisfiability* and *validity*. In [8], we find the following useful definitions and result, which we report without proof. In what follows, we will denote by $\mathbf{A}$ any SMTL-chain.

Definition 2.8 Let $(\mathbf{A}_1, \mathbf{M}_1)$ and $(\mathbf{A}_2, \mathbf{M}_2)$ be models in the languages Γ_1 and Γ_2 respectively and let $\Gamma_1 \subseteq \Gamma_2$. We say that a pair (f, g) is an elementary embedding if:

1. the mapping f is an injection of M_1 into M_2 ,
2. the mapping g is an embedding of $\mathbf{A}_1$ into $\mathbf{A}_2$,
3. for each Γ_1 -formula $\varphi(x_1, \dots, x_n)$ and $a_1, \dots, a_n \in M_1$, it holds that $g(\|\varphi(a_1, \dots, a_n)\|^{\mathbf{A}_1, \mathbf{M}_1}) = \|\varphi(f(a_1), \dots, f(a_n))\|^{\mathbf{A}_2, \mathbf{M}_2}$.

Definition 2.9 Let T be a linear Henkin theory, then the canonical model of T is the structure

$(\mathbf{Lind}_T, \mathbf{CM}(T))$, where $\mathbf{Lind}_T$ is the Lindenbaum algebra of theory T , the domain of $\mathbf{CM}(T)$ consists of the object constants $c_{\mathbf{CM}(T)} = c$ and, for every predicate n -ary symbol $P \in \Gamma$, $P_{\mathbf{CM}(T)}(t_1, \dots, t_n) = [P(t_1, \dots, t_n)]_T$.

From here on, for simplicity, we will write $\mathbf{CM}(T)$ to denote $(\mathbf{Lind}_T, \mathbf{CM}(T))$.

Definition 2.10 Let $(\mathbf{A}, \mathbf{M})$ be a structure and $\text{Alg}((\mathbf{A}, \mathbf{M}))$ be the subalgebra of $\mathbf{A}$ whose domain is the set $\{\|\varphi\|_v^{\mathbf{A}, \mathbf{M}} \mid \varphi, v\}$ of truth degrees of formulas under all $\mathbf{M}$ -evaluation v of variables. Call $(\mathbf{A}, \mathbf{M})$ exhaustive if $\mathbf{A} = \text{Alg}((\mathbf{A}, \mathbf{M}))$.

The following lemma is a direct consequence of Lemma 4 of [8] and we will not prove it here.

Lemma 2.11 Let T_1, T_2, T be SMTL $\forall$ -theories. If T_2 is a conservative expansion of T_1 , then, for each exhaustive model $(\mathbf{A}, \mathbf{M})$ of T_1 , there exists a linear Henkin theory T extending T_2 such that $(\mathbf{A}, \mathbf{M})$ can be elementarily embedded into $\mathbf{CM}(T)$.

3 COMPLETENESS WITH RESPECT TO QUASI-WITNESSED MODELS

In this section we will state and prove the main result of this paper, i.e., that if we add axioms $\text{C}\exists$ and $\text{II}\text{C}\forall$ to any predicate fuzzy logic extending SMTL $\forall$, we obtain a logic that is complete with respect to quasi-witnessed models. Within this section, we will write $\mathcal{L}\forall$ to denote any axiomatic extension SMTL $\forall$.

Definition 3.1 Let Γ be a predicate language, and $(\mathbf{A}, \mathbf{M})$ a model, then we say that $(\mathbf{A}, \mathbf{M})$ is quasi-witnessed if, for every Γ -formula $\varphi(x, y_1, \dots, y_n)$:

1. For each tuple $c_1, \dots, c_n$ of elements in M there exist $a \in M$ such that $\|(\exists x)\varphi(x, c_1, \dots, c_n)\|^{\mathbf{A}, \mathbf{M}} = \|\varphi(a, c_1, \dots, c_n)\|^{\mathbf{A}, \mathbf{M}}$.
2. For each tuple $c_1, \dots, c_n$ of elements in M either $\|(\forall x)\varphi(x, c_1, \dots, c_n)\|^{\mathbf{A}, \mathbf{M}} = 0$, or there exists $b \in M$ such that $\|(\forall x)\varphi(x, c_1, \dots, c_n)\|^{\mathbf{A}, \mathbf{M}} = \|\varphi(b, c_1, \dots, c_n)\|^{\mathbf{A}, \mathbf{M}}$.

Lemma 3.2 If a model $(\mathbf{A}, \mathbf{M})$ of $\mathcal{L}\forall$ is quasi-witnessed, then $(\mathbf{A}, \mathbf{M})$ satisfies $(\text{C}\exists)$ and $(\text{II}\text{C}\forall)$.

Proof Let $(\mathbf{A}, \mathbf{M})$ be a quasi-witnessed model of $\mathcal{L}\forall$, then:

1. Since, by the first condition of Definition 3.1, there exists $a \in M$ such

that $\|\varphi(a)\|^{(\mathbf{A}, \mathbf{M})} = \|(\exists x)\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}$, then $(\mathbf{A}, \mathbf{M}) \models (\exists x)\varphi(x) \rightarrow \varphi(a)$. So, by axiom $(\exists 1)$ and (MP) , $(\mathbf{A}, \mathbf{M}) \models (\exists y)((\exists x)\varphi(x) \rightarrow \varphi(y))$.

2. By the second condition of Definition 3.1, there exists $b \in M$ such that either $\|\varphi(b)\|^{(\mathbf{A}, \mathbf{M})} = \|(\forall x)\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}$, or $\|(\forall x)\varphi(x)\|^{(\mathbf{A}, \mathbf{M})} = 0$. If $\|(\forall x)\varphi(x)\|^{(\mathbf{A}, \mathbf{M})} = 0$, then, $\|\neg(\forall x)\varphi(x)\|^{(\mathbf{A}, \mathbf{M})} = 0$ and, trivially, $(\mathbf{A}, \mathbf{M}) \models \neg(\forall x)\varphi(x) \rightarrow ((\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x)))$. If, on the other hand, $\|\varphi(b)\|^{(\mathbf{A}, \mathbf{M})} = \|(\forall x)\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}$, then $(\mathbf{A}, \mathbf{M}) \models \varphi(b) \rightarrow (\forall x)\varphi(x)$, and, by axiom $(\exists 1)$ and (MP) , $(\mathbf{A}, \mathbf{M}) \models (\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x))$. So, $(\mathbf{A}, \mathbf{M}) \models \neg(\forall x)\varphi(x) \rightarrow ((\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x)))$.

As for witnessed models and $L^{\forall w}$, the converse of the last lemma does not hold: to show this fact, consider the model $([0, 1]_{\Pi}, (N, r_P))$, where $r_P(n) = \frac{1}{m} + \frac{1}{n+2}$, for a fixed but arbitrary positive integer $m > 1$. Such model, when we take $\varphi(x) = P(x)$, validates axiom $(\Pi C\forall)$, but it is not quasi-witnessed because, on the one hand, $\|(\forall x)\varphi(x)\|^{([0, 1]_{\Pi}, (N, r_P))} = \frac{1}{m} > 0$ and on the other hand, for each $n \in N$, $\|\varphi(n)\|^{([0, 1]_{\Pi}, (N, r_P))} > \frac{1}{m} = \|(\forall x)\varphi(x)\|^{([0, 1]_{\Pi}, (N, r_P))}$. So, it does not respect condition 2 of Definition 3.1. However, as in [8], it is possible to prove the following result.

Lemma 3.3 *Let Γ be a predicate language, and $(\mathbf{A}, \mathbf{M})$ an exhaustive model of a Γ -theory T . Then $(\mathbf{A}, \mathbf{M})$ is a $\mathcal{L}^{\forall qw}$ -model of T iff it can be elementarily embedded into a quasi-witnessed model of T .*

Proof $(\Rightarrow)$ Let $(\mathbf{A}, \mathbf{M})$ be an exhaustive $\mathcal{L}^{\forall qw}$ -model of T . By Lemma 2.11, there is a linear Henkin theory T' extending T , such that $(\mathbf{A}, \mathbf{M})$ can be elementarily embedded into $\mathbf{CM}(T')$. Hence $\mathbf{CM}(T')$ is a $\mathcal{L}^{\forall qw}$ -model of T and we have to show that $\mathbf{CM}(T')$ is quasi-witnessed.

Due to the construction of the canonical model, each element of the domain of $\mathbf{CM}(T')$ is a constant. Let $\varphi(x)$ be a formula with one free variable, then:

1. Suppose, on the one hand, that $\|(\forall x)\varphi(x)\|^{(\mathbf{CM}(T'))} > 0$, then $\|\neg(\forall x)\varphi(x)\|^{(\mathbf{CM}(T'))} = 0$. Since, by axiom $(\Pi C\forall)$, we have that $T' \vdash \neg(\forall x)\varphi(x) \rightarrow ((\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x)))$, then, by (MP) , $T' \vdash (\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x))$. Since T' is $\exists$ -Henkin, then there exists some c such that $T' \vdash \varphi(c) \rightarrow (\forall x)\varphi(x)$. So, by axiom $(\forall 1)$, we obtain that $\|\varphi(c)\|^{(\mathbf{CM}(T'))} = \|(\forall x)\varphi(x)\|^{(\mathbf{CM}(T'))}$.

2. Suppose, on the other hand, that, for every constant $c \in \Gamma$, $\|\varphi(c)\|^{(\mathbf{CM}(T'))} \neq \|(\forall x)\varphi(x)\|^{(\mathbf{CM}(T'))}$, then we have necessarily that $\|(\forall x)\varphi(x)\|^{(\mathbf{CM}(T'))} = 0$, because, otherwise, as we have shown before, we have that $\|\varphi(c)\|^{(\mathbf{CM}(T'))} = \|(\forall x)\varphi(x)\|^{(\mathbf{CM}(T'))}$, contradicting the supposition.

The proof of the other condition is similar to Hájek's proof of Lemma 5 in [8] and we will not repeat it here.

- $(\Leftarrow)$ Suppose now that $(\mathbf{A}, \mathbf{M})$ can be elementarily embedded into a quasi-witnessed model of T , hence, by Lemma 3.2, is a $\mathcal{L}^{\forall qw}$ -model of T .

Theorem 3.4 *Let T be a theory and φ a formula in a given predicate language, then $T \vdash_{\mathcal{L}^{\forall qw}} \varphi$ iff $(\mathbf{A}, \mathbf{M}) \models \varphi$ for every quasi-witnessed model $(\mathbf{A}, \mathbf{M})$ of the theory T .*

Proof The completeness of $\mathcal{L}^{\forall}$ with respect to all (not only quasi-witnessed) $(\mathbf{A}, \mathbf{M})$ -models is ensured by Theorem 5 of [8], so we will restrict ourselves to the *quasi-witnessed* part.

- $(\Rightarrow)$ As a consequence of Theorem 5 of [8], we only have to check whether a quasi-witnessed model satisfies axioms $(C\exists)$ and $(\Pi C\forall)$, but this result has been already shown in Lemma 3.2.
- $(\Leftarrow)$ Suppose that $T \not\vdash_{\mathcal{L}^{\forall qw}} \varphi$, then there exists a $\mathcal{L}^{\forall qw}$ -model $(\mathbf{A}, \mathbf{M})$ of T , such that $(\mathbf{A}, \mathbf{M}) \not\models \varphi$. Hence, by Lemma 3.3, there exists a quasi-witnessed model $(\mathbf{A}', \mathbf{M}')$ of T such that $(\mathbf{A}', \mathbf{M}') \not\models \varphi$.

4 THE CASE OF PREDICATE PRODUCT LOGIC

In this section we will show that axioms $(C\exists)$ and $(\Pi C\forall)$ are provable in $\Pi\forall$, i.e., that the logics $\Pi\forall$ and $\Pi\forall^{qw}$ are equivalent. In order to do that, let us recall that $\Pi\forall$ is complete with respect to all models over a product chain and any product chain is isomorphic to the negative cone of a linearly ordered abelian group with an added bottom (See Theorem 2.5 in [3]).

Definition 4.1 *Let $\mathbf{G} = \langle \mathbf{G}, +, -, \mathbf{0} \rangle$ be a totally ordered abelian group, then we denote by G^- the negative part of G , i.e., $G^- = \{x \in G \mid x \leq 0\}$. Moreover, we denote by $\mathfrak{P}(\mathbf{G})$ the structure $\langle G^- \cup \perp, \otimes, \Rightarrow, \perp \rangle$, where $\perp$ is an element which does not belong to G , and $\otimes, \Rightarrow$ are two binary operations defined as follows:*

$$x \otimes y = \begin{cases} x + y & \text{if } x, y \in G^-, \\ \perp & \text{otherwise,} \end{cases}$$

and

$$x \Rightarrow y = \begin{cases} 0 \wedge (y - x) & \text{if } x, y \in G^-, \\ 0 & \text{if } x = \perp, \\ \perp & \text{if } x \in G^- \text{ and } y = \perp. \end{cases}$$

As a consequence of Theorem 2.5 and Remark 2.2 of [3] we have the following useful result.

Proposition 4.2 *Let A be a non-trivial Π -chain⁵. There exists a linearly ordered abelian group $\mathbf{G}$, such that $A \cong \mathfrak{P}(\mathbf{G})$. Moreover, $\mathbf{G}$ is univocally determined up to isomorphism.*

Notice that the isomorphism of the above proposition maps the neutral element of the group onto the maximum element of the product chain and the added bottom $\perp$ to the minimum element of the product chain. This last result allows us to look at the theory of linearly ordered abelian groups. Moreover, let $\mathbf{G}$ be a linearly ordered abelian group and $a, \{a_i\}_{i \in \omega} \in G$: it is well known that, on the one hand, if $\{a_i\}_{i \in \omega}$ is an increasing sequence and has limit a , then $\{a - a_i\}_{i \in \omega}$ is a decreasing sequence and has limit 0. On the other hand, if $\{a_i\}_{i \in \omega}$ is a decreasing sequence and has limit a , then $\{a_i - a\}_{i \in \omega}$ is a decreasing sequence and has limit 0. So, since, by Definition 4.1, the truncated subtraction of the group is the interpretation of product implication and the constant 0 of the group is the isomorphic image of the maximum element 1 of the product chain, then, by means of Proposition 4.2, we can infer the following corollary.

Corollary 4.3 *Let $\mathbf{A}$ be a product chain and $a, \{a_i\}_{i \in \omega} \in A$, then:*

1. *if $\{a_i\}_{i \in \omega}$ is an increasing sequence with limit a , then $\{a \Rightarrow a_i\}_{i \in \omega}$ is an increasing sequence with limit 1,*
2. *if $\{a_i\}_{i \in \omega}$ is a decreasing sequence with limit a , then $\{a_i \Rightarrow a\}_{i \in \omega}$ is an increasing sequence with limit 1.*

With the help of the last corollary, we can prove the main result of this section.

Lemma 4.4 *The quasi-witnessed axioms $(C\exists)$ and $(\Pi C\forall)$ are theorems of $\Pi\forall$.*

⁵A Π -chain is a SMTL-chain which satisfies the equations:

1. $x \cap y = x * (x \Rightarrow y)$,
2. $((z \Rightarrow 0) \Rightarrow 0) \Rightarrow ((x * z) \Rightarrow (y * z) \Rightarrow (x \Rightarrow y))01$.

Proof We will show it semantically. Since $\Pi\forall$ is complete w.r.t. models over linearly ordered product algebras, we have to prove that the quasi-witnessed axioms are tautologies for these models. Let $\mathbf{A}$ be a product chain and let $(\mathbf{A}, \mathbf{M})$ be a model of $\Pi\forall$, then:

$(C\exists)$ Since $\|(\exists y)((\exists x)\varphi(x) \rightarrow \varphi(y))\|^{(\mathbf{A}, \mathbf{M})} = \sup_y \{\sup_x \{\|\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}\} \Rightarrow \|\varphi(y)\|^{(\mathbf{A}, \mathbf{M})}\}$ and variables x and y range over the same values, then, by Corollary 4.3, $\|(\exists y)((\exists x)\varphi(x) \rightarrow \varphi(y))\|^{(\mathbf{A}, \mathbf{M})} = 1$. So, axiom $(C\exists)$ is a theorem of $\Pi\forall$.

$(\Pi C\forall)$ We know that $\|\neg\neg(\forall x)\varphi(x) \rightarrow ((\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x)))\|^{(\mathbf{A}, \mathbf{M})} = \neg\neg \inf_x \{\|\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}\} \Rightarrow \sup_y \{\|\varphi(y)\|^{(\mathbf{A}, \mathbf{M})}\} \Rightarrow \inf_x \{\|\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}\}$. If $\inf_x \{\|\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}\} = 0$, the result is obvious. Otherwise (being a Gödel negation) $\neg\neg \inf_x \{\|\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}\} = 1$ and, therefore, the value of the whole formula will be equal to 1 iff $\|(\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x))\|^{(\mathbf{A}, \mathbf{M})} = \sup_y \{\|\varphi(y)\|^{(\mathbf{A}, \mathbf{M})} \Rightarrow \inf_x \{\|\varphi(x)\|^{(\mathbf{A}, \mathbf{M})}\}\} = 1$, but this is a direct consequence of Corollary 4.3. So, axiom $(\Pi C\forall)$ is a theorem of $\Pi\forall$.

This last result, together with Theorem 3.4, is an alternative way to prove the result of Laskowski and Malekpour in [9].

Corollary 4.5 *Let T be a theory and φ a formula in a given predicate language, then $T \vdash_{\Pi\forall} \varphi$ iff $(\mathbf{A}, \mathbf{M}) \models \varphi$ for every quasi-witnessed model $(\mathbf{A}, \mathbf{M})$ of the theory T .*

Next we adapt and generalize the result in [7]. Actually we can show that there is no other logic of a continuous t-norm that is complete with respect to quasi-witnessed models, but Product or Łukasiewicz. Denote by $\mathcal{L}(\ast)$ the propositional logic of a continuous t-norm $\ast$ (complete with respect to valuations over the standard chain $[0, 1]_\ast$)⁶ and by $\mathcal{L}(\ast)\forall$ its first order expansion.

Lemma 4.6 *If a continuous t-norm $\ast$ is different from Product or Łukasiewicz, then there exists a model of $\mathcal{L}(\ast)\forall$ over $[0, 1]_\ast$ which does not satisfy axioms $(C\exists)$ and $(\Pi C\forall)$.*

Proof If a continuous t-norm $\ast$ is different from Product or Łukasiewicz, then it has at least one element $a \in (0, 1)$ which is idempotent. Let $([0, 1]_\ast, (N, r_P))$ be a model of $\mathcal{L}(\ast)\forall$ and $\{a_n\}_{n \in \omega}$ a sequence of elements of $[0, 1]$, different from a .

⁶By $[0, 1]_\ast = \langle [0, 1], \max, \min, \ast, \Rightarrow, \neg, 0, 1 \rangle$ is denoted the linear algebra defined over the real unit interval by a left-continuous t-norm $\ast$ and its residuum $\Rightarrow$.

(C $\exists$) Consider first order the interpretation such that $\|r_P(n)\|^{([0,1]_*,(N,r_P))} = a_n$, $\{a_n\}_{n \in \omega}$ is an increasing sequence of elements of $[0,1]$ and $\sup\{a_n\}_{n \in \omega} = a \neq 1$. In this model, when we take $\varphi(x) = P(x)$, we have that $\|(\exists y)((\exists x)\varphi(x) \rightarrow \varphi(y))\|^{([0,1]_*,(N,r_P))} = \sup_{m \in \omega} \{ \sup_{n \in \omega} \{a_n\} \Rightarrow a_m \} = \sup_{m \in \omega} \{a \Rightarrow a_m\} = \sup_{m \in \omega} \{a_m\} = a \neq 1$. So, (C $\exists$) is not a theorem of $\mathcal{L}(\ast)\forall$.

(Π C $\forall$) Consider the first order interpretation such that $\|r_P(n)\|^{([0,1]_*,(N,r_P))} = a_n$, $\{a_n\}_{n \in \omega}$ is a decreasing sequence of elements of $[0,1]$ and $\inf\{a_n\}_{n \in \omega} = a \neq 1$. In this model, when we take $\varphi(x) = P(x)$, we have that $\|\neg\neg(\forall x)\varphi(x) \rightarrow (\exists y)(\varphi(y) \rightarrow (\forall x)\varphi(x))\|^{([0,1]_*,(N,r_P))} = \neg\neg(\inf_{n \in \omega} \{a_n\}) \Rightarrow \sup_{m \in \omega} \{a_m\} \Rightarrow \inf_{n \in \omega} \{a_n\} = 1 \Rightarrow \sup_{m \in \omega} \{a_m\} = 1 \Rightarrow a = a \neq 1$. So, (Π C $\forall$) is not a theorem of $\mathcal{L}(\ast)\forall$.

This last result allows us to prove the next general result.

Proposition 4.7 *If $\ast$ is a continuous t-norm, $\mathcal{L}(\ast)\forall$ proves both (C $\exists$) and (Π C $\forall$) iff $\ast$ is either Lukasiewicz or Product.*

Proof One direction is proven in Corollary 4.5 for Product Logic and is a consequence of witnessed completeness for Lukasiewicz. The other direction is a direct consequence of Lemma 4.6.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have studied quasi-witnessed models and introduced quasi-witnessed axioms, that are an extension of witnessed axioms. Finally we have showed that the extension of any SMTL $\forall$ logic with quasi-witnessed axioms is complete with respect to quasi-witnessed models. But quasi-witnessed axioms are provable in Product first order logic and then we recover the quasi-witnessed completeness proved in [9]. We also show that the only logic of a continuous t-norm that prove quasi-witnessed axioms are Lukasiewicz and Product. As future work we have in mind to study two different subjects:

1.- Is it possible to find results of decidability of Product Description Logic based on quasi-witnessed models (modifying and extending Hájek's results for witnessed models)?

2.- Is it possible to characterize the axiomatic extensions of a first order logic of a left-continuous t-norm with the witnessed axioms? And the same question for a first order logic of a strict left-continuous t-norm with the quasi-witnessed axioms?

Acknowledgements: The authors acknowledge partial support of the Spanish projects MLOG2 TIN2007-68005-C04-01/04, "Agreement Technologies" (CONSOLIDER CSD 2007-0022, INGENIO 2010) and the Generalitat de Catalunya under the grant 2009-SGR-1434. The first author is supported by grant JAEPreDoc, n.074 from CSIC. Acknowledgments are due to P. Cintula and P. Dellunde for their suggestions and comments.

References

- [1] F. Baader, et al.. The Description Logic Handbook - Theory, Implementation and Applications. *Cambridge University Press*, Cambridge, 2003.
- [2] F. Bobillo, U. Straccia. Towards a Crisp Representation of Fuzzy Description Logics under Lukasiewicz semantics. *Proceedings of the 17th International Symposium on Methodologies for Intelligent Systems (ISMIS-08)*, Pág. 309 -318, 2008.
- [3] R. Cignoli, A. Torrens. An algebraic analysis of product logic. *Multiple-valued logic*, 5, Pág. 45-65, 2000.
- [4] P. Hájek. Metamathematics of fuzzy logic. *Trends in Logic*, vol.4, Kluwer, Dordrecht, 1998.
- [5] P. Hájek. Making fuzzy description logic more general. *Fuzzy Sets and Systems*, 154, Pág. 1-15, 2005.
- [6] P. Hájek. On witnessed models in fuzzy logic. *Math. Log. Q.*, 53(1), Pág. 66-77, 2007.
- [7] P. Hájek. On witnessed models in fuzzy logic II. *Math. Log. Q.*, 53(6), Pág. 610-615, 2007.
- [8] P. Hájek, P. Cintula. On theories and models in fuzzy predicate logic. *Journal of Symbolic Logic*, 71, Pág. 863-880, 2006.
- [9] M. C. Laskowski, S. Malekpour. Provability in predicate product logic. *Archive for Mathematical Logic*, 46, Pág. 365-378, 2007.

Interpretación de las soluciones de $X \triangleleft R = X$ como bloques de tolerancia borrosa y su relación con el análisis de conceptos borrosos.

Cristina Alcalde¹ Ana Burusco² Ramón Fuentes-González²

¹ Departamento de Matemática Aplicada. Universidad del País Vasco.
Escuela Universitaria Politécnica. Plaza de Europa 1, 20018 San Sebastián, Spain.
c.alcalde@ehu.es

² Departamento de Automática y Computación. Universidad Pública de Navarra
Campus de Arrosadía, 31006 Pamplona, Spain.
{burusco, rfuentes}@unavarra.es

Resumen

Utilizando una relación borrosa $R \in [0, 1]^{E \times E}$ reflexiva y simétrica, se caracterizan ciertos subconjuntos borrosos $B \in [0, 1]^E$, demostrando que juegan el papel de los subconjuntos ordinarios llamados *bloques* en el contexto de relaciones de tolerancia nítidas. Finalmente, se prueba que estos nuevos bloques borrosos B pueden determinarse mediante parte del retículo de Conceptos borrosos asociado al contexto $([0, 1], E, E, R)$.

Palabras Clave: Análisis de Conceptos Formales, Análisis de Conceptos Borrosos, Relación borrosa, Tolerancia borrosa.

1. INTRODUCCIÓN

En este trabajo, dada una tolerancia borrosa $R \in [0, 1]^{E \times E}$ sobre un referencial cualquiera E , (finito o no), demostramos que existe una familia distinguida $\mathcal{T} \neq \emptyset$ de subconjuntos borrosos $B \in [0, 1]^E$ llamados *bloques de tolerancia borrosa* que constituye un cubrimiento del referencial: $E = \bigvee_{B \in \mathcal{T}} B$. Probamos

además que esos bloques son las soluciones de la ecuación relacional borrosa $X \triangleleft R = X$ donde $\triangleleft$ es la composición “triángulo” asociada a una implicación residuada. Finalmente, relacionamos el cálculo de los bloques borrosos B con el cálculo de conceptos borrosos ([7], [1], [3], [13], [14]) asociados al contexto $([0, 1], E, E, R)$ en el que los conjuntos de objetos y atributos son iguales al referencial E .

1.1. BLOQUES ASOCIADOS A TOLERANCIAS

En la teoría de relaciones binarias nítidas, dados un referencial $E \neq \emptyset$ y una *tolerancia* R en E , (es decir, una relación nítida $R \subseteq E \times E$ reflexiva y simétrica), se asocia al par (E, R) la siguiente familia $\mathcal{SE}_R$ de subconjuntos nítidos de E :

$$\mathcal{SE}_R = \{A \in \wp(E) / A \times A \subseteq R\}$$

Los subconjuntos A de $\mathcal{SE}_R$ reciben el nombre de *pre-bloques* [2] del sistema algebraico (E, R) y este par es un *espacio de tolerancia* [9].

El conjunto $\mathcal{SE}_R$ de pre-bloques es no vacío, ya que tanto el subconjunto vacío $\emptyset$ como los subconjuntos unitarios $\{x\}$ de E , son pre-bloques de (E, R) .

El par $(\mathcal{SE}_R, \subseteq)$ es un subconjunto ordenado del retículo $(\wp(E), \subseteq)$. Se puede demostrar [2] que, para cualquier E (finito o no) y para cualquier tolerancia R en E , el conjunto ordenado $(\mathcal{SE}_R, \subseteq)$ es inductivo, por lo que su subconjunto de maximales es no vacío. Sea $\mathcal{BE}_R = \{B \in \mathcal{SE}_R / B \text{ es maximal en } (\mathcal{SE}_R, \subseteq)\}$ ese subconjunto. Si un pre-bloque $A \in \mathcal{SE}_R$ no es maximal diremos que es *pre-bloque propio*. Cada uno de los maximales $B \in \mathcal{BE}_R$ recibe el nombre de *bloque* [2] del espacio de tolerancia (E, R) . El conjunto de bloques $\mathcal{BE}_R$ constituye un cubrimiento de E , es decir, $E = \bigcup_{B \in \mathcal{BE}_R} B$ (Figura 1).

Si R es además transitiva, es decir, R es una equivalencia en E , entonces el conjunto $\mathcal{BE}_R$ de bloques coincide con el conjunto de clases de equivalencia de (E, R) . En este caso, el cubrimiento de E resulta ser una partición del mismo.

1.2. RETÍCULO DE CONCEPTOS FORMALES

Desde una perspectiva distinta, se puede considerar el espacio de tolerancia (E, R) mencionado en la sección

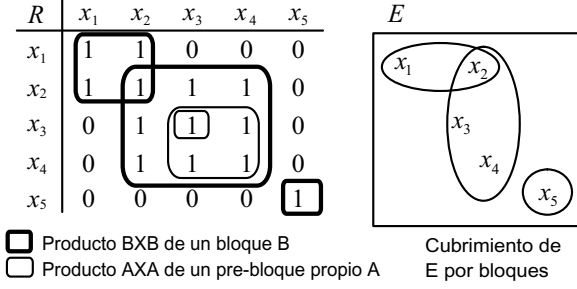


Figura 1: Cubrimiento por bloques.

anterior como el contexto (E, E, R) en el que coinciden los conjuntos de *objetos* y de *atributos* y en el que la relación de *incidencia* es la tolerancia R .

En este caso, el *retículo de conceptos formales* $(\mathcal{L}(E, E, R), \leq)$ ([16], [11]), está formado por el conjunto $\mathcal{L}(E, E, R)$ de pares de subconjuntos $(A, B) \in \wp(E) \times \wp(E)$ tales que $A_1 = B$ y $B_2 = A$, siendo $A_1 = \{y / xRy \ \forall x \in A\}$ y $B_2 = \{x / xRy \ \forall y \in B\}$. Si (A, B) es uno de esos conceptos, entonces A es su *extensión* (subconjunto de objetos) y B es su *intensión* (subconjunto de atributos).

El orden $\leq$ del retículo $(\mathcal{L}(E, E, R), \leq)$ viene dado por

$$(A, B) \leq (C, D) \iff A \subseteq C$$

o lo que es equivalente, $B \supseteq D$.

Se demuestra que $A \neq \emptyset$, $B \neq \emptyset$ y $(A, B) \in \mathcal{L}(E, E, R)$ sí y solo si $A \times B \subseteq R$ es un “*rectángulo maximal*” de la tolerancia R . (Figura 2).

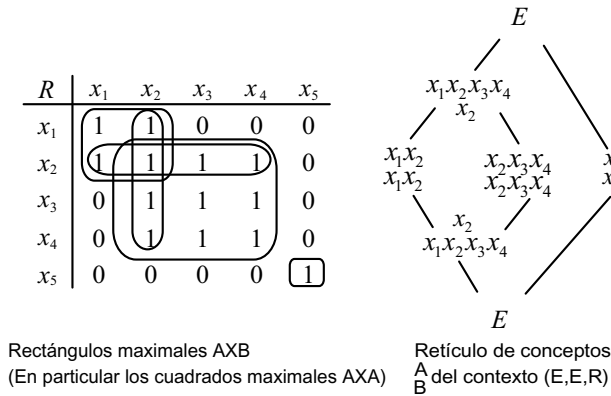


Figura 2: Rectángulos maximales y conceptos formales

1.3. TOLERANCIAS BORROSAS

En este trabajo estudiamos espacios de tolerancia (E, R) en los que R es una *tolerancia borrosa* [15], [8],

también llamada *compatibilidad borrosa* [12]. Es decir, R es una aplicación $R : E \times E \rightarrow [0, 1]$ tal que $R(x, x) = 1 \ \forall x \in E$ y $R(x, y) = R(y, x) \ \forall (x, y) \in E \times E$.

En lo que sigue, la expresión R^{op} representa la relación opuesta de R tal que $R^{op}(x, y) = R(y, x) \ \forall (x, y) \in E \times E$.

Los elementos que utilizamos en el trabajo para obtener los bloques asociados a una de esas tolerancias borrosas R , son los siguientes:

- (i) Una t -norma $\otimes$ en $[0, 1]$ continua a la izquierda, es decir, $\alpha \otimes \sup M = \sup(\alpha \otimes M) \ \forall \alpha \in [0, 1] \ \forall M \subseteq [0, 1]$.
- (ii) La implicación residuada $\rightsquigarrow$ asociada a esa t -norma: $\alpha \rightsquigarrow \beta = \sup\{\gamma / \alpha \otimes \gamma \leq \beta\} \ \forall (\alpha, \beta) \in [0, 1]^2$.
- (iii) Un referencial ordinario E tal que $|E| \geq 2$.
- (iv) El operador “ $\otimes$ -producto” entre subconjuntos borrosos $\tilde{\times} : [0, 1]^E \times [0, 1]^E \rightarrow [0, 1]^E$ asociado a la t -norma $\otimes$, definido, $\forall A \in [0, 1]^E, \ \forall B \in [0, 1]^E$ y $\forall (x, y) \in E \times E$, por:

$$(A \tilde{\times} B)(x, y) = A(x) \otimes B(y)$$

- (v) El operador “ $\rightsquigarrow$ -triángulo” entre subconjuntos borrosos $\triangleleft : [0, 1]^E \times [0, 1]^E \rightarrow [0, 1]^E$ asociado a la implicación $\rightsquigarrow$, definido, $\forall A \in [0, 1]^E, \ \forall R \in [0, 1]^E$ y $\forall y \in E$, por:

$$(A \triangleleft R)(y) = \inf\{A(x) \rightsquigarrow R(x, y) / x \in E\}$$

Definición 1. Llamaremos $\otimes$ -rectángulo en $E \times E$ a una relación borrosa $Q \in [0, 1]^{E \times E}$ que pueda representarse como $\otimes$ -producto $A \tilde{\times} B$ de subconjuntos borrosos de E . Si se verifica que $A = B$, diremos que $Q = A \tilde{\times} A$ es un $\otimes$ -cuadrado.

2. SOLUCIONES DE $X \triangleleft R = X$

En esta sección, se dan algunas propiedades del operador producto $\tilde{\times}$ y se prueba que el operador $\triangleleft$ es residuo de aquél. Posteriormente, se demuestra que dada una tolerancia borrosa R , la ecuación $X \triangleleft R = X$ tiene siempre solución y que cada solución es un subconjunto borroso con cierto carácter de maximal.

Proposición 1. Se verifica:

- (i) $(\bigvee_{A \in \mathcal{F}} A) \tilde{\times} B = \bigvee_{A \in \mathcal{F}} (A \tilde{\times} B) \ \forall \mathcal{F} \subseteq [0, 1]^E$,
 $A \tilde{\times} (\bigvee_{B \in \mathcal{G}} B) = \bigvee_{B \in \mathcal{G}} (A \tilde{\times} B) \ \forall \mathcal{G} \subseteq [0, 1]^E$
- (ii) $(A \tilde{\times} B)^{op} = B \tilde{\times} A \ \forall A, B \in [0, 1]^E$ y, en consecuencia, todo $\otimes$ -cuadrado $A \tilde{\times} A$ es una relación borrosa simétrica: $(A \tilde{\times} A)^{op} = A \tilde{\times} A$.

$$(iii) ((A \tilde{\times} B) \leq R) \Leftrightarrow (B \leq (A \triangleleft R)) \Leftrightarrow (A \leq (B \triangleleft R^{op})).$$

Demostración.

El apartado (i) es consecuencia inmediata de la conmutatividad y del carácter de continua a la izquierda que posee la t -norma $\otimes$. El apartado (ii) es inmediato a partir de la definición de $\tilde{\times}$ y de la conmutatividad de la t -norma $\otimes$. (iii) Supongamos que $A \tilde{\times} B \leq R$. Entonces, para todo x y para todo y en E , se verifica $A(x) \otimes B(y) \leq R(x, y)$. Sea y un elemento cualquiera de E y hagamos que x recorra E . Teniendo en cuenta que la implicación $\rightsquigarrow$ es el residuo de la t -norma $\otimes$, obtenemos que, para todo x , se verifica:

$$B(y) \leq (A(x) \rightsquigarrow R(x, y)),$$

luego

$$B(y) \leq \inf_{x \in E} (A(x) \rightsquigarrow R(x, y)) = (A \triangleleft R)(y).$$

Y ésto ocurre para cualquier y , en consecuencia,

$$B \leq (A \triangleleft R).$$

Supongamos ahora que se cumple esta última desigualdad entre borrosos, entonces

$$B(y) \leq (A(x) \rightsquigarrow R(x, y)) \quad \forall x, y \in E,$$

luego

$$A(x) \otimes B(y) \leq R(x, y) \quad \forall (x, y) \in E \times E.$$

Es decir:

$$A \tilde{\times} B \leq R.$$

Finalmente, teniendo en cuenta que el operador ^{op} es isótono, junto con el resultado del apartado (ii) y lo que acabamos de demostrar, tenemos que

$$(A \tilde{\times} B) \leq R \Leftrightarrow ((A \tilde{\times} B)^{op} \leq R^{op}) \Leftrightarrow (B \tilde{\times} A \leq R^{op}) \Leftrightarrow (A \leq (B \triangleleft R^{op})).$$

□

La primera equivalencia del apartado (iii):

$$(A \tilde{\times} B) \leq R \Leftrightarrow B \leq (A \triangleleft R),$$

demuestra que el operador $\triangleleft$ es el residuo [5] del operador $\tilde{\times}$. En consecuencia, se cumplen las siguientes propiedades de la teoría de residuos:

Proposición 2. (Véase [5]). Los operadores $\triangleleft$ y $\tilde{\times}$ son tales que:

$$A \triangleleft R = \bigvee \{W \in [0, 1]^E / (A \tilde{\times} W) \leq R\}$$

$$A \tilde{\times} B = \bigwedge \{Q \in [0, 1]^E / (A \triangleleft Q) \geq B\}$$

$$A \triangleleft (\bigwedge_{R \in \mathcal{F}} R) = \bigwedge_{R \in \mathcal{F}} (A \triangleleft R) \quad \forall \mathcal{F} \subseteq [0, 1]^E$$

$$A_1 \leq A_2 \implies (A_1 \triangleleft R) \geq (A_2 \triangleleft R) \quad \forall R$$

También, de ese mismo apartado (iii) de la Proposición 1, se deduce el siguiente resultado que utilizaremos más adelante en este trabajo.

Corolario 1. Se verifican las equivalencias:

$$(A \tilde{\times} A) \leq R \Leftrightarrow (A \leq (A \triangleleft R)) \Leftrightarrow (A \leq (A \triangleleft R^{op})).$$

Es conocido [3], que la aplicación residuada $\rightsquigarrow$ es tal que para todo $M \subseteq [0, 1]$ y para todo $\beta \in [0, 1]$ verifica $\sup M \rightsquigarrow \beta = \inf(M \rightsquigarrow \beta)$. No está demostrado que se cumpla una igualdad análoga para el operador $\triangleleft$.

En general, para cualquier R , sólo se puede asegurar la desigualdad: $(\bigvee_{A \in \mathcal{F}} A) \triangleleft R \leq \bigwedge_{A \in \mathcal{F}} (A \triangleleft R)$.

En la siguiente proposición demostramos que para las relaciones borrosas simétricas sí se verifica la igualdad.

Proposición 3. Si $R = R^{op}$ entonces

$$(\bigvee_{A \in \mathcal{F}} A) \triangleleft R = \bigwedge_{A \in \mathcal{F}} (A \triangleleft R) \quad \forall \mathcal{F} \subseteq [0, 1]^E.$$

Demostración.

Si $\mathcal{F} = \emptyset$ la primera de las expresiones separadas por la igualdad equivale a

$$(\bigvee_{[0,1]^E} \emptyset) \triangleleft R = (\emptyset \triangleleft R) = E \times E,$$

y la segunda a $(\bigwedge_{[0,1]^E} \emptyset) = E \times E$. Supongamos que $\mathcal{F} \neq \emptyset$, se verifica la implicación:

$$\begin{aligned} A \leq (\bigvee_{A \in \mathcal{F}} A) \quad \forall A \in \mathcal{F} &\implies \\ &\implies (A \triangleleft R) \geq (\bigvee_{A \in \mathcal{F}} A) \triangleleft R \quad \forall A \in \mathcal{F}, \end{aligned}$$

que muestra que $(\bigvee_{A \in \mathcal{F}} A) \triangleleft R$ es un minorante del subconjunto $\{(A \triangleleft R) / A \in \mathcal{F}\}$. Sea H otro minorante, es decir:

$$H \leq (A \triangleleft R) \quad \forall A \in \mathcal{F}.$$

Según hemos demostrado en la Proposición 1, esta expresión es equivalente a $A \leq (H \triangleleft R^{op}) \quad \forall A \in \mathcal{F}$, y que, al ser $R = R^{op}$, se puede escribir:

$$A \leq (H \triangleleft R) \quad \forall A \in \mathcal{F},$$

lo que a su vez es equivalente a

$$(H \tilde{\times} A) \leq R \quad \forall A \in \mathcal{F}.$$

De este resultado deducimos que $\bigvee_{A \in \mathcal{F}} (H \tilde{\times} A) \leq R$, luego $(H \tilde{\times} \bigvee_{A \in \mathcal{F}} A) \leq R$. Aplicando el opuesto ^{op} , tenemos

$$(\bigvee_{A \in \mathcal{F}} A) \tilde{\times} H \leq R^{op} = R,$$

que, finalmente, equivale a $H \leq (\bigvee_{A \in \mathcal{F}} A) \triangleleft R$. Y esto prueba que $(\bigvee_{A \in \mathcal{F}} A) \triangleleft R$ es el mayor de los minorantes. Es decir, que

$$(\bigvee_{A \in \mathcal{F}} A) \triangleleft R = \bigwedge_{A \in \mathcal{F}} (A \triangleleft R).$$

□

Estudiamos a continuación dos clases $\mathcal{H}$ y $\mathcal{T}$ de subconjuntos borrosos caracterizados por la relación R y el operador $\triangleleft$. Esos subconjuntos servirán como base de las definiciones de bloques que aparecen más adelante en este trabajo.

Consideremos el subconjunto ordinario $\mathcal{H} \subset [0, 1]^E$ formado por los subconjuntos borrosos $A \in [0, 1]^E$ tales que $A \leq (A \triangleleft R)$:

$$\mathcal{H} = \{A \in [0, 1]^E / A \leq (A \triangleleft R)\}$$

Y el subconjunto ordinario $\mathcal{T} \subset \mathcal{H}$ de los subconjuntos borrosos $B \in [0, 1]^E$ tales que $B = (B \triangleleft R)$:

$$\mathcal{T} = \{B \in [0, 1]^E / B = (B \triangleleft R)\}$$

Este último $\mathcal{T}$ puede ser vacío, pero $\mathcal{H} \neq \emptyset$, pues es evidente que $\emptyset \in \mathcal{H}$. Además, el conjunto de los “puntos borrosos” $\mathcal{S} = \{x_\alpha / x \in E, \alpha \neq 0\}$ tales que $x_\alpha(x) = \alpha$ y $x_\alpha(y) = 0$ si $y \neq x$, está contenido en $\mathcal{H}$, ya que $(x_\alpha \triangleleft R)(y) = \alpha \rightsquigarrow R(x, y)$, y como $R(x, x) = 1$, se obtiene:

$$x_\alpha \leq (x_\alpha \triangleleft R) \quad \forall x \in E \quad \forall \alpha \in (0, 1].$$

Proposición 4. (i) El subconjunto parcialmente ordenado $(\mathcal{H}, \leq)$ es un semiideal de $([0, 1]^E, \leq)$.

(ii) Si $R = R^{op}$, el semiideal $(\mathcal{H}, \leq)$ es inductivo y en consecuencia, por el axioma de Zorn [10]: el subconjunto $MAXIMAL(\mathcal{H})$ de elementos maximales de $(\mathcal{H}, \leq)$ es no vacío.

(iii) El conjunto $\mathcal{T}$ es un subconjunto de $MAXIMAL(\mathcal{H})$.

Demostración.

(i) Sea $A \in \mathcal{H}$ y sea $D \leq A$. Entonces se verifica que $(D \triangleleft R) \geq (A \triangleleft R) \geq A \geq D$, lo que prueba que $D \in \mathcal{H}$. (ii) Sea $(\mathcal{C}, \leq)$ una cadena incluida en $(\mathcal{H}, \leq)$. Hay que demostrar que $\mathcal{C}$ está acotada en $\mathcal{H}$. En primer lugar, demostremos que si C y D son elementos de la cadena $\mathcal{C}$ entonces $C \leq (D \triangleleft R)$. En efecto, si $C \leq D$ entonces $C \leq D \leq (D \triangleleft R)$. Si $D \leq C$ entonces $D \leq C \leq (C \triangleleft R) \leq (D \triangleleft R)$. De este resultado se deduce que

$$\bigvee \mathcal{C} \leq (D \triangleleft R) \quad \forall D \in \mathcal{C},$$

es decir, $\bigvee \mathcal{C} \leq \bigwedge_{D \in \mathcal{C}} (D \triangleleft R)$. Finalmente, como $R = R^{op}$, por la Proposición 3, se verifica

$$\bigwedge_{D \in \mathcal{C}} (D \triangleleft R) = (\bigvee \mathcal{C}) \triangleleft R,$$

luego obtenemos que $\bigvee \mathcal{C} \leq (\bigvee \mathcal{C}) \triangleleft R$, que prueba que $\bigvee \mathcal{C} \in \mathcal{H}$ y que, evidentemente, es un mayorante de $\mathcal{C}$. Esto demuestra que $(\mathcal{H}, \leq)$ es inductivo y, en consecuencia, $MAXIMAL(\mathcal{H}) \neq \emptyset$ según el lema de Zorn [10]. (iii) Si $\mathcal{T} = \emptyset$, evidente. Supongamos que $B \in \mathcal{T}$ y sea $A \in \mathcal{H}$ tal que $B \leq A$, entonces, de $A \geq B = (B \triangleleft R) \geq ((A \triangleleft R) \geq A)$ se deduce que $A = B$. Es decir, $B \in MAXIMAL(\mathcal{H})$. □

Si R es una tolerancia borrosa en E , la inclusión $\mathcal{T} \subseteq MAXIMAL(\mathcal{H})$ se convierte en igualdad:

Proposición 5. Si R es tolerancia borrosa, entonces $\mathcal{T} = MAXIMAL(\mathcal{H})$ y, en consecuencia, $\mathcal{T} \neq \emptyset$.

Demostración.

Sabemos que se verifica la inclusión $\mathcal{T} \subseteq MAXIMAL(\mathcal{H})$. Demostremos la inclusión en el otro sentido. Sea $A \in MAXIMAL(\mathcal{H})$. Entonces $A \leq (A \triangleleft R)$. Supongamos que $A < (A \triangleleft R)$. Entonces, existirá al menos un $w \in E$ tal que $A(w) < (A \triangleleft R)(w)$. Sea $A_w \in [0, 1]^E$ definido por $A_w(x) = A(x)$, si $x \neq w$, y $A_w(w) = (A \triangleleft R)(w)$. Se verifica: $A < A_w = A \vee w_{(A \triangleleft R)(w)}$, donde $w_{(A \triangleleft R)(w)}(x) = 0$ si $x \neq w$ y $w_{(A \triangleleft R)(w)}(w) = (A \triangleleft R)(w)$. Entonces,

$$\begin{aligned} A_w \triangleleft R &= (A \vee w_{(A \triangleleft R)(w)}) \triangleleft R = \\ &= (A \triangleleft R) \wedge (w_{(A \triangleleft R)(w)} \triangleleft R). \end{aligned}$$

Demostremos que $A_w \leq A_w \triangleleft R$. Para $x \neq w$, teniendo en cuenta que $\alpha \leq (\alpha \rightsquigarrow \beta) \rightsquigarrow \beta$ que aparece demostrado en [3] y que $R(x, w) = R(w, x)$, se verifica:

$$\begin{aligned} (A_w \triangleleft R)(x) &= (A \triangleleft R)(x) \wedge (w_{(A \triangleleft R)(w)} \triangleleft R)(x) = \\ &= (A \triangleleft R)(x) \wedge ((A \triangleleft R)(w) \rightsquigarrow R(w, x)) = \\ &= (A \triangleleft R)(x) \wedge (\inf_{z \in E} \{A(z) \rightsquigarrow R(z, w)\} \rightsquigarrow R(w, x)) \geq \\ &= (A \triangleleft R)(x) \wedge ((A(x) \rightsquigarrow R(x, w)) \rightsquigarrow R(w, x)) = \\ &= (A \triangleleft R)(x) \wedge ((A(x) \rightsquigarrow R(w, x)) \rightsquigarrow R(w, x)) \geq \\ &= (A \triangleleft R)(x) \wedge A(x) = A(x) = A_w(x). \end{aligned}$$

Para $x = w$, teniendo en cuenta que $R(w, w) = 1$:

$$\begin{aligned} (A_w \triangleleft R)(w) &= \inf_{z \in E} \{A_w(z) \rightsquigarrow R(z, w)\} = \\ &= \inf_{z=w} \{A_w(z) \rightsquigarrow R(z, w)\} \wedge (A_w(w) \rightsquigarrow R(w, w)) = \\ &= \inf_{z=w} \{A(z) \rightsquigarrow R(z, w)\} \wedge 1 \geq \inf_{z \in E} \{A(z) \rightsquigarrow R(z, w)\} = \\ &= (A \triangleleft R)(w) = A_w(w) \end{aligned}$$

Luego $A < A_w \leq (A_w \triangleleft R)$, es decir, $A < A_w \in \mathcal{H}$, lo que es absurdo, pues A es maximal de $\mathcal{H}$. En consecuencia, $A = A \triangleleft R$, que prueba que $A \in \mathcal{T}$. $\square$

Proposición 6. Si $A \in \mathcal{H}$ entonces $((A \triangleleft R) \triangleleft R) \in \mathcal{H}$.

Demostración.

Sea $A \in \mathcal{H}$, es decir tal que $A \leq A \triangleleft R$. Hay que demostrar que también se verifica:

$$((A \triangleleft R) \triangleleft R) \leq (((A \triangleleft R) \triangleleft R) \triangleleft R).$$

En efecto, por ser $\triangleleft$ antitono en el primer argumento: $(A \leq A \triangleleft R) \implies ((A \triangleleft R) \geq (((A \triangleleft R) \triangleleft R))) \implies ((A \triangleleft R) \triangleleft R) \leq (((A \triangleleft R) \triangleleft R) \triangleleft R)$. $\square$

En general, la ecuación $X \triangleleft R = X$ no siempre tiene solución, como se puede comprobar con el siguiente ejemplo:

Sea $E = \{x, y\}$ y R , la relación borrosa de la tabla:

R	x	y
x	0,1	0,3
y	0,9	0

que no es una tolerancia.

Si se considera en $[0, 1]$ la implicación residuada de Lukasiewicz, se puede demostrar que la única solución de $X \triangleleft R = X$ es el subconjunto borroso A tal que $A(x) = 0,55$ y $A(y) = 0,5$. Sin embargo, se comprueba que con la implicación residuada de Gödel, la ecuación anterior no tiene solución.

De la proposición anterior y de la equivalencia $((A \tilde{\times} A) \leq R) \Leftrightarrow (A \leq (A \triangleleft R))$, se deduce el siguiente corolario que asegura la existencia de soluciones cuando R es reflexiva y simétrica.

Corolario 2. Si R es una tolerancia borrosa, la ecuación $X \triangleleft R = X$ tiene solución, y el conjunto de soluciones coincide con el conjunto de borrosos maximales $B \in [0, 1]^E$ tales que $B \tilde{\times} B$ es un “ $\otimes$ -cuadrado” incluido en la tolerancia borrosa R .

Los elementos de $\mathcal{T} = \{B / B = (B \triangleleft R)\}$ constituyen un cubrimiento de E :

Proposición 7. Si R es una tolerancia borrosa, entonces $E = \bigvee_{B \in \mathcal{T}} B$.

Demostración.

Se verifica: $E = \bigvee_{x \in E} x_1$.

Por otra parte, $x_1 \leq (x_1 \triangleleft R)$, ya que $(x_1 \triangleleft R)(y) = R(x, y)$ y $x_1(y) = 0 \leq R(x, y)$ si $y \neq x$ y $x_1(x) = 1 = R(x, x) = (x_1 \triangleleft R)(x)$.

En consecuencia, $((x_1 \triangleleft R) \triangleleft R) \in \mathcal{H}$, por lo que existirá un maximal $B \in \text{MAXIMAL}(\mathcal{H}) = \mathcal{T}$ tal que $x_1 \leq ((x_1 \triangleleft R) \triangleleft R) \leq B$, luego $E = \bigvee_{x \in E} x_1 \leq \bigvee_{B \in \mathcal{T}} B$. $\square$

3. DEFINICIÓN DE $\otimes$ -BLOQUES. CÁLCULO DE ÉSTOS UTILIZANDO LA TEORÍA DE CONCEPTOS BORROSOS

En [4], los autores definen las clases de tolerancia como subconjuntos borrosos B que son normales, ($B(x) = 1$ para algún $x \in E$), y maximales dentro de un cierto conjunto ordenado.

En esta sección, ampliamos las definiciones y los resultados de estos autores [4], ya que exigimos que esos subconjuntos borrosos B tengan el mismo carácter maximal, pero no han de ser necesariamente normales.

Por otra parte, (de acuerdo con la nomenclatura usual en espacios de tolerancias ordinarios [2]), utilizamos el nombre de bloque en lugar de clase para nombrar esos maximales, reservando el nombre de clase para los subconjuntos borrosos $[x]_R \in [0, 1]^E$ definidos por: $[x]_R(y) = R(x, y) \forall y \in E, \forall x \in E$ (Véase por ejemplo [12]).

Finalmente (al igual que se hace en [4]), mostramos su relación con la teoría de Análisis de Conceptos Borrosos ([6],[7], [3])

Consideremos el sistema algebraico (E, R) formado por un referencial con cardinal $|E| \geq 2$ y por una tolerancia borrosa $R \in [0, 1]^{E \times E}$.

Definición 2. Llamaremos $\otimes$ -prebloque A de (E, R) a todo subconjunto borroso $A \in [0, 1]^E$ tal que $A \tilde{\times} A \leq R$. Llamaremos $\otimes$ -bloque de (E, R) a todo maximal B del conjunto de $\otimes$ -prebloques.

De los resultados expuestos en la sección anterior se desprenden estos otros :

1. Los $\otimes$ -bloques asociados a (E, R) serán todos los subconjuntos borrosos $B \in [0, 1]^E$ tales que $B = B \triangleleft R$.
2. Esta definición de $\otimes$ -bloque es coherente, ya que, como se ha demostrado en la sección anterior, está asegurada la existencia de esos maximales.
3. Se puede interpretar el cálculo de $\otimes$ -bloques como el del cálculo de “ $\otimes$ -cuadrados maximales”.
4. El referencial E es unión borrosa de los $\otimes$ -bloques:

$$E = \bigvee_{B \in \mathcal{T}} B$$

Veamos finalmente que con la Teoría de Conceptos Borrosos podemos abordar el cálculo de $\otimes$ -bloques. Para ello interpretamos el sistema (E, R) como el contexto borroso $([0, 1], E, E, R)$ en el que el referencial ordinario E representa tanto el conjunto de *objetos* como el de *atributos* [1] y en el que la relación de *incidencia* es la tolerancia borrosa $R \in [0, 1]^{E \times E}$.

En la teoría de conceptos borrosos, se puede demostrar que un concepto del contexto $([0, 1], E, E, R)$ es un par $(M, N) \in [0, 1]^E \times [0, 1]^E$ tal que $M = (M \triangleleft R) \triangleleft R^{op}$ y $N = M \triangleleft R$. Esta interpretación de los conceptos proporciona una conexión entre la teoría de conceptos borrosos y el cálculo de $\otimes$ -bloques en el caso en que se utilice una implicación residuada y una tolerancia borrosa, tal como se demuestra en la siguiente

Proposición 8. *Si R es una tolerancia borrosa en E , entonces los $\otimes$ -bloques B son las extensiones (o las intensiones) de los conceptos borrosos (B, B) cuyas dos componentes son iguales.*

Demostración.

Se verifica que $R = R^{op}$. Supongamos que B es un $\otimes$ -bloque. Entonces $B = B \triangleleft R$ y, en consecuencia, $B = B \triangleleft R = (B \triangleleft R) \triangleleft R = (B \triangleleft R) \triangleleft R^{op}$, que demuestra que (B, B) es un concepto del contexto $([0, 1], E, E, R)$. Por otra parte, si (M, M) es un concepto, entonces $M = M \triangleleft R$, es decir, M es un $\otimes$ -bloque. $\square$

Agradecimientos

Trabajo parcialmente soportado por el proyecto de investigación 2008/2301 del Gobierno de Navarra y por el Grupo de Investigación de la Universidad del País Vasco *Sistemas Inteligentes y Energía* (Proyecto GIU 07/45).

Referencias

- [1] C. Alcalde A. Burusco, R. Fuentes-González. Contextos con conjuntos de objetos y atributos iguales. *ESTYLF08*, (Mieres - Langreo), 2008.
- [2] W. Bartol, J. Miró, K. Pióro, F. Rosselló, On the coverings by tolerance classes, *Information Sciences* 166, Pág. 193-211, 2004
- [3] R. Bělohlávek. Fuzzy relational Systems. *IFSR International Series on Systems Science and Engineering*, No.20, 2002.
- [4] R. Bělohlávek, T. Funioková, Similarity and Fuzzy Tolerance Spaces, *J. Logic Computat*, Vol 14 No 6, Pág. 828-855, 2004
- [5] T.S. Blyth, M. F. Janowitz. *Residuation Theory*. Pergamon Press, 1972.
- [6] A. Burusco, R. Fuentes-González. The Study of the L-Fuzzy Concept Lattice. *Mathware and Computing*, 1, No.3, Pág.209-218, 1994.
- [7] A. Burusco, R. Fuentes-González. Construction of the L-Fuzzy Concept Lattice. *Fuzzy Sets and Systems*, 97, No.1, Pág.109-114, 1998.
- [8] M. Das, M.K. Chakraborty, T.K. Ghoshal. Fuzzy tolerance relation, fuzzy tolerance space and basis. *Fuzzy Sets and Systems*, 97, Pág.361-369, 1998.
- [9] P. Doherty, W. Lukaszewicz, A. Szalas. Tolerance spaces and approximative representational structures. *Lect. Notes in Computer Sciences*, 2821, Pág.475-489, 2003.
- [10] P. Dubreil, M.L. Dubreil-Jacotin. *Lecciones de álgebra moderna*. Ed. Reverté, 1965.
- [11] B. Ganter, R. Wille, *Formal Concept Analysis*, Springer Verlag, 1999.
- [12] G.J. Klir, B. Yuan. *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, 1995.
- [13] J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. On multi-adjoint concept lattices: definition and representation theorem. *Lect. Notes in Artificial Intelligence*, 4390, Pág.197-209, 2007.
- [14] J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. Formal concept analysis via multi-adjoint concept lattices. *Fuzzy Sets and Systems*, 160(2), Pág.130-144, 2009.
- [15] T.J. Ross. *Fuzzy Logic With Engineering Applications*. McGraw-Hill, Inc., 1995.
- [16] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. *Ordered Sets*, Reidel, Dordrecht-Boston, Pág.445-470, 1982.

On the Notion of Coherence in Fuzzy Answer Set Semantics*

Nicolás Madrid Labrador, Manuel Ojeda-Aciego

Dept. Matemática Aplicada. Universidad de Málaga

{nmadrid, aciego}@ctima.uma.es

Abstract

The notion of coherence, introduced in the context of fuzzy answer set programming (FASP) [6], provides a metalogic condition on the obtained models in FASP. In this work, we relate it with the concept of N -contradiction which is used in the definition of antonyms.

1 Introduction

Answer set semantics is an intuitive and elegant generalisation of the stable model semantics which provides a powerful solution for knowledge representation and non-monotonic reasoning problems. Its applicability has been widely manifested by a lot of researches in different areas.

Originally, they were intended to deal with two negations, one strong and one default negation. The use of these two types of negation is advocated in many contexts of interest, in particular in [10] their use is justified in relation to web rules.

In [6], in order to generalize the answer set semantics to general residuated logic programs, a generalization of consistency was proposed, it was called coherence in order to distinguish it from other generalizations in the fuzzy framework.

In this approach, we provide additional insight and motivation in favour of the notion of coherence instead of other generalizations of consistency. We show that non-coherent interpretations contradict the negation meta-rule by an excess of information and we relate the use of the strong negated propositional symbols with computing with N -contradictory concepts.

* Partially supported by the Spanish Science Ministry by project TIN2006-15455-C03-01 and by Junta de Andalucía project P06-FQM-02049.

In this paper, we start by briefly recalling the preliminary definitions needed in order to introduce coherent interpretations in the framework of general residuated logic programs. Then we further motivate the coherent condition as generalization of consistency and the relationship with the concept of N -contradictory L fuzzy sets studied in [8].

2 Preliminary definitions

As stated in the introduction, the notion of coherence was introduced in the framework of general residuated logic programming, a generalization of logic programming in which the underlying mathematical basis is that of residuated lattices. Let us start this section recalling the definition of residuated lattice:

Definition 1 A residuated lattice is a triple $\mathcal{L} = ((L, \leq), *, \leftarrow)$ such that:

1. $(L, \leq)$ is a complete and bounded lattice with largest element 1 and least element 0.
2. $(L, *, 1)$ is a commutative monoid unit element 1.
3. $*$ and $\leftarrow$ form an adjoint pair, i.e:

$$z \leq (x \leftarrow y) \text{ iff } y * z \leq x \quad \text{for all } x, y, z \in L.$$

In the rest of the paper we will consider a residuated lattice enriched with two negation operators, $(L, *, \leftarrow, \sim, \neg)$. The two negations will modelize the notions of strong negation $\sim$ and default negation $\neg$ often used in logic programming. As usual, a negation operator, over $\mathcal{L}$, is any decreasing mapping $n: L \rightarrow L$ satisfying $n(0) = 1$ and $n(1) = 0$.

The difference between strong and default negation in our context is essentially semantical, and relates to the method we use to infer the truth value of one negated propositional symbol.

The syntax of general residuated logic programs assumes a set Π of propositional symbols. If $p \in \Pi$, then

both p and $\sim p$ are called *literals*. We will denote arbitrary literals with the symbol ℓ (possible subscripted), and the set of all literals as *Lit*.

Definition 2 Given a residuated lattice with two negations $(L, *, \leftarrow, \sim, \neg)$, a general residuated logic program $\mathbb{P}$ is a set of weighted rules of the form

$$\langle \ell \leftarrow \ell_1 * \dots * \ell_m * \neg \ell_{m+1} * \dots * \neg \ell_n; \quad \vartheta \rangle$$

where ϑ is an element of L and $\ell, \ell_1, \dots, \ell_n$ are literals.

Definition 3 A fuzzy L -interpretation is a mapping $I: \text{Lit} \rightarrow L$; that is, an L -fuzzy subset of literals.

A fuzzy L -interpretation can be lifted to the domain of all body rule as:

$$\begin{aligned} I(\ell_1 * \dots * \ell_m * \neg \ell_{m+1} * \dots * \neg \ell_n) = \\ = I(\ell_1) * \dots * I(\ell_m) * \neg I(\ell_{m+1}) * \dots * \neg I(\ell_n) \end{aligned}$$

Notice that we denote the syntax operators and the corresponding operators in the residuated lattice with the same symbol.

We say that I satisfies a rule $\langle \ell \leftarrow \mathcal{B}; \quad \vartheta \rangle$ if and only if $I(\mathcal{B}) * \vartheta \leq I(\ell)$ or, equivalently, $\vartheta \leq I(\ell \leftarrow \mathcal{B})$.

Finally, I is a model of $\mathbb{P}$ if it satisfies all rules (and facts) in $\mathbb{P}$.

The fuzzy answer set semantics described in [6, 7] is defined in two steps. Firstly, a treatment of strong negation in the context of residuated logic programming is provided in terms of the notion of coherence as a generalization in the fuzzy framework of the concept of consistence. Then, fuzzy answer sets for general residuated logic programs are defined as a suitable generalization of the *Gelfond-Lifschitz* reduct.

As our interest in this work is on the notion of coherence, our natural environment is that of extended residuated logic programs, that is, those which do not contain default negation.

Note that, as our interpretations are defined on the set of literals, every extended program has a least model which can be obtained, for instance, by iterating the immediate consequence operator, see [2]. However, one has to take into account the interaction between opposite literals. For example, in the classical case we reject the inconsistent models, i.e p and $\sim p$ cannot be true at the same time.

This notion of inconsistency is generalized to the fuzzy framework, where one can allow that two opposite literals live together ... under some requirements.

Definition 4 A fuzzy L -interpretation I over *Lit* is coherent if the inequality $I(\sim p) \leq \sim I(p)$ holds for every propositional symbol p .

Now, the key definition in extended residuated logic programs is the following:

Definition 5 Let $\mathbb{P}$ be an extended residuated logic program, we say that $\mathbb{P}$ is coherent if its least model is coherent.

Now, a natural question arises: Why coherent interpretations provide an interesting generalization? We have three main reasons. Firstly, it is easy to implement, because it only depends on the negation operator; secondly, it allows to formalise lack of knowledge. For example I such that $I(\ell) = 0$ for all $\ell \in \text{Lit}$ is always coherent. And finally, coherence coincides with consistency in the classical framework.

In the next section we present additional motivations on the notion of coherence and we compare it with another approach in the fuzzy framework: the N -contradictory fuzzy sets.

3 On coherence

In the previous section, the notions of coherent interpretation and coherent program were introduced. Here, we focus on widely explaining the concept of coherence. Specifically, we start by recalling some properties of coherent interpretations, then we present additional motivations to consider the coherence as an adequate generalization of consistency and, finally, we relate it with other approaches.

3.1 Recalling some properties of coherence

Although the definition of coherent program might look a hard restriction, the following property of coherent interpretations shows that a program is coherent if and only if it has, at least, one coherent model.

Proposition 1 Let I and J be two interpretations satisfying $I \leq J$. If J is coherent, then I is coherent.

Proof Let p be a propositional symbol. By using $I \leq J$, coherence of J and the decreasing property of $\sim$ we have $I(\sim p) \leq J(\sim p) \leq \sim J(p) \leq \sim I(p)$ $\square$

Corollary 1 An extended residuated logic program is coherent if and only if it has at least one coherent model.

Proposition 1 and its contrapositive, which states that one cannot recover coherence by adding knowledge, play an important role in the usefulness of coherent interpretations.

In order to continue with some properties of the notion of coherence, take into account that an interpretation

I assigns a truth degree to any negative literal $\sim p$ independently from the negation operator. This way, if we have two different negation operators ($\sim_1$ and $\sim_2$) we can talk about the coherence of I wrt any of these operators.

Proposition 2 *Let $\sim_1$ and $\sim_2$ be two negation operators such that $\sim_1 \leq \sim_2$, then any interpretation I that is coherent wrt $\sim_1$ is coherent wrt $\sim_2$.*

Proof Let p be a propositional symbol, then $I(\sim p) \leq \sim_1 I(p) \leq \sim_2 I(p)$. $\square$

The following example shows us the importance in the negation operator selected as strong negation to determine the coherence of a residuated logic program.

Example Consider the lattice $[0, 1]$ with its usual order the Gödel connectives:

$$x * y = \min(x, y)$$

$$x \leftarrow y = \begin{cases} 1 & \text{if } x \geq y \\ x & \text{else} \end{cases}$$

and the following program $\mathbb{P}$:

$$\begin{aligned} r_1 : \langle p \leftarrow 1; \quad 1 \rangle \\ r_2 : \langle q \leftarrow p; \quad 0.8 \rangle \\ r_3 : \langle \sim q \leftarrow 1; \quad 0.7 \rangle \end{aligned}$$

The least model is $M = \{(p, 1); (q, 0.8); (\sim q, 0.7)\}$. If we consider the usual negation $n(x) = 1 - x$ to determine the coherence of the program we obtain that $\mathbb{P}$ is not coherent, and the least model semantics fails in this case. However, if we consider the negation:

$$\bar{n}(x) = \begin{cases} 1 & \text{if } x \leq 0.8 \\ 0 & \text{if } x > 0.8 \end{cases}$$

the program is coherent and the least model semantics provides a meaning to the program. $\square$

We define an ordering among extended residuated logic programs as follows: Let $\mathbb{P}_1$ and $\mathbb{P}_2$ be two extended programs, then $\mathbb{P}_1 \subseteq \mathbb{P}_2$ if and only if for each rule $\langle r_i; \vartheta_1 \rangle$ in $\mathbb{P}_1$ there exists another rule¹ $\langle r_i; \vartheta_2 \rangle$ in $\mathbb{P}_2$ such that $\vartheta_1 \leq \vartheta_2$.

Proposition 3 *Let $\mathbb{P}_1 \subseteq \mathbb{P}_2$ be two extended programs then the least model of $\mathbb{P}_1$ is smaller than the least model of $\mathbb{P}_2$.*

Proof Follows by the monotonicity properties of the conjunctive $*$ and the supremum. $\square$

¹Note that the only difference between both rules is the assigned weight.

Therefore we can say that the greater a program is the more information it provides. Now it is easy to prove that the coherence is also decreasingly conserved under the order of extended residuated logic programs.

Corollary 2 *Let $\mathbb{P}_1 \subseteq \mathbb{P}_2$ be two extended programs. If $\mathbb{P}_2$ is coherent then $\mathbb{P}_1$ is coherent as well.*

3.2 Motivation for the condition of coherence

There are many ideas underlying the concept of inconsistency: conflicting inference, inferring contradiction formulas, lack of models, etc. Coherence focuses on the idea of excess of information, which leads to a conflict with the negation meta-rule. Let n be a negation operator, the negation meta-rule is defined as follows: “if p has truth value ϑ then $n(p)$ has assigned the truth value $n(\vartheta)$ ”. Contradicting the negation meta-rule by excess of information means that the program rules infer more information for a propositional symbol p than it could be inferred using the negation meta-rule. Let us see, using the properties described in the last section, that non-coherent interpretations are affected by this excess of information.

Notice first that a coherent L -interpretation never contradicts the negation meta-rule (wrt $\sim$) by excess of information. Given a coherent interpretation I , if I contradicts the negation metarule (wrt $\sim$) it is easy to prove that every interpretation less than or equal to I contradicts this meta-rule as well. Contrariwise, if we consider the following L -interpretation:

$$\bar{I}(\ell) = \begin{cases} I(p) & \text{if } \ell \text{ is the propositional symbol } p \\ \sim I(p) & \text{if } \ell \text{ is the literal symbol } \sim p \end{cases}$$

we obtain a non-contradictory interpretation including more information in I . Therefore a coherent interpretation either satisfies the negation meta-rule or contradicts it by lack of information. According to the explanation described above, considering coherent interpretation seems a convenient option to avoid a contradiction with the negation metarule by excess of information.

When I is a non-coherent L -interpretation, then there exists a propositional symbol p such that either $I(\sim p) > \sim I(p)$ or $I(\sim p)$ and $\sim I(p)$ are incomparable elements in L . In both cases a non-coherent interpretation implies a contradiction with the negation meta-rule. Is this contradiction given by an excess of information? Certainly. The contrapositive of Proposition 1 tells us that by adding information to I we will never obtain a coherent interpretation, thus the lack of information is not the reason of this contradiction. Therefore the only possibility to obtain a non

contradictory² interpretation is by removing information. Thus if we obtain an incoherent interpretation as least fixpoint of a logic program, it has been due to an excess of information in the program (possibly erroneous information). As a result, rejecting non-coherent interpretations seems convenient as well.

An important remark is that coherence can be interpreted with an empirical sense and that the strong negation operator can (and should) be fixed at the beginning. More precisely, when the propositional symbol are fixed, each one has an empirical motivation, i.e a definition given in a natural language. At this point, the programmer has to determine what the strong negated propositional symbols mean. A possibility is described in the next section and is related to computing with antonyms. This way, $\sim p$ could symbolize an antonym of p and the negation operator used represents the degree of contradiction between p and $\sim p$ (see section 3.4). Therefore the choice of the strong negation operator is determined by empirical motivation and not for technical stuff.

3.3 Coherence and α -consistence

Of course this is not the only generalization of consistency in the fuzzy framework. In several approaches is usual to find the following, and popular, generalization.

Definition 6 *Let $*$ be a t-norm and $\sim$ a negation operator. We say that an interpretation $I: Lit \rightarrow L$ on the set of literals is α -consistent if for all propositional symbol p we have that $I(p) * I(\sim p) \leq \alpha$.*

Note that, by the adjoint condition, $I(p) * I(\sim p) \leq \alpha$ iff $I(\sim p) \leq \alpha \leftarrow I(p)$. In other words, α -consistence provides an upper bound to the value of $I(\sim p)$ in terms of $I(p)$ and the parameter α . On its turn, recall that a coherent interpretation directly provides such an upper bound, namely $\sim I(p)$, which depends only on the operator intended to interpret the strong negation. Hence both concepts determine similar constraints in order to consider accepted interpretations, but they are not equivalent. The differences between them are actually crucial to prefer coherence instead of α -consistence as a suitable generalization of consistency. We start by providing a relationship between 0-consistent interpretations and coherent interpretations.

Proposition 4 *Let I be an L -interpretation and let $(*, \leftarrow)$ be an adjoint pair. Then I is 0-consistent wrt $*$ if and only if I is coherent wrt $n(x) = 0 \leftarrow x$.*

²with respect to the negation meta-rule

Proof

$$I(p) * (\sim p) \leq 0 \quad \text{if and only if} \quad I(\sim p) \leq 0 \leftarrow I(p)$$

by the adjoint property of $*$ and $\leftarrow$. $\square$

Notice that the condition “ $(*, \leftarrow)$ is an adjoint pair” is not a restriction since for all t-norm $*$ we can define an implication operator $\leftarrow_R$ such that $(*, \leftarrow_R)$ forms an adjoint pair (see [5]).

The statement in Proposition cannot be extended for $\alpha \neq 0$: consider an L -interpretation I which assigns to p the value 1 and to $\sim p$ the value α is always α -consistent. However it is impossible to find a negation operator N such that I were coherent wrt N .

Another feature of α -consistent interpretations is the symmetry: the value of p bounds the value of $\sim p$ as the value of $\sim p$ bounds the value of p . That feature can be simulated also in coherent interpretations if the chosen negation is bijective. However, in some cases, it is convenient to allow for an asymmetric behaviour between opposite literals. For example, if p denotes “the container is full” and $\sim p$ denotes “we can introduce stuff in the container” asymmetry is required. If the value assigned to p is 1 then the container is completely full and we can introduce nothing in it, therefore the value 1 for p bounds the value of $\sim p$ by 0. However if we assign to $\sim p$ the value 1 the container could be half full. Hence the bound 0 for p , in order to preserve symmetry, is highly restrictive in this case.

3.4 Coherence and N -contradiction

The concept of N -contradiction is defined over fuzzy sets in [8] and is closely related to computing with antonyms [1, 9]; in fact, one condition for (A, B) to be an antonym pair is that A is N -contradictory with B .

The aim of this section is to connect coherence and strong negation with the concept of N -contradiction and, consequently, with computing with antonyms.

Recall that, given a lattice L , an L -fuzzy set A defined over the universe $X \neq \emptyset$ is a set $A = \{(x, \mu_A(x)) : x \in X\}$ such that $\mu_A(x) \in L$ for all $x \in X$. The function μ_A is called the membership function of A and usually an L -fuzzy set is denoted directly by its membership function. A fuzzy answer set is an L -fuzzy answer set where L is the unit real interval $[0, 1]$.

The following definition describes when an L -fuzzy set is N -contradictory with respect to another L -fuzzy set.

Definition 7 *Let N be a negation operator, an L -fuzzy set A is N -contradictory with respect to the L -fuzzy set B if and only if $A(x) \leq N(B(x))$ for all element x in the universe*

The following lemma shows that the negation operator used to establish the N -contradiction also represents a grade of contradiction:

Lemma 1 *Let N_1 and N_2 be two negations operator such that $N_1(x) \leq N_2(x)$ for all $x \in L$. Let A and B be two L -fuzzy sets. If A is N_1 -contradictory wrt B then A is N_2 -contradictory wrt B*

Proof As A is N_1 -contradictory wrt B then $A(x) \leq N_1(B(x))$ for all $x \in L$. Using the hypothesis $N_1 \leq N_2$, we obtain the inequality: $A(x) \leq N_1(B(x)) \leq N_2(B(x))$ for all $x \in L$.

□

As we said above, the negation operator in Definition 7 indicates a grade of contradiction between both L -fuzzy sets. For example, consider the greatest negation operator:

$$N_{\top}(x) = \begin{cases} 0 & \text{if } x = 1 \\ 1 & \text{if } x < 1 \end{cases}$$

Then, an L -fuzzy set is N -contradictory wrt another L -fuzzy set (N is a non fixed negation operator) if and only if is $N_{\top}$ contradictory. That means that $N_{\top}$ determines the lowest level of contradiction. On the other hand, the least negation operator:

$$N_{\perp}(x) = \begin{cases} 0 & \text{if } x \neq 1 \\ 1 & \text{if } x = 1 \end{cases}$$

determines the biggest level of contradiction in the sense that if an L -fuzzy set is $N_{\perp}$ -contradictory wrt another L -fuzzy set then is N -contradictory for all negation operator N . This last level of contradiction establishes that an element in the universe cannot has a positive value in both L -fuzzy sets.

Let us clarify the concept of N -contradiction through an example:

Example Consider the following fuzzy set defined over the interval $[0, 100]$:

$$CloseTo20(x) = \begin{cases} 0 & \text{if } x \leq 10 \\ \frac{x-10}{5} & \text{if } 10 \leq x \leq 15 \\ 1 & \text{if } 15 \leq x \leq 25 \\ \frac{30-x}{5} & \text{if } 25 \leq x \leq 30 \\ 0 & \text{if } x \geq 30 \end{cases}$$

which represents the numbers of $[0, 100]$ which are close to the number 20. It is not difficult to obtain N -contradictory sets wrt $CloseTo20$. A $N_{\perp}$ -contradictory set wrt $CloseTo20$ is $CloseTo100$:

$$CloseTo100(x) = \begin{cases} 0 & \text{if } x \leq 90 \\ \frac{x-90}{5} & \text{if } 90 \leq x \leq 95 \\ 1 & \text{if } x \geq 95 \end{cases}$$

Moreover, is possible to find L -fuzzy sets which are also contradictories with $CloseTo20$ but in a lesser level. For example the following fuzzy set is $(1-x)$ -contradictory wrt $CloseTo20$:

$$CloseTo35(x) = \begin{cases} 0 & \text{if } x \leq 25 \\ \frac{x-25}{5} & \text{if } 25 \leq x \leq 30 \\ 1 & \text{if } 30 \leq x \leq 40 \\ \frac{45-x}{5} & \text{if } 40 \leq x \leq 45 \\ 0 & \text{if } x \geq 45 \end{cases}$$

□

To finish the section, we show that the strong negation and the coherence conditions are related with the concept of N -contradiction. Let A and B be two L -fuzzy sets. Assume that B is N -contradictory wrt A . Suppose also that the general residuated logic program $\mathbb{P}$ contains the following propositional symbols (together them meta-interpretations):

$p \equiv \text{The element } \delta \text{ belongs to } A$

$q \equiv \text{The element } \delta \text{ belongs to } B$

If we want to represent the N -contradictory relation between A and B for the element δ in $\mathbb{P}$, and then in the information inferred by it, we identify the strong negation operator with N and we include the following rule:

$$\langle \sim p \leftarrow q \quad ; \quad 1 \rangle$$

Observe that all fuzzy answer set of this program (with the above rule includes in it) holds $I(q) \leq I(\sim p) \leq \sim I(p) = N(I(p))$. Hence the information inferred by the program holds the relation of N -contradictory between A and B for the element δ .

The same representation of the notion of N -contradiction can be done by identifying q with $\sim p$. Each choice has good and bad features. Identifying q with $\sim p$ reduces the number of literals and thus it reduces the computational complexity. However not doing it supplies the possibility to represent various N -contradictory L -fuzzy sets with respect to a given one.

4 Conclusion and future work

We have recalled the basic definition of answer set semantics for general residuated logic programs. We have focused on motivating the use of the notion of coherence as a generalization of consistency in the fuzzy framework. We have presented several reasons to consider coherent interpretations: they do not contradict the negation metarule by an excess of information and are useful to reasoning with N -contradictory concepts.

We have started with the first steps towards reasoning with antonyms in general residuated logic programs. Therefore, future work should go towards the complete introduction of antonyms in this framework.

References

- [1] E. Castiñeira, C. Torres-Blanc, and S. Cubillo. Un modelo para medir la N-contradicción de conjuntos borrosos. In *Proc of ESTYLF08*: 113–119, 2008.
- [2] C. V. Damásio and L. M. Pereira. Monotonic and residuated logic programs. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'01*, pages 748–759. Lect. Notes in Artificial Intelligence, 2143, 2001.
- [3] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP-88*, pages 1070–1080, 1988.
- [4] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
- [5] P. Hájek. *Metamathematics of Fuzzy Logic*. Trends in Logic. Kluwer Academic, 1998.
- [6] N. Madrid and M. Ojeda-Aciego. Towards a fuzzy answer set semantics for residuated logic programs. In *Proc of WI-IAT'08. Workshop on Fuzzy Logic in the Web*, pages 260–264, 2008.
- [7] N. Madrid and M. Ojeda-Aciego. On coherence and consistence in fuzzy answer set semantics for residuated logic programs. *Lect. Notes in Computer Science*, 5571:60–67, 2009.
- [8] E. Trillas, C. Alsina, and J. Jacas. On contradiction in fuzzy logic. *Soft Computing*, 3(4):197–199, 1999.
- [9] E. Trillas, C. Moraga, S. Guadarrama, S. Cubillo, and E. Castiñeira. Computing with antonyms. In *Forging New Frontiers: Fuzzy Pioneers I*, volume 217 of *Studies in Fuzziness and Soft Computing*, pages 133–153. Springer, 2007.
- [10] G. Wagner. Web rules need two kinds of negation. In *Principles and Practice of Semantic Web Reasoning*, volume 2901 of *Lecture Notes in Computer Science*, pages 33–50, 2003.

Grado de parentesco entre predicados *

Itziar García-Honrado¹ Enric Trillas¹ Sergio Guadarrama¹

¹ European Centre for Soft Computing

Mieres (Asturias), España.

{itziar.garcia, enric.trillas, sergio.guadarrama}@softcomputing.es

Resumen

Basándose en las ideas de L. Wittgenstein sobre el parecido de familia (en alemán *Familienähnlichkeit* y en inglés *Family Resemblance*), se presenta una nueva formalización matemática del parentesco entre predicados y sus usos, asignando a cada uso de un predicado un conjunto borroso. Para ello, se define el conjunto de los ‘parientes’ de cierto uso de un predicado, y se calculan los grados de parecido con sus parientes. Además, si entre dos conjunto borrosos que no sean parientes hay un grado de parecido alto, se definen los parientes de segundo orden y un grado de parentesco de segundo orden.

Palabras Clave: Predicado, Parecido de familia, Grado de parentesco, Indistinguibilidad.

1. INTRODUCCIÓN

Este artículo se inspira en [5] el trabajo previo, que constituye un primer acercamiento a la formalización del concepto de parecido de familia, ‘familienähnlichkeit’, también ‘family resemblance’, de Wittgenstein.

Wittgenstein es el iniciador de la conocida filosofía del lenguaje; en [7], recogió el carácter dinámico de las palabras y el lenguaje a través de los llamados juegos del lenguaje. Las palabras sufren modificaciones con el tiempo, motivadas por su uso en contextos distintos o por hacer referencia a nuevos elementos, como ejemplo, la palabra ‘indio’, que en un principio designaba

*Este trabajo ha sido financiado por la Fundación para el Progreso del Soft Computing (Asturias, España), y CICYT (España) bajo el proyecto TIN2008-06890-C02-01.

exclusivamente a los nativos de India, pero que tras el descubrimiento de América pasó a designar también a los americanos nativos.

En [4], se ve cómo un uso concreto de un predicado, puede representarse por medio de un conjunto borroso. Así que se intentará estudiar la relación de parentesco entre los conjuntos borrosos que puedan representar distintos usos de un mismo predicado, partiendo de un conjunto borroso que represente un uso concreto del predicado y que se tomará como patrón. Se definirá el conjunto de los parientes de ese patrón, y el grado de parentesco de los parientes con él. Más adelante, se extenderá el conjunto de parientes a parientes de segundo orden y se definirá también un grado de parentesco de segundo orden.

2. FAMILIAS DE CONJUNTOS BORROSOS CON PARENTESCO

Sobre un universo de discurso X , un predicado P ordena los elementos de X , si se puede decir que ‘ x es menos P que y ’, ($x \leq_P y$), con $x, y \in X$. Posteriormente, se puede construir un conjunto borroso, μ_P , que represente un uso del predicado, capturando dicho orden en la forma, ‘si $x \leq_P y$, entonces $\mu_P(x) \leq \mu_P(y)$ ’ (para más detalles ver [4]); es decir, que μ_P pueda considerarse un ‘grado’ para el predicado.

Para un predicado P se pueden construir diferentes grados, siempre que cada grado capture el orden que el predicado induce sobre el universo de discurso; por ejemplo el uso del predicado en diferentes contextos motivará una modificación en el grado (véase Figura 1).

Así pues, el predicado $P = \text{alto}$, definido en un universo de personas introduce un orden, gracias a la característica numérica ‘altura’; ‘una persona x es menos alta que otra y ’, si $\text{altura}(x) \leq \text{altura}(y)$, pero el grado en que se considera que una persona es alta depende del contexto. Por ejemplo no se considera que alguien

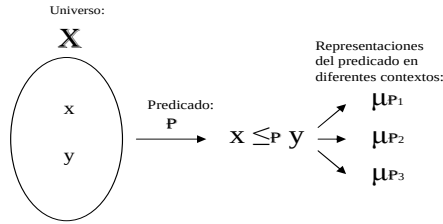


Figura 1: Representación de un mismo predicado en distintos contextos

de 1.70 metros tenga el mismo grado de alto en el contexto de una escuela de primaria, o en el contexto de un equipo de baloncesto profesional.

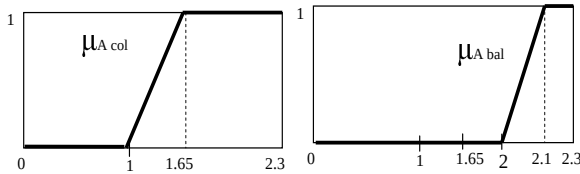


Figura 2: En una escuela Figura 3: En un equipo de baloncesto

El ejemplo muestra que el conjunto borroso que representa el predicado alto en un equipo de baloncesto, sufre una modificación con respecto a la representación del predicado alto en una escuela, en este caso porque las alturas de un colegio son más heterogéneas y casi sin alumnos con altura superior a 1.70; sin embargo, en un equipo de baloncesto casi todos los jugadores tendrán una altura superior a 1.90 e inferior a 2.10. No obstante, las dos funciones muestran cierto parecido, por lo menos en cuanto a su forma.

Los elementos que verifican el uso del predicado P con grado uno, son los *prototipos* de ese uso P , y se denota por $S(\mu_P)$ a su conjunto, $S(\mu_P) = \{x \in X; \mu(x) = 1\}$. Por analogía, los *anti-prototipos* serán los elementos que no verifican el predicado, que lo hacen con grado cero y se denotará su conjunto $Z(\mu_P) = \{x \in X; \mu(x) = 0\}$.

La relación de parentesco se define entre conjuntos borrosos μ, σ tales que $S(\mu) \neq \emptyset$, $Z(\mu) \neq \emptyset$, $S(\sigma) \neq \emptyset$ y $Z(\sigma) \neq \emptyset$, como sucede con las funciones de las figuras 2 y 3. Estos conjuntos borrosos se llaman *conjuntos borrosos completamente normalizados*.

Observación 2.1 En lo que sigue, se supondrá que todos los grados, $\mu_P^i (i \in I)$, de un mismo predicado P definido en el universo de discurso X , compartirán algunos prototipos y anti-prototipos. Esto es, $\bigcap_{i \in I} Z(\mu_P^i) \neq \emptyset$ y $\bigcap_{i \in I} S(\mu_P^i) \neq \emptyset$.

En lo sucesivo, manejaremos un conjunto, $F^*(X)$, de algunos conjuntos borrosos completamente normalizados, con $X \subset R$, al que pertenecerá el patrón de su familia, y sus parientes que se definirán más adelante.

Si tenemos un conjunto borroso no normalizado con máximo y mínimo, podemos transformarlo en uno completamente normalizado, bajo la transformación lineal

$$\mu^* = \frac{\mu - \min(\mu)}{\max(\mu) - \min(\mu)}.$$

Cabe considerar que dos conjuntos borrosos representan un mismo predicado si su comportamiento sigue ciertas conductas, como que compartan prototipos y anti-prototipos, y que crezcan o decrezcan a la vez (para que el conjunto borroso capture el orden que induce el predicado en el conjunto). Basándonos en esto se define la relación de parentesco ([5]), como,

Definición 2.2 Sea $X \subset R$ el universo de discurso, se define la relación de parentesco ('family resemblance') entre dos conjuntos borrosos μ, σ en $[0, 1]^X$, $\mathbf{fr} \subset [0, 1]^X \times [0, 1]^X$, por $(\mu, \sigma) \in \mathbf{fr}$ si y sólo si,

1. $Z(\mu) \cap Z(\sigma) \neq \emptyset$, $S(\mu) \cap S(\sigma) \neq \emptyset$
2. μ es creciente en $A \subset X$ si y sólo si σ es creciente en A .
3. μ es decreciente en $A \subset X$ si y sólo si σ es decreciente en A .

Observación 2.3 Se considerará que un conjunto borroso $\mu \in [0, 1]^X$ es creciente en $A \subset X$ si para cualquier $x, y \in A$ tales que $x \leq y$, se verifica que $\mu(x) \leq \mu(y)$.

Análogamente se considerará que $\mu \in [0, 1]^X$ es decreciente en $A \subset X$ si para cualquier $x, y \in A$ tales que $x \leq y$, se verifica que $\mu(y) \leq \mu(x)$.

Esta relación es una relación reflexiva y simétrica, pero no transitiva, como ocurre con algunas relaciones de familia entre personas. Si se considera que alguien es un familiar de otro si y sólo si hay lazos sanguíneos: "Juan es familia de su madre" "Juan y su primo por parte de padre son familia", sin embargo "el primo de Juan y la madre de Juan" no tendrían lazos sanguíneos y, por tanto, bajo esta relación no serían considerados familiares.

Veamos gráficamente que la relación $\mathbf{fr}$ no es transitiva. En la figura 4, se tiene que $(\mu, \delta) \in \mathbf{fr}$, $(\sigma, \delta) \in \mathbf{fr}$, pero $(\mu, \sigma) \notin \mathbf{fr}$, ya que μ y σ no comparten prototipos.

Siguiendo con el predicado "alto", de las figuras 2 y 3 se aprecia que ambos predicados mantienen relación

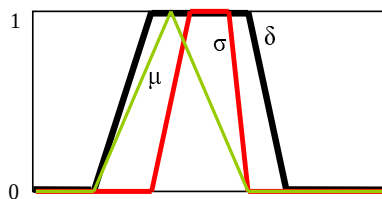


Figura 4:

de parentesco. Se verifica que, $Z(\mu_{A\text{ col}}) \cap Z(\mu_{A\text{ bal}}) = [0, 1] \neq \emptyset$, $S(\mu_{A\text{ col}}) \cap S(\mu_{A\text{ bal}}) = [2, 10, 2, 30] \neq \emptyset$ y además verifican las condiciones de monotonía.

Se definirá la familia de un predicado representado por el conjunto borroso μ dentro de un conjunto de conjuntos borrosos previamente seleccionado $F(X)^*$, como $[\mu] = \{\sigma \in F^*(X); (\mu, \sigma) \in \mathbf{fr}\}$. Estas familias no forman una partición del conjunto $F(X)^*$, ya que no son disjuntas dos a dos.

2.1. SINÓNIMOS

Una forma particular de sinonimia de predicados, que se recoge en [5], es la siguiente,

Definición 2.4 Dos predicados P y P^* definidos sobre $X \subset R$ y representados por los conjuntos borrosos μ_P y μ_{P^*} respectivamente, son sinónimos si existe una aplicación biyectiva $u : X \rightarrow X$, tal que

1. $\mu_{P^*} = \mu_P \circ u^{-1}$
2. Si $x \leq y$, entonces $u(x) \leq u(y)$
3. $u^{-1}(Z(\mu_P)) \subset Z(\mu_P)$
4. $u^{-1}(S(\mu_P)) \subset S(\mu_P)$

Por tanto para todo $\mu_P \in F^*(X)$, $X \subset R$, se tiene que $(\mu_P, \mu_{P^*}) = (\mu_P, \mu_P \circ u^{-1}) \in \mathbf{fr}$.

Los dos usos del predicado ‘alto’, $\mu_{A\text{ col}}$ y $\mu_{A\text{ bal}}$ presentados en las figuras 2 y 3, además de tener parecido de familia pueden considerarse sinónimos, ya que existe una biyección u , tal que $\mu_{A\text{ col}} = \mu_{A\text{ bal}} \circ u^{-1}$.

2.2. PREDICADO EN DISTINTOS UNIVERSOS DE DISCURSO

En esta sección se estudiarán los cambios que sufren los predicados al utilizarse en dos universos de discurso distintos, X e Y .

Por ejemplo, el predicado ‘simpático’ suele definirse sobre personas, pero podría migrar y utilizarse sobre

los objetos o animales y así decir que cierto objeto o animal es simpático.

Definición 2.5 Se dice que el uso de un predicado P definido sobre el universo X (representado por $\mu \in [0, 1]^X$) *migra* al predicado Q en el universo Y (representado por $\sigma \in [0, 1]^Y$) cuando existan una función biyectiva $\varphi : X \rightarrow Y$ y una función monótona $F : [0, 1] \rightarrow [0, 1]$ verificando,

1. Si $x \leq_P y$, entonces $\varphi(x) \leq_Q \varphi(y)$, para $x, y \in X$
2. $\varphi^{-1}(Z(\sigma)) \subset Z(\mu)$
3. $\varphi^{-1}(S(\sigma)) \subset S(\mu)$
4. $F(0) = 0$ y $F(1) = 1$
5. $\sigma = F \circ \mu \circ \varphi^{-1}$

De la definición se deduce que $(\mu, \sigma \circ \varphi) \in \mathbf{fr}$ y $(\sigma, \mu \circ \varphi^{-1}) \in \mathbf{fr}$.

Ejemplo 2.6 Sea el predicado $P = \text{bajo}$, definido sobre el conjunto de personas con altura en centímetros dentro del intervalo $[0, 200]$ y representado por el conjunto borroso $\mu_b(x) = 1 - \frac{x}{200}$. Sean $\varphi : [0, 200] \rightarrow [0, 1]$, definida por $\varphi(x) = \frac{x}{200}$ y $F = \text{id}$. Es:

$$F \circ \mu_b \circ \varphi^{-1}(y) = \mu_b(200y) = 1 - y, \quad \text{para } y \in [0, 1].$$

La función obtenida es el conjunto borroso que representa el predicado pequeño en el intervalo $[0, 1]$.

Si se utilizase $F(x) = x^2$, se obtendría $F \circ \mu_b \circ \varphi^{-1}(y) = (1 - y)^2 = y^2 - 2y + 1$, un uso diferente del predicado pequeño en $[0, 1]$.

3. INDISTINGUIBILIDAD PARA MEDIR EL GRADO DE PARENTESCO ENTRE CONJUNTOS BORROSOS

En esta sección, se representará el grado de parentesco entre dos conjuntos borrosos, por medio de una indistinguibilidad [6, 1]. Es decir, una función $I : [0, 1]^X \times [0, 1]^X \rightarrow [0, 1]$ verificando:

- Reflexividad $I(\mu, \mu) = 1$
- Simetría $I(\mu, \sigma) = I(\sigma, \mu)$
- T-transitividad, con una t-norma continua T : $T(I(\mu, \sigma), I(\sigma, \lambda)) \leq I(\mu, \lambda)$, para cualesquiera μ, σ, λ en $F^*(X)$.

Hasta ahora, sólo se ha considerado que dos conjuntos borrosos están relacionados de forma nítida si pertenecen a la relación **fr**. Dentro del conjunto de predicados que mantienen la relación **fr**, se definirá una indistinguibilidad para calcular el grado parentesco.

Para construir prod-indistinguibilidades se utilizará el teorema de Ovchinnikov (véase [3]): Cualquier prod-indistinguibilidad en $F^*(X)$ se puede representar por,

$$E(\mu, \sigma) = Inf_{f \in F} \left(\frac{f(\mu)}{f(\sigma)}, \frac{f(\sigma)}{f(\mu)} \right),$$

con F una familia de funciones $f : F^*(X) \rightarrow R^+ - \{0\}$.

Como se hizo en [5], se construirá una prod-indistinguibilidad para los conjuntos borrosos en $F(X)$ que sean integrables.

Sean $f_1(\mu) = \int_X \mu(x)dx$, $f_2(\mu) = \int_X (X - Z(\mu))$. Entonces se define,

$$Ovch(\mu, \sigma) = \min \left(\frac{f_1(\mu)}{f_1(\sigma)}, \frac{f_1(\sigma)}{f_1(\mu)}, \frac{f_2(\mu)}{f_2(\sigma)}, \frac{f_2(\sigma)}{f_2(\mu)} \right),$$

para $\mu, \sigma \in F^*(X)$ integrables.

Así mismo, se construirá otra prod-indistinguibilidad, para comparar los elementos con un conjunto borroso $\mu \in F(X)^*$ integrable, **patrón** de la familia $[\mu]$. Se define la función, $f_\mu : F^*(X) \rightarrow R^+ - \{0\}$, $f_\mu(\alpha) = \frac{\int_X \min(\mu, \alpha)}{\int_X \max(\mu, \alpha)}$ y la prod-indistinguibilidad I_μ , como $I_\mu(\alpha, \beta) = \min \left(\frac{f_\mu(\alpha)}{f_\mu(\beta)}, \frac{f_\mu(\beta)}{f_\mu(\alpha)} \right)$, para $\alpha, \beta \in F^*$ integrables.

En particular, $I_\mu(\mu, \sigma) = \min \left(\left(\frac{\int_X \min(\mu, \sigma)}{\int_X \max(\mu, \sigma)} \right)^{-1}, \frac{\int_X \min(\mu, \sigma)}{\int_X \max(\mu, \sigma)} \right) = \frac{\int_X \min(\mu, \sigma)}{\int_X \max(\mu, \sigma)}$.

Se utilizará esta prod-indistinguibilidad para calcular el grado de parentesco de sus parientes con μ y también con conjuntos borrosos que no sean parientes. Es decir, se calculará $I_\mu(\mu, \sigma)$, para todo $\sigma \in F^*(X)$ integrable.

Combinando ambas prod-indistinguibilidades se define el *Grado de Parentesco* entre los conjuntos borrosos de la familia $[\mu]$, como

$$GP_\mu(\alpha, \beta) = \min(Ovch(\alpha, \beta), I_\mu(\alpha, \beta)),$$

para $\alpha, \beta \in F(X)^*$ integrables. Gracias al teorema de Ovchinnikov, se obtiene de nuevo que GP_μ una prod-

indistinguibilidad.

Si se calcula ese grado de parentesco del patrón con sus parientes,

$$GP_\mu(\mu, \sigma) = \min \left(Ovch(\mu, \sigma), \frac{\int \min(\mu, \sigma)}{\int \max(\mu, \sigma)} \right),$$

para $\sigma \in [\mu]$ integrable.

Ejemplo 3.1 Tomado el conjunto borroso μ como patrón (representado en la figura 5), se calcula el grado de parentesco con sus parientes, μ_1 y μ_2 (representados respectivamente en las figuras 6 y 7),

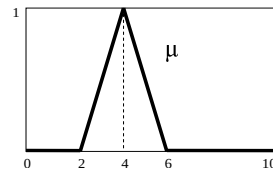


Figura 5:

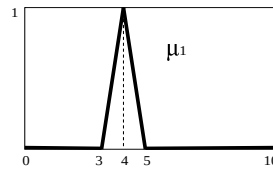


Figura 6:

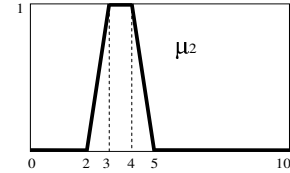


Figura 7:

$$GP_\mu(\mu, \mu_1) = \min(Ovch(\mu, \mu_1), I_\mu(\mu, \mu_1)) = \min\left(\min\left(\frac{2}{4}, \frac{4}{2}, \frac{1}{2}, \frac{2}{1}\right), \frac{2}{4}\right) = \frac{1}{2} = 0,5.$$

$$GP_\mu(\mu, \mu_2) = \min(Ovch(\mu, \mu_2), I_\mu(\mu, \mu_2)) = \min\left(\min\left(\frac{3}{4}, \frac{4}{3}, \frac{2}{2}, \frac{1,5}{2,5}\right), \frac{1,5}{2,5}\right) = \frac{1,5}{2,5} = 0,6.$$

La familia de parientes de μ , $[\mu] = \{\sigma \in F(X)^*; (\mu, \sigma) \in \mathbf{fr}\} = \{\mu, \mu_1, \mu_2\}$, y los grados de parentesco de cada elemento de la familia con el patrón, se pueden representar con el conjunto borroso discreto, $[\mu]^*$. En el ejemplo 3.1, se obtendría $[\mu]^* = 0,5/\mu_1 + 0,6/\mu_2 + 1/\mu$, es decir un conjunto borroso discreto formado a su vez por conjuntos borrosos que pertenecen con cierto grado, $[\mu]^* \in [0, 1]^{[0, 1]^X}$.

El patrón, μ , de la familia, será el prototipo de $[\mu]^*$, por pertenecer con grado 1.

Se puede calcular también el grado de parentesco de μ con otros conjuntos difusos en $F(X)^*$ integrables, aunque no sean parientes de μ .

Ejemplo 3.2 En este ejemplo se calculará el grado de parentesco de dos conjuntos borrosos que no están relacionados por $\mathbf{fr}$,

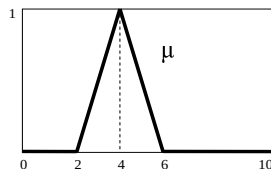


Figura 8:

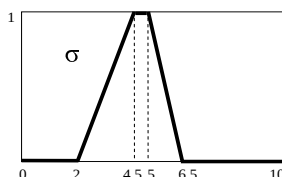


Figura 9:

$$GP_{\mu}(\mu, \sigma) = \frac{\min(Ovch(\mu, \sigma), I_{\mu}(\mu, \sigma))}{\min(\min(\frac{2}{2,25}, \frac{2,25}{2}, \frac{4}{4,5}, \frac{4,5}{4}), \frac{1,7770}{2,7230})} = \frac{1,7770}{2,7230} = 0,6526.$$

Consideramos que el grado de parentesco es alto (por ser mayor de 0.5), aunque no sean parientes.

3.1. PARIENTES DE SEGUNDO ORDEN

En el ejemplo 3.2, se ve que dos conjuntos borrosos que no son de la misma familia, tienen sin embargo un alto grado de parentesco. Esto nos conduce a pensar que se podría crear una familia de parientes de segundo orden más grande de conjuntos borrosos relacionados.

Para tal fin, se calculará el grado de solapamiento ([2]) entre ambos conjuntos de parientes.

Se define la grado de solapamiento de dos conjuntos borrosos discretos $A = a_1/x_1 + \dots + a_n/x_n$, y $B = b_1/y_1 + \dots + b_m/y_m$, por la fórmula:

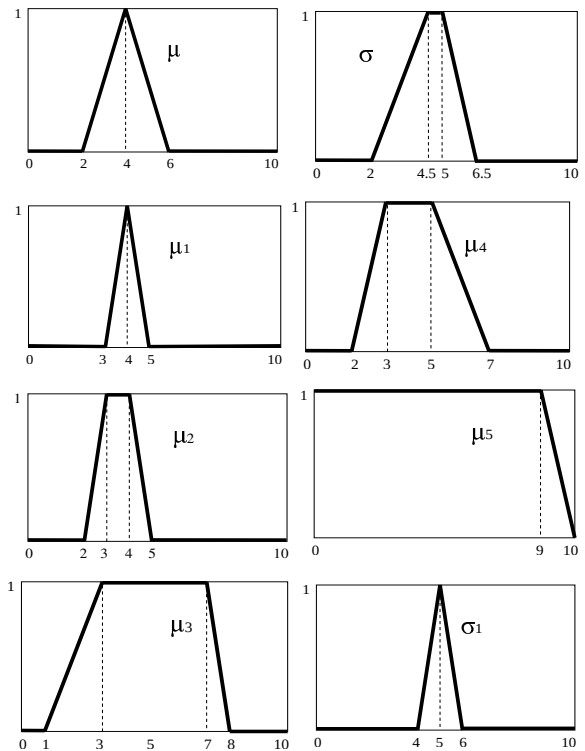
$$Solap(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)},$$

tomando la cardinalidad borrosa $|A| = \sum_{i \in \{1, \dots, n\}} a_i$. El solapamiento entre dos conjuntos verifica las propiedades reflexiva y simétrica, pero no la transitiva.

Si el solapamiento es alto (lo consideraremos alto si es superior a 0.5) se podrían considerar los parientes de segundo orden, y se llamará conjunto de parientes de segundo orden a la unión de los dos conjuntos de parientes $[\mu, \sigma] = [\mu] \cup [\sigma]$.

Sea $F^*(X)$ el conjunto de todos los conjuntos borrosos de la figura 10. Tomando como patrón μ y sabiendo que σ tiene un alto grado de parentesco con μ (calculado en el ejemplo 3.2), se obtiene la familia de parientes de μ , $[\mu] = \{\mu, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5\}$, y la de σ , $[\sigma] = \{\sigma, \sigma_1, \mu_3, \mu_4, \mu_5\}$

Se calculan todos los grados de parecido $GP_{\mu}(\mu, \alpha)$ y $GP_{\sigma}(\mu, \beta)$, con $\alpha \in [\mu]$ y $\beta \in [\sigma]$:

Figura 10: $F^*(X)$

- $GP_{\mu}(\mu, \mu) = 1$
- $GP_{\mu}(\mu, \mu_1) = 0,5$
- $GP_{\mu}(\mu, \mu_2) = 0,6$
- $GP_{\mu}(\mu, \mu_3) = 0,3636$
- $GP_{\mu}(\mu, \mu_4) = 0,5714$
- $GP_{\mu}(\mu, \mu_5) = 0,2094$
- $GP_{\sigma}(\sigma, \sigma) = 1$
- $GP_{\sigma}(\sigma, \mu_3) = 0,2727$
- $GP_{\sigma}(\sigma, \mu_4) = 0,4286$
- $GP_{\sigma}(\sigma, \mu_5) = 0,1571$
- $GP_{\sigma}(\sigma, \sigma_1) = 0,4$

Con estos cálculos se puede representar las familias como,

- $[\mu]^* = 0,5/\mu_1 + 0,6/\mu_2 + 0,3636/\mu_3 + 0,5714/\mu_4 + 0,2094/\mu_5 + 1/\mu,$
- $[\sigma]^* = 0,2727/\mu_3 + 0,4286/\mu_4 + 0,1571/\mu_5 + 0,4/\sigma_1 + 1/\sigma,$

y se tienen las cardinalidades,

- $|[\mu]^*| = 2,2444, |[\sigma]^*| = 1,5584,$
- $|([\mu] \cap [\sigma])^*| = \frac{\min(0,3636, 0,2727)}{\mu_3} + \frac{\min(0,5714, 0,4286)}{\mu_4} + \frac{\min(0,2094, 0,1571)}{\mu_5} = 0,8584.$

Por lo tanto, $Solap([\mu]^*, [\sigma]^*) = \frac{|A \cap B|}{\min(|A|, |B|)} = \frac{0,8584}{1,2584} = 0,6821$.

Se puede considerar este solapamiento alto; por tanto cabe obtener el siguiente conjunto de parientes de segundo orden,

$$[\mu, \sigma] = \{\mu, \sigma, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \sigma_1\}.$$

3.2. GRADO DE PARENTESCO DE SEGUNDO ORDEN

Partiendo ahora de los grados de parentesco con μ y con σ de dos conjuntos borrosos, se definirá el *Grado de Parentesco de segundo orden*,

$$GP_{\mu, \sigma}^2(\alpha, \beta) = \max(GP_{\mu}(\alpha, \beta), GP_{\sigma}(\alpha, \beta)),$$

para todo $\alpha, \beta \in [\mu, \sigma]$.

Ejemplo 3.3 Calculemos el grado de parentesco de segundo grado dentro del conjunto $[\mu, \sigma]$, de los conjuntos μ_1, σ_1 de la Figura 10 pertenecientes a familias distintas de parientes, $\mu_1 \in [\mu]$ y $\sigma_1 \in [\sigma]$.

$$GP_{\mu, \sigma}^2(\mu_1, \sigma_1) = \max(GP_{\mu}(\mu_1, \sigma_1), GP_{\sigma}(\mu_1, \sigma_1)) = \max(0,5696, 0,9314) = 0,9314.$$

El grado obtenido es muy alto, ya que σ_1 no es más que una traslación de μ_1 .

Veamos como se comporta este grado,

- Si $Ovch(\alpha, \beta) \leq I_{\mu}(\alpha, \beta)$ ó $I_{\sigma}(\alpha, \beta)$, se obtiene $GP_{\mu, \sigma}^2(\alpha, \beta) = Ovch(\alpha, \beta)$
- En otro caso,
 - si $I_{\mu}(\alpha, \beta) \leq I_{\sigma}(\alpha, \beta)$, se obtiene $GP_{\mu, \sigma}^2(\alpha, \beta) = I_{\sigma}(\alpha, \beta)$.
 - si $I_{\sigma}(\alpha, \beta) \leq I_{\mu}(\alpha, \beta)$, se obtiene $GP_{\mu, \sigma}^2(\alpha, \beta) = I_{\mu}(\alpha, \beta)$.

Es decir, si los conjuntos no se parecen mucho entre sí, su grado de parentesco de segundo orden no será muy alto.

4. CONCLUSIONES

En este artículo se presenta un modelo que relaciona distintas representaciones de los predicados en el lenguaje. En primer lugar, dado un uso de un predicado por un conjunto borroso, que llamamos patrón, se calculan los parientes de este conjunto borroso, que

pueden ser sinónimos del patrón, migraciones a otro universo de discurso, o simplemente otros conjuntos borrosos que se comporten como el patrón (esto es que verifiquen con él la relación **fr**).

Posteriormente, se calcula un grado de parentesco entre el patrón y sus parientes. El grado de parentesco que se propone en este artículo es más conservador que el que se introduce en [5], pues es menor, y además se define mediante una *Prod*-indistinguibilidad, por lo que permite una mayor interacción de los conjuntos borrosos involucrados que en el grado propuesto en [5] conseguido a través de una *W*-indistinguibilidad. Si se encuentra un conjunto borroso con grado de parentesco alto y que no pertenece a la familia del patrón, se comprueba si entre las familias del patrón y de este conjunto borroso, hay un solapamiento alto. Si así fuese, se obtienen los parientes de segundo orden, que son la unión de las familias del patrón y de dicho conjunto borroso. Finalmente, se define grado de parentesco de segundo orden en esta nueva familia extendida.

Referencias

- [1] L. Valverde E. Trillas. An inquiry into indistinguishability operators. *Aspects of Vagueness. H.J. Skala, S. Termini, E. Trillas (Eds.). Reidel*, págs. 231–256, 1984.
- [2] S. Guadarrama and M. Garrido. Concept-Analyzer: A tool for analyzing fuzzy concepts. En *Proceedings of IPMU*, págs. 1084–1089, 2008.
- [3] S.V. Ovchinnikov. Representation of transitive fuzzy relations. *Aspects of Vagueness. H.J. Skala, S. Termini, E. Trillas (Eds.). Reidel*, págs. 105–118, 1984.
- [4] E. Trillas. On a model for the meaning of predicates. A naïve approach to the genesis of fuzzy sets. En Rudolf Seising, editor, *Fuzzy Sets Theory. Philosophy and Criticism*, volumen 243, *Studies in Fuzziness and Soft Computing*, capítulo 9, págs. 175–205. Springer Verlag, 2009.
- [5] E. Trillas, C. Moraga, and A. Sobrino. On ‘family resemblances’ with fuzzy sets. *Proceedings IFSA-EUSFLAT*, págs. 306–311, 2009.
- [6] E. Trillas and L. Valverde. On implication and indistinguishability in the setting of fuzzy logic. En D. Dubois, H. Prade, and R. R. Yager, editores, *Readings in Fuzzy Sets for Intelligent Systems*, págs. 97–104. Kaufmann, San Mateo, CA, 1993.
- [7] L. Wittgenstein. *Philosophical Investigations*. Basil Blackwell, Oxford, Tercera Edición, 1981.

FROM COMPUTER ALGEBRA TO DISCRETIZED CONTINUOUS LOGIC

Eugenio Roanes-Lozano¹ Javier Montero² Antonio Hernando³ Luis M. Laita⁴

¹ Depto. de Álgebra, Universidad Complutense de Madrid, eroanes@mat.ucm.es

² Depto. de Estadística e Investigación Operativa, Universidad Complutense de Madrid, monty@mat.ucm.es

³ Depto. de Sistemas Inteligentes Aplicados, Universidad Politécnica de Madrid, ahernando@eui.upm.es

⁴ Depto. de Inteligencia Artificial, Universidad Politécnica de Madrid, laita@fi.upm.es

Abstract

The aim of this paper is to present a new algebraic approach from computer algebra to a discretized continuous logic. It makes use of a previous model of p -valued logic (where p is a prime number) based on the use of Gröbner bases of polynomial ideals. A five-valued logic (i.e., $p = 5$) with some modal operators has been considered as a compromise between precision and complexity of the polynomials involved. Therefore the continuous truth values are discretized into five intervals corresponding to the likelihood levels: impossible or very unlikely / unlikely / dubious / probably / almost sure or absolutely sure. It is therefore possible to obtain the likelihood level of any given logic formula. Moreover, it is possible to perform knowledge extraction and verification of small Rule Based Expert System whose knowledge is represented by this logic. An implementation in the computer algebra system *Maple* is included.

Palabras Clave: Continuous Logic, Computer Algebra, Gröbner Bases, Rule Based Expert Systems.

1 INTRODUCTION

In previous research, the authors have developed a polynomial model for knowledge extraction and consistency checking in Rule Based Expert Systems (RBES). The underlying logic of such a polynomial model is the classical Boolean logic or finitely-valued logics with modal operators [12, 13]. In particular, the proposed model is based on the theory of Gröbner bases (GB) [2, 3]. Following this approach, the authors have so

far designed and developed RBES in different fields [9, 11, 15]. These works are summarized in Section 2.

In [14] the authors developed an algebraic approach to minimal polynomial continuous logic (MPL) that can be applied to perform knowledge extraction and verification (consistency checking) of general RBES whose underlying logic is MPL, but, unfortunately, it only provides partial results and sufficient conditions, unlike the Boolean and finitely-valued cases.

An implementation of any continuous logic, fuzzy logic in particular, presents serious computational and estimation difficulties. The aim of this paper is to perform a logical effective calculations and to perform knowledge extraction and consistency checking in RBES whose underlying logic is continuous, by means of a discretized continuous logic.

In this way, our research extends previous works by Kapur and Narendran [8] and Hsiang [7] (classical Boolean logic case) and Alonso et al. [1, 4] (finitely-valued logics case), where techniques for performing effective calculations in logic using Gröbner bases are treated.

2 PREVIOUS RESULTS

2.1 POLYNOMIAL IDEALS AND GRÖBNER BASES

A polynomial ideal is a subset of a polynomial ring which fulfils some specific requirements: it is also a ring and the product of any element of the ideal by any element of the ring lies within the ideal. The ideal generated by the polynomials, $p_1, \dots, p_m$ (this ideal is mathematically denoted by $\langle p_1, \dots, p_m \rangle$) is the minimum ideal containing $p_1, \dots, p_m$ [5].

The key result in ideal theory is due to Buchberger [2, 3, 5]. It happens that different bases may generate the same ideal and Buchberger designed a theory and an algorithm for finding a specific basis, which he

called “Gröbner basis” (as a tribute to his PhD advisor), that characterized each polynomial ideal [2, 3, 5]. A constructive method for calculating the “normal form” (NF) of a polynomial modulo an ideal (the residue of the polynomial modulo the ideal) was also included in the theory. The most relevant application of Buchberger approach is the solution of the “ideal membership problem”: if g is a polynomial and L is an ideal, then

$$g \in L \text{ if and only if } NF(g, L) = 0$$

2.2 A POLYNOMIAL MODEL FOR FINITELY-VALUED MODAL LOGIC

Let us suppose that the logic considered is a p -valued one (with some modal operators) and that the propositional variables are $P_1, \dots, P_w$. Then the polynomial residue class ring

$$\mathbb{Z}_p[p_1, \dots, p_w]/I$$

where I is the ideal

$$I = \langle p_i^p, \dots, p_w^p \rangle$$

with certain operations translating the connectives of the logic is isomorphic to the p -valued logic. Given a propositional formula, α , we use $\varphi(\alpha)$ to denote the polynomial representing α (in the Boolean case this representation is closely related to Boole’s mathematical approach to logic). Details about this translation in the p -valued case can be found in [13]. According to Theorem 1 (below), the problem of checking if a propositional formula α can be inferred (formally termed as “tautological consequence”) from others, $\beta_1, \dots, \beta_m$, may be dealt by checking a polynomial ideal membership:

Theorem 1: A propositional formula α is a tautological consequence of a set of formulae $\{\beta_1, \dots, \beta_m\}$, if and only if

$$\varphi(\neg\alpha) \in \langle \varphi(\neg\beta_1), \dots, \varphi(\neg\beta_m) \rangle$$

A long detailed proof of this beautiful theorem (in the p -valued case) can be found in [13].

2.3 RULE-BASED EXPERT SYSTEMS

A Rule-Based Expert System (RBES) consists of three basic components:

- The *Knowledge Base*. It is concerned with the information contained in the Expert System trying to model the knowledge of human experts. In a

RBES, this knowledge is characterized by means of a set of production rules which are used along with the input of the RBES to derive the output of the system. The process of developing this knowledge base requires the choice of a representation paradigm for modeling all the information described in natural language by human experts. In the case of the method described in the present paper, the suitable information is represented by means of polynomials. Consequently, in our system, the information related to the input (facts), output and the knowledge base (production rules) must be translated to polynomials. In order to make easy this translation, we are previously required to represent first all this information in terms of propositional logic.

- The *inference engine*. It is related to the technique used to make deductions automatically (that is to say, the mechanism by means of which the Expert System derives the output from the given input). By means of a previous mathematical result (see Theorem 1), problems associated to “deduction” may be translated into algebraic problems, so that we can use a mathematical algebraic software as inference engine.
- An interactive *Graphic User’s Interface*. Through it, users may easily introduce the information concerning the input of the Expert System and observe the resulting output the Expert System automatically infers from the former.

2.4 A POLYNOMIAL MODEL FOR RULE-BASED EXPERT SYSTEMS

In the case of a RBES, Theorem 1 can be formulated in the following way:

Theorem 2: A certain formula q can be inferred from the knowledge in the RBES (described by the production rules $R_1, \dots, R_r$) and the facts $\{F_1, \dots, F_f\}$ if and only if:

$$\varphi(\neg q) \in \langle \varphi(\neg R_1), \dots, \varphi(\neg R_r), \varphi(\neg F_1), \dots, \varphi(\neg F_f) \rangle$$

In order to simplify this expression, we can define the ideal J , generated by the production rules in the knowledge base,

$$J = \langle \varphi(\neg R_1), \dots, \varphi(\neg R_r) \rangle$$

and the ideal K generated by the given facts,

$$K = \langle \varphi(\neg F_1), \dots, \varphi(\neg F_f) \rangle$$

By means of these two ideals, the expression above would be transformed into:

$$\varphi(\neg q) \in J + K$$

and the latter membership can, again, be decided using normal forms, as it is equivalent to:

$$NF(\neg(q), J + K) = 0$$

if working in $\mathbb{Z}_p[p_1, \dots, p_w]/I$, or to:

$$NF(\neg(q), I + J + K) = 0$$

if working in $\mathbb{Z}_p[p_1, \dots, p_w]$. Knowledge extraction in this kind of RBES can be therefore effectively calculated. Moreover, $(\mathbb{Z}_p[p_1, \dots, p_w]/I)/(J + K)$ is in fact isomorphic to the logic structure associated to the RBES (see [12] for details). Consequently, consistency checking for a given set of facts $\{\varphi(\neg F_1), \dots, \varphi(\neg F_f)\}$ is equivalent to the non-degeneracy of ideal $J + K$ of $(\mathbb{Z}_p[p_1, \dots, p_w]/I)$ into the whole ring, what is equivalent to:

$$GB(J + K) \neq \{1\}$$

if working in $\mathbb{Z}_p[p_1, \dots, p_w]/I$, or to:

$$GB(I + J + K) \neq \{1\}$$

if working in $\mathbb{Z}_p[p_1, \dots, p_w]$.

2.5 MAPLE IMPLEMENTATION

The implementation in a computer algebra system *Maple*, *CoCoA*, etc. is surprisingly brief. For instance, all the code corresponding to Kleene's five-valued logic (with some modal operators) is included afterwards. Firstly the polynomial variables, ring, variable ordering and ideal *iI* have to be defined (*I* is a reserved word in *Maple*, so we denote ideal *I* by *iI*):

```
> with(Groebner):
> with(Ore_algebra):
> SV:=x[1],x[2],x[3],x[4]:
> fu:=v->v^5-v:
> A:=poly_algebra(SV,characteristic=5):
> Orde:=MonomialOrder(A,'plex'(SV)):
> iI:=map(fu,[SV]):
```

and then the unary and binary connectives can be defined:

```
> NEG :=(m::algebraic) ->
>   NormalForm(4*m+4,iI,Orde):
> POS1 :=(m::algebraic) ->
>   NormalForm(expand(4*m^4),iI,Orde):
> POS2 :=(m::algebraic) ->
>   NormalForm(expand(3*m^4+4*m^3+4*m^2
>   +4*m),iI,Orde):
> NEC2 :=(m::algebraic) ->
>   NormalForm(expand(2*m^4+2*m^3+m),
>   iI,Orde):
> NEC1 :=(m::algebraic) ->
```

```
>   NormalForm(expand(m^4+4*m^3+m^2+
>   4*m),iI,Orde):
> '&OR' :=(m::algebraic,n::algebraic) ->
>   NormalForm(expand(2*m^4*n^2+
>   4*m^3*n^3+2*m^2*n^4+2*m^4*n+2*m*n^4+
>   m^3*n+3*m^2*n^2+m*n^3+2*m*n+m+n),
>   iI,Orde):
> '&AND' :=(m::algebraic,n::algebraic) ->
>   NormalForm(expand(3*m^4*n^2+m^3*n^3+
>   3*m^2*n^4+3*m^4*n+3*m*n^4+4*m^3*n+
>   2*m^2*n^2+4*m*n^3+3*m*n),iI,Orde):
> '&IMP' :=(m::algebraic,n::algebraic) ->
>   NEG(m) &OR n:
```

3 DISCRETIZED CONTINUOUS LOGIC

In this section we present an algebraic approach to a discretized continuous logic.

Our approach is straightforward once we have a model for finitely-valued logics, as shown below. Notice that at a first stage we shall in fact define a crisp partition of the unit interval, leaving for a future research the case with fuzzy classes, either in the sense of Ruspini [16] or Montero *et al.* [6, 10]. At this stage, five likelihood levels will be considered in the discretized continuous logic:

- impossible or very unlikely,
- unlikely,
- dubious,
- probably,
- almost sure or absolutely sure.

The continuous truth values are discretized as follows:

- $[0, 1/5) \rightsquigarrow$ impossible or very unlikely $\rightsquigarrow$
 $\rightsquigarrow 0$ in the five-valued logic,
- $[1/5, 2/5) \rightsquigarrow$ unlikely $\rightsquigarrow$
 $\rightsquigarrow 1$ in the five-valued logic,
- $[2/5, 3/5) \rightsquigarrow$ dubious $\rightsquigarrow$
 $\rightsquigarrow 2$ in the five-valued logic,
- $[3/5, 4/5) \rightsquigarrow$ probably $\rightsquigarrow$
 $\rightsquigarrow 3$ in the five-valued logic,
- $[4/5, 1] \rightsquigarrow$ almost sure or absolutely sure $\rightsquigarrow$
 $\rightsquigarrow 4$ in the five-valued logic.

Given a certain logic formula and the numerical truth values of the propositional variables in the logic formula (ranging in $[0, 1]$), the process to obtain the likelihood level of the formula in the discretized continuous logic is the following:

- 1) compute the polynomial corresponding to the logic formula,
- 2) calculate the the corresponding truth values of the propositional variables in the five-valued logic (ranging in $\{0, 1, 2, 3, 4\}$), as detailed above,
- 3) substitute the latter truth values in the polynomial,
- 4) find to which likelihood level the truth value obtained for the formula corresponds.

Let us note that, once the values are discretized, the value obtained for disjunction and conjunction correspond to maximum and minimum (if one of the usual finitely-valued logics such as Łukasiewicz's or Kleene's are chosen). For instance, the truth table of the connective "conjunction" is, in the five-valued case:

$\wedge$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	2	2	2
3	0	1	2	3	3
4	0	1	2	3	4

3.1 IMPLEMENTATION

The code included in Section 2.5 can be reused for this discretization of a continuous logic. Two more procedures have to be added:

- `tVal` substitutes the numerical values in the polynomial translating the logic formula (steps 2) and 3) of Section 3),
- `evalFu` finds to which likelihood level the truth value obtained for the formula corresponds (step 4) of Section 3).

Truth value introduction is done in *Maple* as follows: if variable `x[2]` has a truth value 0.65 we assign:

```
> val(x[2]) := .65;
```

Let us include a simple example afterwards:

```
> val(x[1]) := .3;
> val(x[2]) := .1;
> val(x[3]) := .75;
```

```
> form1 := x[1] &OR (x[2] &OR x[3]);
> evalFu(tVal(form1));
      Probably
```

4 AN ALGEBRAIC APPROACH TO RBES

The algebraic model is the same as in the previous section, but, instead of computing likelihood levels from truth values, we perform knowledge extraction (and verification). In particular, in the above five-valued logic we have that

- $\Box_1 f$ is true iff f has the truth value 4 (true),
- $\Box_2 f$ is true iff f has a truth value ≥ 3 ,
- $\Diamond_2 f$ is true iff f has a truth value ≥ 2 ,
- $\Diamond_1 f$ is true iff f has a truth value ≥ 1 ,
- $\Diamond_1 f \vee \Diamond_1 \neg f$ is always true.

as can be checked in the following truth table

f	$\Diamond_1 f \vee \Diamond_1 \neg f$	$\Diamond_1 f$	$\Diamond_2 f$	$\Box_2 f$	$\Box_1 f$
0	4	0	0	0	0
1	4	4	0	0	0
2	4	4	4	0	0
3	4	4	4	4	0
4	4	4	4	4	4

Consequently,

- $\Box_1 f$ is true iff f has the truth value 4,
- $\Box_2 f \wedge \Diamond_1 \neg f$ is true iff f has the truth value 3,
- $\Diamond_2 f \wedge \Diamond_2 \neg f$ is true iff f has the truth value 2,
- $\Box_2 \neg f \wedge \Diamond_1 f$ is true iff f has a truth value 1,
- $\Box_1 \neg f$ is true iff f has the truth value 0.

as can be checked in the following truth table

f	$\Box_1 f$	$\Box_2 f \wedge \Diamond_1 \neg f$	$\Diamond_2 f \wedge \Diamond_2 \neg f$	$\Box_2 \neg f \wedge \Diamond_1 f$	$\Box_1 \neg f$
0	4	0	0	0	0
1	0	4	0	0	0
2	0	0	4	0	0
3	0	0	0	4	0
4	0	0	0	0	4

Therefore, when translating the rules from the experts' vocabulary into logic, it has to be considered for which truth values we want the rule to be fired.

Example 1: For instance, the rule $R1$:

$$\Box_2(q) \rightarrow g$$

is fired if

the likelihood level of q is “probably”
or “almost sure or absolutely sure”

i.e., if

q has the truth values 3 or 4 in the five-valued logic.

Once discretized, knowledge extraction and consistency checking can be performed using normal forms and Gröbner bases, exactly as in Section 2.4.

4.1 EXAMPLES IN MAPLE

Example 2: Let us check Example 1 with *Maple*: 0 is obtained as the normal form of g with respect to the ideal generated by a base of ideal I , the negation of the antecedent and a base of ideal J , when the antecedent is:

$$\Box_1 f \quad \text{or} \quad \Box_2 f \wedge \Diamond_1 \neg f$$

and not when the antecedent is:

$$\Diamond_2 f \wedge \Diamond_2 \neg f \quad \text{or} \quad \Box_2 \neg f \wedge \Diamond_1 f \quad \text{or} \quad \Box_1 \neg f$$

as shown afterwards:

```
> R1:=NEC2(q) &IMP g:
> J:=[NEG(R1)]:
> B:=Basis([op(iI),NEG(NEC1(q)),op(J)],Orde):
> NormalForm(NEG(g),B,Orde):
0
> B:=Basis([op(iI),NEG(NEC2(q) &AND
> POS1(NEG(q))),op(J)],Orde):
> NormalForm(NEG(g),B,Orde):
0
> B:=Basis([op(iI),NEG(POS2(q) &AND
> POS2(NEG(q))),op(J)],Orde):
> NormalForm(NEG(g),B,Orde):
4 g + 4
> B:=Basis([op(iI),NEG(NEC2(NEG(q)) &AND
> POS1(q)),op(J)],Orde):
> NormalForm(NEG(g),B,Orde):
4 g + 4
> B:=Basis([op(iI),NEG(NEC1(NEG(q))),op(J)],
> Orde):
> NormalForm(NEG(g),B,Orde):
4 g + 4
```

Example 3: Let us consider the following tiny 10-rules RBES (we already include the *Maple* code, in order to show its simplicity):

```
> R1:=NEC1(q) &IMP p2:
> R2:=q &IMP p3:
> R3:=r &IMP p2:
> R4:=r &IMP p3:
> R5:=(q &AND x) &IMP g:
> R6:=NEC1(s) &IMP x:
> R7:=s &IMP p3:
> R8:=u &IMP g:
> R9:=v &IMP p1:
> R10:=w &IMP p2;
```

Therefore, the ideal of rules (J) is:

```
> J:=[ NEG(R1),NEG(R2),NEG(R3),NEG(R4),
> NEG(R5),NEG(R6),NEG(R7),NEG(R8),
> NEG(R9),NEG(R10)]:
```

Let us state as true (i.e., consider the facts):

$$\Box_1(s) \quad \text{and} \quad q.$$

Then the ideal of facts (K) is:

```
> K:=[NEG(NEC1(s)),NEG(q)]:
```

Now, do x and g follow from these facts? Let us first check the consistency for this set of facts, i.e., that ideal $I + J + K$ is not the whole ring (i.e., that the corresponding Gröbner basis is not [1]):

```
> B:=Basis([op(iI),op(J),op(K)], Orde);
B := [1 + g, 1 + p3, p2 + 1, ...]
```

Now we can check whether x and g follow from this facts:

```
> NormalForm(NEG(x),B,Orde);
0
> NormalForm(NEG(g),B,Orde);
0
```

so both of them do follow.

Example 4: Let us consider the same tiny RBES and let us now state as true (i.e., consider the facts):

$$\Box_2(s) \wedge \Diamond_1(\neg(s)) \quad \text{and} \quad q.$$

Then the ideal of facts (K) is:

```
> K:=[NEG(NEC2(s) &AND POS1(NEG(s))),NEG(q)];
```

Now, do x and g follow from these facts? The following Gröbner basis is not [1], so there is consistency for this set of facts:

```
> B:=Basis([op(iI),op(J),op(K)], Orde);
5
B := [g + 4 g, 1 + p3, p2 + 1, ...]
```

And:

```
> NormalForm(NEG(x),B,Orde);
                                     4 x + 4
> NormalForm(NEG(g),B,Orde);
                                     4 g + 4
```

are not 0, so neither x nor g follow from this set of facts.

5 CONCLUSIONS

The approach presented in this paper provides a first computational frame for performing effective computations (knowledge extraction and verification) in RBES whose underlying logic is a continuous logic, by means of its discretization. The drawback is the complexity of the polynomials involved, that only allow to treat medium-size RBES (in the Boolean case RBES with figures like a hundred rules can be treated in a standard computer in a matter of seconds). Future research will consider a more sophisticated discretization approach in order to implement alternative modal or fuzzy logics.

Acknowledgements

This work was partially supported by grants TIN2009-07901 (Government of Spain) and UCM2008-910563 (UCM Research Group 58/08, ACEIA).

References

- [1] J. A. Alonso, E. Briales: Lógicas Polivalentes y Bases de Gröbner. In: C. Martin, ed.: *Actas del V Congreso de Lenguajes Naturales y Lenguajes Formales*. University of Seville, Seville (1995) 307–315.
- [2] B. Buchberger: Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of Symbolic Computation* 41/3-4 (2006) 475–511.
- [3] B. Buchberger: Applications of Gröbner Bases in Non-Linear Computational Geometry. In: J. R. Rice, ed.: *Mathematical Aspects of Scientific Software*. Springer-Verlag, IMA Vol. 14, New York (1988) 60–88.
- [4] J. Chazarain, A. Riscos, J. A. Alonso, E. Briales: Multivalued Logic and Gröbner Bases with Applications to Modal Logic. *Journal of Symbolic Computation* 11 (1991) 181–194.
- [5] D. Cox, J. Little, D. O'Shea: Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, New York, 1992.
- [6] A. Del Amo, J. Montero, G. Biging, V. Cutello: Fuzzy classification systems. *European Journal of Operational Research* 156 (2004) 459–507.
- [7] J. Hsiang: Refutational Theorem Proving using Term-Rewriting Systems. *Artificial Intelligence* 25 (1985) 255–300.
- [8] D. Kapur, P. Narendran: An Equational Approach to Theorem Proving in First-Order Predicate Calculus. In: *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*, vol. 2 (1985) 1146–1153.
- [9] L.M. Laita, E. Roanes-Lozano, V. Maojo, L. de Ledesma, L. Laita: An Expert System for Managing Medical Appropriateness Criteria Based on Computer Algebra Techniques. *Computers and Mathematics with Applications* 51/5 (2000) 473–481.
- [10] J. Montero, D. Gómez, H. Bustince: On the relevance of some families of fuzzy sets. *Fuzzy Sets and Systems* 158 (2007) 2429–2442.
- [11] C. Pérez-Carretero, L.M. Laita, E. Roanes-Lozano, L. Lázaro, J. González-Cajal, L. Laita: A Logic and Computer Algebra-Based Expert System for Diagnosis of Anorexia. *Mathematics and Computers in Simulation* 58 (2002) 183–202.
- [12] E. Roanes-Lozano, L. M. Laita, E. Roanes-Macías: Maple V in A.I.: The Boolean Algebra Associated to a KBS. *CAN Nieuwsbrief* 14 (1995) 65–70.
- [13] E. Roanes-Lozano, L. M. Laita and E. Roanes-Macías, A Polynomial Model for Multivalued Logics with a Touch of Algebraic Geometry and Computer Algebra. *Mathematics and Computers in Simulation* 45/1 (1998) 83–99.
- [14] E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, A Symbolic-Numeric Approach to MPL Continuous Logic and to Rule Based Expert Systems whose Underlying Logic is MPL. *The Open Applied Mathematics Journal* 1 (2008) 126–133.
- [15] C. Rodríguez-Solano, L.M. Laita, E. Roanes-Lozano, L. López-Corral, L. Laita, A Computational System for Diagnosis of Depressive Situations. *Expert System with Applications* 31 (2006) 47–55.
- [16] E.H. Ruspini: A new approach to clustering. *Information and Control* 15 (1969) 22–32.

On fuzzy homomorphisms between hyperrings*

I. P. Cabrera P. Cordero G. Gutiérrez J. Martínez M. Ojeda-Aciego

Dept. Matemática Aplicada. Univ. de Málaga. Spain

ipcabrera@uma.es, pcordero@uma.es, gloriagb@ctima.uma.es, javim@ctima.uma.es, aciego@uma.es

Abstract

We focus on the study of the structure of hyperrings; in this paper, we recall the basics of crisp homomorphisms between hyperstructures, particularly, between hyperrings and, then, the notion of fuzzy homomorphism between hyperrings is established and its main properties are analysed.

Keywords: Hyperstructures, fuzzy homomorphisms, fuzzy ideals, fuzzy congruences.

- on the other hand, some ideas underlying fuzzy set theory form the crux on which the development of the different approaches to multiple-valued and fuzzy logics has been based.

The use of hyperstructures in conjunction with fuzzy logic has been shown to be fruitful in some areas certainly related to artificial intelligence and soft computing, such as fuzzy logic programming with multilattices [16]. The study of hyperstructures and its (fuzzy) congruences is important both from a theoretical standpoint and for its applications in the field of logic-based approaches to uncertainty. Regarding applications, the notion of congruence is intimately related to the foundations of fuzzy reasoning and its relationships with other logics of uncertainty. More focused on the theoretical aspects of Computer Science, some authors [1, 17] have pointed out the relation between congruences, fuzzy automata and determinism.

In previous works [3, 4] we have studied the relationship between the crisp notions of congruence, homomorphism and substructure on some non-deterministic algebras which have been used in the literature, i.e. hypergroups and join spaces. Moreover, we have provided suitable extension of these notions to the fuzzy case.

In this work, we focus on a particular topic related to fuzzy hyperalgebra, which develops fuzzy versions of hyperstructures. Specifically, we study the theory of hyperrings and fuzzy homomorphisms between them.

The structure of the paper is the following: after stating the preliminary definitions, we recall the basics of the theory of crisp homomorphisms between hyperstructures, recalling specially the results and isomorphism theorems which relate homomorphisms, congruences and ideals. Then, the main contribution of the paper is presented: the extension of the previous relations to the fuzzy case.

1 Introduction

The study of hyperstructures started seventy-five years ago with Marty's paper [15] which firstly used a multiple-valued operator. Nowadays, the theory of hyperstructures is being thoroughly studied, focusing in particular classes such as hypergroups, hyper-near rings, semi-hyperrings, join spaces, etc.

Meanwhile, fuzzy sets were introduced by Zadeh [18] and since then there has been a number of authors who started the development of fuzzy algebra.

The two types of extensions presented in the previous paragraphs have started to be studied jointly, giving rise to the so-called fuzzy hyperalgebra. Several areas have benefitted from the developments in the area of hyperstructures and fuzzy set theory, in particular, artificial intelligence and soft computing:

- hyperstructures can be used as a useful tool to modelling computing with uncertainty, in that the result of an operation is not a single value and can be considered a set of possible values;

* Partially supported by the Spanish Science Ministry by project TIN2006-15455-C03-01 and by Junta de Andalucía project P06-FQM-02049.

2 Preliminary definitions

Firstly, let us introduce some preliminary concepts:

Definition 1 A **hypergroupoid** is a pair $(A, +)$ consisting of a non-empty set A together with a hyperoperation $+: A \times A \rightarrow 2^A \setminus \emptyset$.

Definition 2 A hypergroupoid $(A, +)$ is a **canonical hypergroup** if the following properties hold:

- (i) for every $x, y, z \in A$ $x + (y + z) = (x + y) + z$
- (ii) for every $x, y \in A$ $x + y = y + x$
- (iii) there exists $0 \in A$ such that $0 + x = x$ for all $x \in A$
- (iv) for every $x \in A$ there exists a unique element $x' \in A$ such that $0 \in x + x'$; (we shall write $-x$ for x' and we call it the opposite of x .)
- (v) for every $x, y, z \in A$ if $z \in x + y$ then $y \in x - z$.

In the above definition, for $X, Y \subseteq A$, we denote

$$X + Y = \bigcup_{\substack{x \in X \\ y \in Y}} x + y$$

The following equalities follow easily from the axioms: $-(-x) = x$ and $-(x + y) = -x - y$

Note that in the rest of the paper we will frequently write singletons without braces.

Let us now introduce the definition of hyperring we will work with. We will use that given by Krasner, in which the sum is a hyperoperation with the structure of canonical hypergroup and the product is an associative operation, together with distributivity and adequate conditions for the neutral element of the sum.

Definition 3 (Krasner [14]) A **hyperring** is an algebraic structure $(A, +, \cdot, 0)$ which satisfies:

1. $(A, +)$ is a canonical hypergroup with the neutral element 0.
2. Relating to the multiplication, $(A, \cdot)$ is a semigroup having 0 as a bilaterally absorbing element.
3. The multiplication is distributive respect to the hyperoperation $+$

3 On crisp homomorphisms

We begin by discussing the different versions of the concept of homomorphism on hypergroupoids (also called multigroupoids) appearing in the literature. Some authors that deal with these and other hyperstructures use the following definitions of homomorphism [6].

Definition 4 Let $(A, \cdot)$ and $(B, \cdot)$ be hypergroupoids. A map $h: A \rightarrow B$ is said to be:

- **Benado-homomorphism** if $h(ab) \subseteq h(a)h(b)$, for all $a, b \in A$.
- **Algebraic-homomorphism** if $h(ab) = h(a)h(b)$, for all $a, b \in A$.

Regarding the terminology, we depart here a bit from the usual one. The first one was the original definition by Benado [2], which has been used in several recent papers [6, 7, 12]. However, it is noticeable that, finally, the authors concentrate mostly on the equality-based definition.

The terminology used in those papers is to call homomorphism to Benado's ones and call *good* (or *strong*) homomorphism to algebraic ones. We have adopted the term *algebraic* instead of good or strong because this type of homomorphism immediately allows the lifting of classical homomorphisms to the so-called powerset extension. Obviously, the advantage of using algebraic homomorphisms is that one can transfer properties from the powerset to the hypergroupoid very easily, so that the presentation of multivalued concepts is greatly simplified.

The term homomorphism should induce the properties of the initial hyperalgebra on the image set. It can be easily checked that this is the case for algebraic-homomorphisms but, in general, it is not true for Benado-homomorphisms.

The notion of homomorphism can be easily extended to the structure of hyperrings, since the product operation is classical. The formal definition is given below:

Definition 5 Let $(A, +, \cdot, 0)$ and $(B, +, \cdot, 0)$ be hyperrings. A map $h: A \rightarrow B$ is said to be a **hyperring homomorphism** if $h(a + b) = h(a) + h(b)$ and $h(ab) = h(a)h(b)$, for all $a, b \in A$ and $h(0) = 0$.

Ideals are crucial in ring theory, and they have also been studied in the context of hyperrings; its definition in this case is given below:

Definition 6 Let $(A, +, \cdot, 0)$ be a hyperring. A subset $I \subseteq A$ is said to be an **ideal** of A if the following conditions hold:

- $i - j \subseteq I$ for all $i, j \in I$.
- $aI \subseteq I$ for all $a \in A$.
- $Ia \subseteq I$ for all $a \in A$.

Finally, we recall below the notion of congruence relation on a hyperring that we will extend.

Definition 7 Let $(A, +, \cdot, 0)$ be a hyperring. A **congruence** on A is an equivalence relation $\equiv$ which for all $a, b, c, d \in A$ satisfies that if $a \equiv b$ and $c \equiv d$ then

- for all $x \in a + c$ there exists $y \in b + d$ such that $x \equiv y$,
- for all $y \in b + d$ there exists $x \in a + c$ such that $x \equiv y$ and
- $ac \equiv bd$.

In classical settings, it is usual to consider the kernel relation associated to a homomorphism; this idea has been used in the framework of hyperoperations as follows:

Definition 8 Any hyperring homomorphism $h: A \rightarrow B$ defines a congruence relation, namely kernel relation $\equiv_h$, defined as

$$a \equiv_h b \text{ if and only if } h(a) = h(b)$$

The relationship between the concepts of homomorphism, congruence and ideals in the framework of hyperstructures is similar to that in the classical case.

4 Fuzzy homomorphisms on hyperrings

A fuzzy relation is a mapping φ from $A \times B$ into $[0, 1]$, that is to say, any fuzzy subset of $A \times B$. The powerset extension of a fuzzy relation is defined as, $\widehat{\varphi}: 2^A \times 2^B \rightarrow [0, 1]$ with

$$\widehat{\varphi}(X, Y) = \left(\bigwedge_{x \in X} \bigvee_{y \in Y} \varphi(x, y) \right) \wedge \left(\bigwedge_{y \in Y} \bigvee_{x \in X} \varphi(x, y) \right)$$

The composition of fuzzy relations φ and ψ is defined as follows:

$$(\psi \circ \varphi)(a, c) = \bigvee_{b \in B} \varphi(a, b) \wedge \psi(b, c)$$

A fuzzy relation ρ on $A \times A$ is said to be

1. **reflexive** if $\rho(x, x) = 1$, for every $x \in A$
2. **symmetric** if $\rho(x, y) = \rho(y, x)$, for all $x, y \in A$
3. **transitive** if for all $x, a, y \in A$ we have

$$\rho(x, a) \wedge \rho(a, y) \leq \rho(x, y)$$

A reflexive, symmetric and transitive fuzzy relation on A is called a **fuzzy equivalence**. A fuzzy equivalence ρ on A is called a **fuzzy equality** if for any $x, y \in A$, $\rho(x, y) = 1$ implies $x = y$.

Definition 9 ([3]) A fuzzy equivalence relation ρ on a hyperring $(A, +, \cdot, 0)$ is said to be a **fuzzy congruence relation** if, for all $a_1, a_2, b_1, b_2 \in A$:

- $\rho(a_1, b_1) \wedge \rho(a_2, b_2) \leq \widehat{\rho}(a_1 + a_2, b_1 + b_2)$ and

- $\rho(a_1, b_1) \wedge \rho(a_2, b_2) \leq \rho(a_1 a_2, b_1 b_2)$.

The fuzzification of the concept of function that we adopt has been introduced in [13], also studied in [9, 10, 11], and more recently in [5]. We will introduce the extension of the notion of perfect fuzzy function.

Definition 10 ([9]) Let ρ and σ be fuzzy equalities defined on the sets A and B , respectively. A **partial fuzzy function** φ from A to B is a mapping $\varphi: A \times B \rightarrow [0, 1]$ satisfying the following conditions for all $a, a' \in A$ and $b, b' \in B$:

$$\text{ext1 } \varphi(a, b) \wedge \rho(a, a') \leq \varphi(a', b)$$

$$\text{ext2 } \varphi(a, b) \wedge \sigma(b, b') \leq \varphi(a, b')$$

$$\text{part } \varphi(a, b) \wedge \varphi(a, b') \leq \sigma(b, b')$$

If, in addition, the following condition holds:

$$\text{f1 } \text{For all } a \in A \text{ there is } b \in B \text{ such that } \varphi(a, b) = 1$$

then we say that φ is a **perfect fuzzy function**.

It is not difficult to show that the element b in condition (f1) above is unique. As a result, every perfect fuzzy function defines a crisp mapping from A to B called **crisp description** of φ .

Definition 11 Let $(A, +, \cdot, 0)$ and $(B, +, \cdot, 0)$ be hyperrings endowed with fuzzy equalities ρ and σ , respectively, such that $\sigma(a, b) = \sigma(-a, -b)$

A perfect fuzzy function $\varphi \in [0, 1]^{A \times B}$ is said to be a **fuzzy homomorphism** if for all $a_1, a_2 \in A$ and $b_1, b_2 \in B$, the following conditions hold:

$$\text{Sum-compat } \varphi(a_1, b_1) \wedge \varphi(a_2, b_2) \leq \widehat{\varphi}(a_1 + a_2, b_1 + b_2)$$

$$\text{Prod-compat } \varphi(a_1, b_1) \wedge \varphi(a_2, b_2) \leq \varphi(a_1 a_2, b_1 b_2)$$

$$\text{Neutral } \varphi(0, 0) = 1$$

Moreover, φ is said to be **complete** if the two following conditions hold:

1. if $\bigvee_{y \in Y} \varphi(a, y) = 1$, then there exists $y \in Y$ such that $\varphi(a, y) = 1$.
2. if $\bigvee_{x \in X} \varphi(x, b) = 1$, then there exists $x \in X$ such that $\varphi(x, b) = 1$.

Remark: Hereafter, unless stated otherwise, we will always consider that we are working with a complete fuzzy homomorphism φ between hyperrings A and B and fuzzy equalities ρ and σ , respectively.

Proposition 12 Given φ between A and B , the crisp description h of φ is a hyperring homomorphism.

The notion of fuzzy homomorphism between hyperring behaves properly with respect to the composition of fuzzy relations, in that the composition of fuzzy homomorphisms is a fuzzy homomorphism. Furthermore, the composition is associative and there exists an identity for this composition. As a result, the class of hyperrings together with the fuzzy homomorphisms between them forms a category.

Let us concentrate now on the relationship between fuzzy homomorphism and congruences.

Definition 13 The **fuzzy kernel relation** induced by φ in A is defined as $\rho_\varphi(a, a') = \varphi(a, h(a'))$.

We adopt here the term *kernel* as an extension of the crisp case because of the inequality

$$\varphi(a, b) \wedge \varphi(a', b) \leq \rho_\varphi(a, a')$$

Moreover, in [5], the authors prove the following result.

Proposition 14 Let φ a perfect fuzzy function from A to B . For all $a, a' \in A$,

$$\rho_\varphi(a, a') = \bigvee_{b \in B} \varphi(a, b) \wedge \varphi(a', b)$$

In the case of fuzzy homomorphisms between hyperring, we prove that this fuzzy equivalence relation is also a fuzzy congruence on the initial hyperring.

Theorem 15 Consider φ between A and B . The fuzzy kernel relation ρ_φ is a fuzzy congruence relation which includes the fuzzy equality ρ in A .

PROOF: Let us see that ρ_φ is compatible with the sum operation.

$$\begin{aligned} \widehat{\rho}_\varphi(a_1 + a_3, a_2 + a_4) &= \\ &= \bigwedge_{a \in a_1 + a_3} \bigvee_{a' \in a_2 + a_4} \rho_\varphi(a, a') \wedge \bigwedge_{a' \in a_2 + a_4} \bigvee_{a \in a_1 + a_3} \rho_\varphi(a, a') \\ &= \widehat{\varphi}(a_1 + a_3, h(a_2 + a_4)) \quad \text{as } \rho_\varphi(a, a') = \varphi(a, h(a')) \\ &= \widehat{\varphi}(a_1 + a_3, h(a_2) + h(a_4)) \quad \text{by Prop. 12} \\ &\geq \varphi(a_1, h(a_2)) \wedge \varphi(a_3, h(a_4)) \\ &= \rho_\varphi(a_1, a_2) \wedge \rho_\varphi(a_3, a_4) \end{aligned}$$

The compatibility with the multiplication follows from

$$\begin{aligned} \rho_\varphi(a_1 a_3, a_2 a_4) &= \varphi(a_1 a_3, h(a_2 a_4)) \\ &= \varphi(a_1 a_3, h(a_2) h(a_4)) \quad \text{by Prop. 12} \\ &\geq \varphi(a_1, h(a_2)) \wedge \varphi(a_3, h(a_4)) \\ &= \rho_\varphi(a_1, a_2) \wedge \rho_\varphi(a_3, a_4) \end{aligned}$$

Now, let us show that $\rho \leq \rho_\varphi$:

$$\begin{aligned} \rho(a, a') &= \rho(a, a') \wedge \varphi(a', h(a')) \\ &\leq \varphi(a, h(a')) = \rho_\varphi(a, a') \quad \text{by (ext1)} \quad \square \end{aligned}$$

In the rest of this section we will show the canonical decomposition theorem for a complete fuzzy homomorphism and a fuzzy congruence relation. For suitable extensions on the notions of injectivity and surjectivity we will rely on the definitions given in [9].

Definition 16 A perfect fuzzy function $\varphi \in [0, 1]^{A \times B}$ is said to be:

- **surjective** if for all $b \in B$ there exists $a \in A$ such that $\varphi(a, b) = 1$.
- **injective** if $\varphi(a, b) \wedge \varphi(a', b) \leq \rho(a, a')$, for all $a, a' \in A$ and $b \in B$.
- **bijective** if it is injective and surjective.

The image set is $\text{Im } \varphi = \{b \in B \mid \text{there exists } a \in A \text{ with } \varphi(a, b) = 1\}$.

In order to define the different homomorphisms involved in the decomposition theorem, we have to introduce the quotient set associated to a fuzzy equivalence relation.

Definition 17 Let $(A, +, \cdot, 0)$ be a hyperring and ρ be a fuzzy equivalence relation in A . An **equivalence class** of an element $a \in A$ is defined as

$$\rho(a) \in [0, 1]^A \quad \text{with} \quad \rho(a)(a') = \rho(a, a')$$

The **quotient set** is defined as $A/\rho = \{\rho(a) \mid a \in A\}$ and a fuzzy equality $\bar{\rho}$ can be defined in A/ρ as $\bar{\rho}(\rho(a), \rho(a')) = \rho(a, a')$.

The **fuzzy projection** π from A to A/ρ is defined as $\pi(a, \rho(a')) = \rho(a, a')$.

Proposition 18 Let $(A, +, \cdot, 0)$ be a hyperring, ρ a fuzzy equality in A and ρ_A be a fuzzy congruence relation in A that includes ρ . The **fuzzy projection** π from A to A/ρ_A is a surjective fuzzy homomorphism where the hyperoperations in A/ρ_A are given by

$$\begin{aligned} \rho_A(a_1) + \rho_A(a_2) &= \{\rho_A(d) \mid d \in a_1 + a_2\} \\ \rho_A(a_1) \cdot \rho_A(a_2) &= \{\rho_A(d) \mid d \in a_1 a_2\} \end{aligned}$$

the zero element is $\rho_A(0)$ and the fuzzy equality is $\bar{\rho}_A$.

Remark: In order to prove that the canonical inclusion from the image of a homomorphism is an injective fuzzy homomorphism, we recall the following result from [9]: given φ between A and B , there exists a unique crisp function f such that $\varphi(a, b) = \sigma(f(a), b)$. This f actually coincides with the crisp description h of φ , which satisfies $\varphi(a, h(a)) = 1$.

Lemma 19 Given φ between A and B , then the inclusion ι from $\text{Im } \varphi$ to B defined as $\iota(b, b') = \sigma(b, b')$ is an injective fuzzy homomorphism.

Theorem 20 Any complete fuzzy homomorphism φ from A to B can be canonically decomposed as $\varphi = \iota \circ \bar{\varphi} \circ \pi$ where π is the fuzzy projection from A to A/ρ_φ , ι is the inclusion from $\text{Im } \varphi$ to B , and $\bar{\varphi}$ is the isomorphism from A/ρ_φ to $\text{Im } \varphi$ defined as $\bar{\varphi}(\rho_\varphi(a), b) = \varphi(a, b)$, and the operations and the fuzzy equality in $\text{Im } \varphi$ being the corresponding restrictions of those in B .

PROOF: We will prove (ext1), (inj) and (surj) for $\bar{\varphi}$ since the rest of properties are straightforward:

$$\begin{aligned} \text{ext1 } \bar{\varphi}(\rho_\varphi(a), b) \wedge \bar{\varphi}(\rho_\varphi(a'), b) &= \\ \varphi(a, b) \wedge \varphi(a', b) &= \varphi(a, b) \wedge \varphi(a, h(a')) = \\ \varphi(a, b) \wedge \varphi(a, h(a')) \wedge \varphi(a', h(a')) &\leq \\ \sigma(b, h(a')) \wedge \varphi(a', h(a')) &\leq \varphi(a', b) = \bar{\varphi}(\rho_\varphi(a'), b) \end{aligned}$$

$$\text{inj } \bar{\varphi}(\rho_\varphi(a), b) \wedge \bar{\varphi}(\rho_\varphi(a'), b) = \varphi(a, b) \wedge \varphi(a', b) \leq \varphi(a, a') = \bar{\varphi}(\rho_\varphi(a), \rho_\varphi(a')).$$

surj For all $b \in \text{Im } \varphi$ there exists $a \in A$ such that $\varphi(a, b) = 1$ and then $\bar{\varphi}(\rho_\varphi(a), b) = 1$

Now, let us check that $\varphi = \iota \circ \bar{\varphi} \circ \pi$:

$$\begin{aligned} (\iota \circ \bar{\varphi} \circ \pi)(a, b) &= \\ &= \bigvee_{\substack{\rho_\varphi(a') \in A/\rho_\varphi \\ b' \in \text{Im } \varphi}} \left(\pi(a, \rho_\varphi(a')) \wedge \bar{\varphi}(\rho_\varphi(a'), b') \wedge \iota(b', b) \right) \\ &= \bigvee_{\substack{a' \in A \\ b' \in \text{Im } \varphi}} \left(\rho_\varphi(a, a') \wedge \varphi(a', b') \wedge \sigma(b', b) \right) \\ &\stackrel{(\text{ext2})}{\leq} \bigvee_{a' \in A} \left(\rho_\varphi(a, a') \wedge \varphi(a', b) \right) \\ &\stackrel{(\text{def } \rho_\varphi)}{=} \bigvee_{a' \in A} \left(\varphi(a, h(a')) \wedge \varphi(a', b) \right) \\ &\stackrel{(f1)}{=} \bigvee_{a' \in A} \left(\varphi(a, h(a')) \wedge \varphi(a', h(a')) \wedge \varphi(a', b) \right) \\ &\stackrel{(\text{part})}{\leq} \bigvee_{a' \in A} \left(\varphi(a, h(a')) \wedge \sigma(h(a'), b) \right) \stackrel{(\text{ext2})}{\leq} \varphi(a, b) \end{aligned}$$

Conversely, $\varphi(a, b) = \sigma(h(a), b) = \pi(a, \rho_\varphi(a)) \wedge \bar{\varphi}(\rho_\varphi(a), h(a)) \wedge \sigma(h(a), b) \leq (\iota \circ \bar{\varphi} \circ \pi)(a, b)$. $\square$

5 Fuzzy hyperideals and homomorphisms

This section is devoted to the fuzzy extension of the classical relation between crisp ideals and homomorphisms.

First of all, we adopt the definition of fuzzy hyperideal defined in [19]. Nevertheless, the superfluous conditions have been removed.

Definition 21 Let $(A, +, \cdot, 0)$ be a hyperring. A fuzzy subset μ of A is a **fuzzy hyperideal** if it satisfies: for all $a, b \in A$,

1. $\mu(a) \wedge \mu(b) \leq \mu(x)$ for all $x \in a - b$
2. $\mu(a) \vee \mu(b) \leq \mu(ab)$

The kernel relation in a hyperring can be expressed in terms of the kernel of the corresponding homomorphism. In this section we prove that the same occurs in the fuzzy case, due to the convenient properties including in our definition of fuzzy homomorphism.

Let $(A, +, \cdot, 0)$ and $(B, +, \cdot, 0)$ be hyperrings endowed with fuzzy equalities ρ and σ , respectively, such that $\sigma(a, b) = \sigma(-a, -b)$

Let us consider now the fuzzy kernel relation induced by φ in A , $\rho_\varphi \in [0, 1]^{A \times A}$, defined as $\rho_\varphi(a, a') = \varphi(a, h(a'))$, where h is the crisp description of φ .

Note that $\rho_\varphi(0) \in [0, 1]^A$ is a fuzzy subset of A and

$$\begin{aligned} \rho_\varphi(0)(a) &= \rho_\varphi(0, a) = \varphi(0, h(a)) = \sigma(0, h(a)) \\ &= \sigma(h(a), 0) = \varphi(a, 0) \end{aligned}$$

Theorem 22 Consider φ between A and B and the fuzzy kernel relation ρ_φ . Then $\rho_\varphi(0)$ is a fuzzy hyperideal.

PROOF: Firstly, observe that $\varphi(a, b) = \sigma(h(a), b) = \sigma(-h(a), -b) = \sigma(h(-a), -b) = \varphi(-a, -b)$ for all $a \in A$ and $b \in B$. On the other hand,

$$\begin{aligned} \rho_\varphi(0)(a) \wedge \rho_\varphi(0)(b) &= \varphi(0, h(a)) \wedge \varphi(0, h(b)) = \\ \varphi(0, h(a)) \wedge \varphi(0, -h(b)) &\leq \widehat{\varphi}(0, h(a) - h(b)) = \\ \widehat{\varphi}(0, h(a - b)) &= \bigwedge_{y \in h(a-b)} \varphi(0, y) = \\ \bigwedge_{x \in a-b} \varphi(0, h(x)) &= \bigwedge_{x \in a-b} \rho_\varphi(0)(x) \end{aligned}$$

So, $\rho_\varphi(0)(a) \wedge \rho_\varphi(0)(b) \leq \rho_\varphi(0)(x)$ for all $x \in a - b$.

Related to the multiplication,

$$\begin{aligned} \rho_\varphi(0)(a) &= \rho_\varphi(0, a) = \rho_\varphi(0, a) \wedge \rho_\varphi(b, b) \\ &\leq \rho_\varphi(0, a \cdot b) = \rho_\varphi(0)(a \cdot b) \end{aligned}$$

and analogously, $\rho_\varphi(0)(b) \leq \rho_\varphi(0)(a \cdot b)$. Thus,

$$\rho_\varphi(0)(a) \vee \rho_\varphi(0)(b) \leq \rho_\varphi(0)(a \cdot b) \quad \square$$

Once proven that the kernel of a fuzzy homomorphism between hyperrings is a fuzzy hyperideal, we consider whether the existing relationship between the fuzzy hyperideal and the congruence defined by the fuzzy homomorphisms is what one could expect.

In the crisp case, given an ideal I , a congruence is defined [8] by

$$a \equiv b \pmod{I} \quad \text{if and only if} \quad (a - b) \cap I \neq \emptyset$$

The natural form of fuzzifying this construction would be the following: if μ is a fuzzy hyperideal of A , the fuzzy relation $\rho_\mu \in [0, 1]^{A \times A}$ should be

$$\rho_\mu(a, a') = \bigvee_{x \in a - a'} \mu(x)$$

In a nutshell, the fuzzy subset $\rho_\varphi(0)$ plays the role of the kernel of the homomorphism in the crisp case.

Proposition 23 Consider φ between A and B and the fuzzy kernel relation ρ_φ . Then,

$$\rho_\varphi(a, a') = \bigvee_{x \in a - a'} \rho_\varphi(0)(x)$$

PROOF: Firstly we prove $\rho_\varphi(a, a') \leq \bigvee_{x \in a - a'} \rho_\varphi(0)(x)$

From Proposition 14,

$$\rho_\varphi(a, a') = \varphi(a, h(a')) = \bigvee_{b \in B} \varphi(a, b) \wedge \varphi(a', b) \quad (1)$$

On the other hand, for all $b \in B$,

$$\begin{aligned} \varphi(a, b) \wedge \varphi(a', b) &\leq \widehat{\varphi}(a - a', b - b) \\ &= \bigwedge_{x \in a - a'} \bigvee_{y \in b - b} \varphi(x, y) \wedge \bigwedge_{y \in b - b} \bigvee_{x \in a - a'} \varphi(x, y) \\ &\leq \bigwedge_{y \in b - b} \bigvee_{x \in a - a'} \varphi(x, y) \leq \bigvee_{x \in a - a'} \varphi(x, 0) \\ &= \bigvee_{x \in a - a'} \rho_\varphi(0)(x) \end{aligned}$$

Thus, $\bigvee_{b \in B} \varphi(a, b) \wedge \varphi(a', b) \leq \bigvee_{x \in a - a'} \rho_\varphi(0)(x)$ and by (1), one obtains the inequality required.

Now, we check that $\bigvee_{x \in a - a'} \varphi(x, 0) \leq \rho_\varphi(a, a')$. It suffices to prove that $\rho_\varphi(a, a')$ is an upper bound.

$$\begin{aligned} \rho_\varphi(0)(x) &= \rho_\varphi(x, 0) = \rho_\varphi(x, 0) \wedge \rho_\varphi(a', a') \\ &\leq \widehat{\rho}_\varphi(x + a', a') = \bigwedge_{z \in x + a'} \rho_\varphi(z, a') \leq \rho_\varphi(a, a') \quad \square \end{aligned}$$

6 Conclusions

In this paper, we have introduced a suitable notion of fuzzy homomorphism between hyperrings and have studied the results and isomorphism theorems which relate fuzzy homomorphisms between hyperrings, fuzzy congruences and fuzzy hyperideals.

References

- [1] R. Bělohávek. Determinism and fuzzy automata. *Information Sciences*, 143: 205–209, 2002.
- [2] M. Benado. Les ensembles partiellement ordonnés et le théorème de raffinement de Schreier. I. *Čechoslovak. Mat. Ž.*, 4(79):105–129, 1954.
- [3] I. P. Cabrera, P. Cordero, G. Gutiérrez, J. Martínez, and M. Ojeda-Aciego. Fuzzy congruence relations on nd-groupoids. *Intl J on Computer Mathematics* 86:1684–1695, 2009.
- [4] I. P. Cabrera, P. Cordero, G. Gutiérrez, J. Martínez, and M. Ojeda-Aciego. Congruence relations on some hyperstructures. *Annals of Math. and Artif. Intell.* Article in press (doi: 10.1007/s10472-009-9146-5).
- [5] M. Ćirić, J. Ignjatović, and S. Bogdanović. Uniform fuzzy relations and fuzzy functions. *Fuzzy Sets and Systems*, 160(8):1054–1081, 2009.
- [6] P. Corsini. A new connection between hypergroups and fuzzy sets. *Southeast Asian Bull. Math.*, 27(2):221–229, 2003.
- [7] B. Davvaz. Some results on congruences on semihypergroups. *Bulletin of the Malaysian Mathematical Sciences Society. Second Series*, 23(1):53–58, 2000.
- [8] B. Davvaz. Isomorphism theorems of hyperrings. *Indian J. Pure Applied Mathematics*, 35(3):321–331, 2004.
- [9] M. Demirci. Fuzzy functions and their applications. *J. Math. Anal. Appl.*, 252:495–517, 2000.
- [10] M. Demirci. Gradation of being fuzzy function. *Fuzzy Sets and Systems*, 119(3):383–392, 2001.
- [11] M. Demirci. Constructions of fuzzy functions based on fuzzy equalities. *Soft Computing*, 7(3):199–207, 2003.
- [12] G. Gentile. Multiendomorphisms of hypergroupoids. *International Mathematical Forum*, 1(9-12):563–576, 2006.
- [13] F. Klawonn. Fuzzy points, fuzzy relations and fuzzy function. In V. Novák and I. Perfilieva, editors, *Discovering World with Fuzzy Logic*, pages 431–453. Physica-Verlag, 2000.
- [14] M. Krasner. A class of hyperrings and hyperfields. *International Journal of Mathematics and Mathematical Sciences*, 6(2):307–311, 1983.
- [15] F. Marty. Sur une généralisation de la notion de groupe. In *8th Congress Math. Scandinaves*, pages 45–49, 1934.
- [16] J. Medina and M. Ojeda-Aciego and J. Ruiz-Calviño. Fuzzy logic programming via multilattices. *Fuzzy Sets and Systems*. N. 158, Vol. 6, Pág. 674–688, 2007.
- [17] T. Petković. Congruences and homomorphisms of fuzzy automata. *Fuzzy Sets and Systems*, 157:444–458, 2006.
- [18] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [19] J. Zhan, B. Davvaz, and K. P. Shum. A new view of fuzzy hypernear-rings. *Inf. Sci.*, 178(2):425–438, 2008.

ESTUDIO DE LA PROBABILIDAD DE ZADEH PARA T-NORMAS ARQUIMEDIANAS

J. Hernández¹ I. Montes² D. Martinetti² S. Montes²

¹ Massachusetts Institute of Technology, javierh@mit.edu

² Universidad de Oviedo, {uo182551,martinettidavide.uo,montes}@uniovi.es

Resumen

Una de las primeras formas de medir la probabilidad de un suceso borroso fue propuesta por Zadeh. A pesar de que desde entonces han surgido muchas otras alternativas, dicha definición sigue siendo considerada aún en la actualidad en algunos ámbitos. Cuando la intersección y la unión de dos conjuntos borrosos se define mediante la t-norma del mínimo y su t-conorma dual, respectivamente, esta medida es realmente una medida de probabilidad según los axiomas de Kolmogorov. Ésta fue la t-norma elegida inicialmente, pero evidentemente cualquier otra podría ser considerada dependiendo del contexto. En este trabajo se plantea una caracterización de las t-normas arquimedianas en función de su compatibilidad con dicho concepto de probabilidad.

Palabras Clave: probabilidad de Zadeh, t-norma arquimediana, t-norma estricta, t-norma nilpotente.

1 INTRODUCCIÓN

La falta de información asociada a todo experimento puede ser de dos tipos bien diferenciados: incertidumbre o imprecisión. Hablamos de incertidumbre cuando el espacio muestral está compuesto por una serie de alternativas bien especificadas, pero no sabemos cuál de ellas ha sido el resultado del experimento. Un ejemplo típico de incertidumbre es el resultado del lanzamiento de una moneda, conocemos con precisión las posibles alternativas (cara o cruz), pero no sabemos cual de las dos ocurrirá. La teoría de la probabilidad se encarga del estudio de este tipo de falta de información. Por

otro lado, hablamos de imprecisión cuando se sabe cuál de las alternativas ha ocurrido, pero ésta no puede ser descrita de forma precisa. Esto ocurriría, siguiendo con el ejemplo anterior, si vemos el resultado del lanzamiento de la moneda, pero ésta es muy antigua y está demasiado gastada. En realidad sabemos el resultado del experimento, pero no podemos describirlo con precisión, podríamos decir que el resultado “parece ser cruz” o afirmaciones por el estilo. La teoría que se encarga del estudio de la imprecisión es la de los conjuntos borrosos.

Aunque de naturaleza totalmente distinta, es posible que ambas faltas de información estén presentes en un mismo problema. Esto hace que la teoría de la probabilidad y la teoría de los conjuntos borrosos puedan y deban trabajar en conjunto en algunas ocasiones, con el fin de poder manejar de forma unificada y coherente la imprecisión y la incertidumbre. Este hecho ha sido puesto de manifiesto por muchos autores (ver, por ejemplo, [5, 9, 10, 11, 15, 19, 20]) desde la introducción del concepto de conjunto borroso en 1965 [18]. El primer intento de trabajar con ambas teorías de forma conjunta fue el de Loginov en 1966 [10]. Sin embargo su idea inicial no era natural puesto que para conjuntos borrosos no se verifica la ley del *tercero excluido*. El segundo intento fue el propuesto por Zadeh en 1968 [19], basándose en el hecho de que la probabilidad de un conjunto nítido puede ser vista como la esperanza de su función característica y definiendo, a partir de aquí la probabilidad de un conjunto borroso. Posteriormente muchas otras teorías han contribuido con nuevas formas de cuantificar la probabilidad de un conjunto borroso, dando lugar a medidas de probabilidad borrosas, medidas de posibilidad, etc. (ver por ejemplo, entre otras muchas, [3, 14, 15, 16]). A pesar de ello, el concepto de probabilidad de un conjunto borroso introducido por Zadeh es aún utilizado en algunas ocasiones (ver, por ejemplo, [1, 4, 6, 13, 16, 17]).

En [19], Zadeh consideró que la intersección de dos conjuntos borrosos venía definida mediante la t-norma

del mínimo y que la unión venía definida por su t-conorma dual, es decir, por el máximo. Bajo estas suposiciones, la medida introducida por Zadeh verificaba los clásicos axiomas de Kolmogorov, con lo cual era una medida de probabilidad en el sentido clásico. No obstante, esta propiedad no se mantiene en general para cualquier t-norma y su t-conorma dual asociada. El objetivo de este trabajo es caracterizar todas las t-normas arquimedianas de acuerdo con su compatibilidad con dicha axiomática. Así, se realiza un estudio general tanto para t-normas estrictas como nilpotentes. De dicho estudio se obtendrá como consecuencia inmediata la caracterización de los parámetros compatibles para algunas de las familias de t-normas más habituales (Frank, Yager, Aczél-Alsina, Dombi, Sugeno-Weber, Mayor-Torrens, Schweizer-Sklar y Hamacher).

Así, la organización de este trabajo queda como sigue. Comenzaremos con una primera sección en la que se recordarán los conceptos previos necesarios para la comprensión del resto del manuscrito. Posteriormente, en la Sección 3, se analizará el comportamiento respecto a la axiomática de Kolmogorov de las t-normas estrictas e idempotente. En la Sección 4 se realiza un planteamiento análogo para las t-normas nilpotentes. La aplicación de los estudios anteriores para algunas familias interesantes de t-normas se realiza en la Sección 5. Finalmente se termina el trabajo con una sección de conclusiones y puntos abiertos.

2 CONCEPTOS BÁSICOS

En esta sección introduciremos algunos conceptos básicos necesarios en el desarrollo del resto del trabajo. Muchos de ellos son ampliamente conocidos, por lo que nuestro objetivo no es sólo recordarlos al lector, sino fijar la notación utilizada. Comenzaremos recordando el concepto de probabilidad analizado en este trabajo. En el análisis de dicho concepto será imprescindible tener en cuenta las diversas formas de definir la unión y la intersección de dos conjuntos borrosos. Puesto que dichas definiciones vienen dadas a través de las t-conormas y las t-normas, respectivamente, un repaso de dichos conceptos será también necesario.

Zadeh ([19]) cuantificó la probabilidad de un suceso del tipo “el día está caliente”, “ x es *aproximadamente* igual a 5”, “en las veinte tiradas de una moneda hubo *muchas* más caras que cruces”, es decir, de sucesos representados por conjuntos borrosos. Su definición de probabilidad de un suceso borroso se basaba en la idea de que en el caso clásico, la probabilidad de un suceso medible A puede expresarse como $P(A) = \int_A dP = \int_{\Omega} \chi_A(w) dP(w) = E[\chi_A]$, donde χ_A representa la función característica de A , es decir, $\chi_A(w) = 1$ si $w \in A$ y cero en otro caso. Así pues,

de forma análoga, definió la probabilidad de un suceso borroso A con función de pertenencia medible, a través de la integral de Lebesgue-Stieltjes:

$$P(A) = \int_{\Omega} A(w) dP(w) = E[A],$$

donde Ω representa el referencial y $A(w)$ el valor de la función de pertenencia del punto w al conjunto A .

En este primer acercamiento al problema, consideró el operador mínimo, que denotaremos por T_M , para definir la intersección y el operador máximo para la unión, es decir,

$$\left. \begin{aligned} A \cap B(w) &= \min(A(w), B(w)) \\ A \cup B(w) &= \max(A(w), B(w)) \end{aligned} \right\}, \quad \forall w \in \Omega.$$

Considerando estas operaciones demostró que su definición de probabilidad sobre una T_M -tribu de algún espacio no vacío Ω era una medida de probabilidad en el sentido clásico de Kolmogorov, es decir, que

- Ax1) $P(\Omega) = 1$.
- Ax2) $P(A) \geq 0$, para todo suceso A .
- Ax3) $P(A \cup B) = P(A) + P(B)$ para cualesquiera A y B con $A \cap B = \emptyset$.

No obstante, las operaciones de intersección y unión de conjuntos nítidos, admiten otras muchas extensiones en el caso de conjuntos borrosos, que vienen dadas a través de las t-normas y las t-conormas. Recordemos que una t-norma T es una aplicación de $[0, 1]^2$ en $[0, 1]$ con elemento neutro 1, conmutativa, asociativa y no decreciente en ambas componentes. Así, en general la intersección de dos conjuntos borrosos viene definida por $A \cap B(w) = T(A(w), B(w))$, $\forall w \in \Omega$, para cualquier t-norma T fijada.

El primer operador utilizado por Zadeh, el mínimo, es además el ejemplo más conocido de t-norma y será denotado aquí por T_M . Otras t-normas clásicas son la t-norma del producto T_P , definida por $T_P(x, y) = x \cdot y$ y la t-norma de Łukasiewicz T_L , definida por $T_L(x, y) = \max(x + y - 1, 0)$.

Recordemos también que una t-conorma tiene exactamente las mismas propiedades que una t-norma salvo que el elemento neutro es en este caso el 0. Así, la unión de dos conjuntos borrosos se define en general, para cualquier t-conorma S , como sigue: $A \cup B(w) = S(A(w), B(w))$, $\forall w \in \Omega$.

En este trabajo vamos a considerar que la intersección viene definida por una t-norma cualquiera y la unión por su t-conorma dual, es decir,

$$S(x, y) = 1 - T(1 - x, 1 - y), \forall (x, y) \in [0, 1]^2.$$

Con este planteamiento vamos a ver como dada un t-norma cualquiera, no siempre se verifica que P sea

una medida de probabilidad sobre un T -tribu (T -probabilidad). Recordemos que dada una t -norma T , una subclase $\mathcal{T}$ de las partes borrosas de Ω se dice que es una T -tribu si [2, 7]: *i)* $\emptyset \in \mathcal{T}$; *ii)* el complementario de todo conjunto de $\mathcal{T}$ está en $\mathcal{T}$; y para toda sucesión $(A_n)_{n \in \mathbb{N}}$ en $\mathcal{T}$, se tiene que $\cap_{n \in \mathbb{N}} A_n \in \mathcal{T}$. En particular vamos a caracterizar el comportamiento de algunas clases importantes de t -normas respecto a la axiomática de Kolmogorov, o lo que es lo mismo, analizar el comportamiento de la función P definida por Zadeh para otras T -tribus distinta de la T_M -tribu. Para ello, debemos concluir esta sección de conceptos previos con un análisis más detallado de dichas clases de t -normas. Un estudio completo sobre t -normas y t -conormas, en general, puede verse en [8].

Una forma de obtener nuevas t -normas a partir de una dada es a partir de los automorfismos. Un automorfismo (o $[0, 1]$ -automorfismo si existiese ambigüedad) es toda aplicación estrictamente creciente $\varphi : [0, 1] \rightarrow [0, 1]$ tal que $\varphi(0) = 0$, $\varphi(1) = 1$. La inversa de un automorfismo es también un automorfismo y así, dada una t -norma, su φ -transformación T_φ definida por $T_\varphi(x, y) = \varphi^{-1}(T(\varphi(x), \varphi(y)))$ es también una t -norma. En general la φ -transformación da lugar a una t -norma distinta de la inicial, aunque por ejemplo en el caso de mínimo, de tiene que $(T_M)_\varphi = T_M$, para todo automorfismo φ .

En este trabajo nos ocuparemos en particular de un tipo de t -normas que son φ -transformaciones de las principales t -normas (mínimo, producto y Łukasiewicz). Todas ellas constituyen, como veremos, dos clases particulares de t -normas continuas. Recordemos que una t -norma continua es aquella que es continua en cada componente y que esta propiedad se preserva por φ -transformaciones.

Antes de pasar a definir tales clases de t -normas, necesitamos recordar dos conceptos más: elemento idempotente y divisor de cero. Decimos que un valor $x \in (0, 1)$ es un divisor de cero de una t -norma cualquier T si existe otro valor $y \in (0, 1)$ tal que $T(x, y) = 0$. Por otro lado, decimos que un valor $x \in [0, 1]$ es un elemento idempotente de T si $T(x, x) = x$. Los valores 0 y 1 son pues elementos idempotentes para cualquier t -norma.

En base a estos elementos podemos clasificar ciertas familias de t -normas. Así:

- Una t -norma T es idempotente si todos los puntos del intervalo $(0, 1)$ son elementos idempotentes de T .
- Una t -norma T es nilpotente si es continua y todo los elementos del intervalo $(0, 1)$ son divisores de cero de T .

- Una t -norma T es estricta si es continua y estrictamente creciente en cada componente.
- Una t -norma T es arquimediana si $T(x, x) < x, \forall x \in (0, 1)$.

Se tiene que la única t -norma idempotente es el mínimo. Por otro lado, todas las t -normas nilpotentes y estrictas con t -normas continuas arquimedianas y también el recíproco es cierto, es decir, toda t -norma continua arquimediana es o bien nilpotente o bien estricta. Ahora bien, una t -norma es estricta, respectivamente nilpotente, si y sólo si es una φ -transformación de la t -norma producto, respectivamente Łukasiewicz. Con lo cual tenemos caracterizadas todas la t -normas continuas arquimedianas como transformaciones de la t -norma producto o de la t -norma de Łukasiewicz.

3 T-NORMAS ERICTAS Y NILPOTENTES

La verificación o no de los axiomas de Kolmogorov por parte del concepto de probabilidad radica en la verificación del tercer axioma. Al haber demostrado Zadeh [19] que con la t -norma del mínimo este concepto verificaba los tres axiomas y no depender de la t -norma elegida más que el tercero, en el que interviene la unión y la intersección de dos conjuntos borrosos, la comprobación de la compatibilidad de un t -norma con dicha axiomática se reduce a comprobar si verifica el tercer axioma. Teniendo esto en cuenta y que una t -norma estricta no tiene ningún divisor de cero, se puede demostrar fácilmente el siguiente resultado.

PROPOSICIÓN 3.1 *Dada una t -norma estricta T , un referencial Ω y una T -tribu $\mathcal{T}$ de Ω . La aplicación $P : \mathcal{T} \rightarrow [0, 1]$ definida por*

$$P(A) = \int_{\Omega} A(w) dP(w) = E[A],$$

para todo $A \in \mathcal{T}$ con función de pertenencia medible es una T -probabilidad, es decir, verifica los tres axiomas de Kolmogorov con respecto a la t -norma T .

A pesar del buen comportamiento de las t -normas estrictas respecto a esta propiedad, no ocurre lo mismo con las t -normas nilpotentes. Vamos a ver, por ejemplo, como la transformación a través del automorfismo $\varphi(x) = x^2$ de la t -norma de Łukasiewicz no verifica el tercer axioma de Kolmogorov.

EJEMPLO 3.2 *Según acabamos de comentar en la sección anterior, si consideramos $\varphi(x) = x^2$, la t -norma $(T_L)_\varphi$, que denotaremos por simplicidad por T_{L2} , es una t -norma nilpotente. Dado un referencial unipuntual $\Omega = \{w\}$, para el que evidentemente*

$P(w) = 1$ y los conjuntos borrosos A y B definidos por $A(w) = 0'5$ y $B(w) = 0'4$, se tiene que

$$A \cap B(w) = \sqrt{\max\{A(w)^2 + B(w)^2 - 1, 0\}} = 0,$$

con lo que $A \cap B = \emptyset$. Sin embargo,

$$A \cup B(w) = 1 - \sqrt{\max\{(0'5)^2 + (0'6)^2 - 1, 0\}} = 1,$$

con lo que $P(A \cup B) = 1$, pero

$$P(A) + P(B) = \int_{\Omega} A(w) dP(w) + \int_{\Omega} B(w) dP(w) = 0'9.$$

Según lo visto anteriormente no todas las t-normas nilpotentes son compatibles con esta axiomática. De hecho, tal como vamos a demostrar, la única que es compatible es la de Łukasiewicz. Para ello, necesitamos un par de lemas previos. El primero será dado en general para t-normas nilpotentes y es necesario establecer algunos conceptos previos a su planteamiento. El segundo será dado para el caso particular de t-normas nilpotentes que verifican una determinada condición de compatibilidad.

DEFINICIÓN 3.3 ([12]) Sea T una t-norma cualquiera.

- Para todo $z \in [0, 1]$, se define el z -nivel de T , que denotaremos por $L(T, z)$, como el conjunto $L(T, z) = \{(x, y) \in [0, 1]^2 | T(x, y) = z\}$.
- Dados dos puntos $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$ de $[0, 1]^2$ decimos que son T -equivalentes, y lo denotamos por $p_1 \approx_T p_2$, si existe un $z \in [0, 1]$ tal que ambos puntos pertenecen al mismo z -nivel.
- Llamaremos soporte de T , y lo denotaremos por $\text{supp}(T)$ a la unión de todos los z -niveles, para todo $z \in (0, 1]$.
- Dos rectángulos R y R' en $[0, 1]^2$, es decir, dos conjuntos de la forma $\{x_1, x_2\} \times \{y_1, y_2\} \subseteq [0, 1]^2$ con $x_1 < x_2$ e $y_1 < y_2$, se dicen T -equivalentes si sus vértices correspondientes son T -equivalentes. Se dicen T -alineados si al menos tres de sus vértices son T -equivalentes.
- Decimos que T satisface la condición de Reidmeister si para cada par de rectángulos en el soporte de T que están T -alineados son T -equivalentes.

LEMA 3.4 ([12]) Toda t-norma nilpotente satisface la condición de Reidmeister.

Si la t-norma es compatible con el tercer axioma de Kolmogorov, hemos demostrado otros resultados particulares, que aparecen recogidos en el siguiente lema.

LEMA 3.5 Sea T una t-norma nilpotente verificando la siguiente condición

$$T(x, y) = 0 \Rightarrow T(1 - x, 1 - y) = 1 - x - y.$$

Entonces:

- El conjunto de los divisores de cero están incluido en $\{(x, y) \in [0, 1]^2 | x + y \leq 1\}$.
- $\sup\{z \in [0, 1] | T(z, z) = 0\} = 1/2$ si y sólo si $T = T_L$.

Como consecuencia de los Lemas 3.4 y 3.5 se puede demostrar el siguiente resultado que caracteriza totalmente el comportamiento de las t-normas nilpotentes en el problema tratado en este trabajo.

TEOREMA 3.6 La única t-norma nilpotente que hace que la probabilidad P verifique el tercer axioma de Kolmogorov es la de Łukasiewicz.

Un resumen de los principales resultados presentados en esta sección puede verse a continuación.

Familia	Compatible
Idempotente (T_M)	Sí
Estricta ($(T_P)_\varphi$)	Para todo automorfismo φ
Nilpotente ($(T_L)_\varphi$)	Si y sólo si $\varphi = Id$

4 FAMILIAS DE T-NORMAS

Vamos a utilizar los resultados de la sección anterior para analizar el comportamiento de las t-normas continuas arquimedianas de algunas de las familias más conocidas (para más detalles ver, por ejemplo, el capítulo 4 de [8]). Por simplicidad, diremos que una t-norma es compatible, cuando al considerarla para definir la intersección, y su dual para definir la unión, de dos conjuntos difusos, la probabilidad definida por Zadeh sea una T -probabilidad. En cada caso, vamos además a recordar la expresión que tienen las t-normas continuas arquimedianas de dicha familia.

Así, sea T una t-norma continua arquimediana,

- Si T es de la familia de Schweizer-Sklar, es decir, si $T_\lambda^{SS}(x, y)$ viene definido por
 - $T_P(x, y)$ si $\lambda = 0$,
 - $(\max((x^\lambda + y^\lambda - 1), 0))^{1/\lambda}$ si $\lambda \in (-\infty, \infty) - \{0\}$,

entonces T_λ^{SS} es compatible para todo $\lambda \in (-\infty, 0] \cup \{1\}$.

- Si T es de la familia de Hamacher, es decir, si $T_\lambda^H(x, y)$ viene definido por

- 0 si $\lambda = 0$ y $x = y = 0$,
- $\frac{xy}{\lambda + (1-\lambda)(x+y-xy)}$ en otro caso con $\lambda \in [0, \infty)$,

entonces T_λ^H es siempre compatible.

- Si T es de la familia de Frank, es decir, si $T_\lambda^F(x, y)$ viene definido por

- $T_P(x, y)$ si $\lambda = 1$,
- $T_L(x, y)$ si $\lambda = \infty$,
- $\log_\lambda \left(1 + \frac{(\lambda^x - 1)(\lambda^y - 1)}{\lambda - 1} \right)$ si $\lambda \in (0, \infty) - \{1\}$,

entonces T_λ^F es siempre compatible.

- Si T es de la familia de Yager, es decir, si $T_\lambda^Y(x, y)$ viene definido por

- $\max(1 - ((1-x)^\lambda + (1-y)^\lambda)^{1/\lambda}, 0)$ si $\lambda \in (0, \infty)$,

entonces sólo T_1^Y es compatible.

- Si T es de la familia de Aczél-Alsina, es decir, si $T_\lambda^{AA}(x, y)$ viene definido por

- $e^{-((- \log x)^\lambda + (- \log y)^\lambda)^{1/\lambda}}$ si $\lambda \in (0, \infty)$,

entonces T_λ^{AA} es siempre compatible.

- Si T es de la familia de Dombi, es decir, si $T_\lambda^D(x, y)$ viene definido por

- $\left(1 + ((\frac{1-x}{x})^\lambda + (\frac{1-y}{y})^\lambda)^{1/\lambda} \right)^{-1}$ si $\lambda \in (0, \infty)$,

entonces T_λ^D es siempre compatible.

- Si T es de la familia de Sugeno-Weber, es decir, si $T_\lambda^{SW}(x, y)$ viene definido por

- $T_P(x, y)$ si $\lambda = \infty$,
- $\max\left(\frac{x+y-1+\lambda xy}{1+\lambda}, 0\right)$ si $\lambda \in (-1, \infty)$,

entonces sólo T_∞^{SW} y T_0^{SW} son compatibles.

- La única t-norma de la familia de Mayor-Torrens que es continua y arquimediana es la de Łukasiewicz y por tanto, según el Teorema 3.6, es compatible.

La información anterior se puede resumir en forma de tabla, como sigue:

Familia	Idempotente ó Arquim. cont.	Compatible
Schweizer-Sklar $\lambda \in [-\infty, \infty]$	$\lambda \in [-\infty, \infty)$	$\lambda \in [-\infty, 0]$ ó $\lambda = 1$
Hamacher $\lambda \in [0, \infty]$	$\lambda \in [0, \infty)$	$\lambda \in [0, \infty)$
Frank $\lambda \in [0, \infty]$	$\lambda \in [0, \infty]$	$\lambda \in [0, \infty]$
Yager $\lambda \in [0, \infty]$	$\lambda \in (0, \infty]$	$\lambda \in \{1, \infty\}$
Aczél-Alsina $\lambda \in [0, \infty]$	$\lambda \in (0, \infty]$	$\lambda \in (0, \infty]$
Dombi $\lambda \in [0, \infty]$	$\lambda \in (0, \infty]$	$\lambda \in (0, \infty]$
Sugeno-Weber $\lambda \in [-1, \infty]$	$\lambda \in (-1, \infty]$	$\lambda \in \{0, \infty\}$
Mayor-Torrens $\lambda \in [0, 1]$	$\lambda \in \{0, 1\}$	$\lambda \in \{0, 1\}$

5 CONCLUSIONES

En este trabajo hemos demostrado que dada una t-norma cualquiera T , no siempre se verifica que P sea una medida de probabilidad sobre una T -tribu (T -probabilidad). En particular, vemos como esto es cierto para las t-normas idempotentes y estrictas, pero no así para las nilpotentes, para las que sólo es cierto en el caso particular de la t-norma de Łukasiewicz. Dentro de las familias más relevantes y utilizando los estudios anteriores, hemos demostrado para qué miembros de las mismas se verifica la propiedad anterior y para cuales no. Como consecuencia de esto, presentamos una familia, la de Frank, cuya compatibilidad es perfecta, es decir, en las que todos sus miembros satisfacen que P es una T -probabilidad. Algunas otras de las familias analizadas son totalmente compatibles para todos sus elementos continuos y arquimedianos o idempotentes (el rango de definición del parámetro λ en la segunda y la tercera columna coincide), pero no para todos sus elementos (el rango de definición de λ en la segunda columna es un subconjunto estricto del considerado en la primera columna).

Como futuros trabajos, nos gustaría completar el presente estudio con el análisis de todas las t-normas continuas, es decir, considerar también el caso de las sumas ordinales.

Agradecimientos

La investigación que se presenta en este trabajo ha sido subvencionada parcialmente por el Proyecto MTM2007-61193, concedido por el Ministerio de Educación y Ciencia. Este proyecto está financiado en parte con fondos FEDER.

Referencias

- [1] J.F. Baldwin, J. Lawry, T.P. Martin. A note on the conditional probability of fuzzy subsets of a continuous domain. *Fuzzy Sets and Systems* **96** (1998) 211–222.

- [2] D. Butnariu, E.P. Klement. *Triangular Norm-based Measures and Games with Fuzzy Coalitions*, Kluwer Academic Publishers, 1993.
- [3] D. Dubois, H. Prade. Toward fuzzy analysis: Integration and derivation of fuzzy functions. In "Fuzzy Algebra, Analysis, Logics," Tech. Rep. TR-EE 78/13, Part C. Purdue Univ., Lafayette, Indiana, 1978.
- [4] T. Flaminio, L. Godo. A logic for reasoning about the probability of fuzzy events. *Fuzzy Sets and Systems* **158** (2007) 625–638.
- [5] M.A. Gil. Statistical management of fuzzy elements in random experiments. Part 1: A discussion on treating fuzziness as a kind of randomness. *Information Sciences* **69** (1993) 229–242.
- [6] S. Janssens, B. De Baets, H. De Meyer. Bell-type inequalities for quasi-copulas. *Fuzzy Sets and Systems* **148** (2004) 263–278.
- [7] E.P. Klement. Construction of fuzzy σ -algebras using triangular norms. *J. Math. Anal. Appl.* **85** (1982) 543–565.
- [8] E.P. Klement, R. Mesiar, E. Pap. *Triangular Norms*, Kluwer Academic Publishers, 2000.
- [9] M. Laviolette, J.W. Seaman, J.D. Barrett, W.H. Woodall. A Probabilistic and Statistical View of Fuzzy Methods. *Technometrics* **37** (1995) 249–261.
- [10] V.I. Loginov. Probability Treatment of Zadeh Membership Functions and Their Use in Pattern Recognition. *Engineering Cybernetics* (1966) 68–89.
- [11] M. Navara. Probability theory of fuzzy events. In *EUSFLAT/LFA 2005 Proceedings* Barcelona, Spain, September 7-9, 2005, pp. 325–329.
- [12] M. Petrík, P. Sarkoci. Convex combinations of nilpotent triangular norms. *J. Math. Anal. Appl.* **350** (2009) 271–275.
- [13] J. Pykacz, B. D’Hooghe. Bell-type inequalities in fuzzy probability calculus. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **9** (2001) 263–275.
- [14] D. Ralescu. Fuzzy Probabilities and their Applications to Statistical Inference. In: B. Bouchon-Meunier, R.R. Yager, L.A. Zadeh (Eds.), *Advances in Intelligent Computing-IPMU94*, Lecture Notes in Computer Science, vol. 945, Springer, Paris, 1995, pp. 217–222.
- [15] N.D. Singpurwalla, J.M. Booker. Membership Functions and Probability Measures of Fuzzy Sets. *Journal of the American Statistical Association* **99** (2004) 867–877.
- [16] R.R. Yager. A note on probabilities of fuzzy events. *Information Sciences* **18** (1979) 113–129.
- [17] R.R. Yager. Drawing reasonable conclusions from information under similarity modelled contexts. *Int. J. Granular Computing, Rough Sets and Intelligent Systems* **1** (2009) 81–104.
- [18] L.A. Zadeh. Fuzzy Sets. *Inform. Control* **8** (1965) 338–353.
- [19] L.A. Zadeh. Probability Measures of Fuzzy Events. *Journal of Mathematical Analysis and Applications* **23** (1968) 421–427.
- [20] L.A. Zadeh. Discussion: Probability Theory and Fuzzy Logic Are Complementary Rather Than Competitive. *Technometrics* **37** (1995) 271–276.

On the logics of similarity-based approximate and strong entailments

Francesc Esteva¹ Lluís Godo¹ Ricardo O. Rodríguez² Thomas Vetterlein³

¹ IIIA - CSIC, 08193 Bellaterra, Spain – {esteva,godo}@iiia.csic.es

² University of Buenos Aires, 1428 Buenos Aires, Argentina – ricardo@dc.uba.ar

³ Medical University of Vienna, 1090 Vienna, Austria – Thomas.Vetterlein@meduniwien.ac.at

Abstract

We consider two kinds of logics for approximate reasoning: one is weaker than classical logic and the other is stronger. In the first case, we are led by the principle that from given premises we can jump to conclusions which are only approximately (or possibly) correct. In the second case, which was not considered so far, in contrast, we follow the principle that conclusions must remain (necessarily) correct even if the premises are slightly changed. In this paper we recall the definitions and characterizations of the first logic, and we investigate the basic properties of the second logic, as well as its soundness and completeness with respect to Ruspini's semantics based on fuzzy similarity relations.

1 Introduction

In order to open up a topic for systematic investigation in a formal way, we are required to fix a framework of reference; we need to choose a model to which our considerations refer. Roughly speaking, we often proceed as follows. To discuss a topic means to discuss the relative or absolute properties of a manifold of different situations. To make the subject under consideration precise a particular set of characteristic properties needs to be selected. We are then led to a collection of situations which vary with respect to certain properties and are identified otherwise. The properties are in turn identified with the set of situations in which they hold.

To reason in this framework we may use classical propositional logic. In this logic we find what we need: yes-no propositions modeled by subsets, and connectives which are interpreted by set-theoretic operations like meet, join, and complement. Furthermore, implication corresponds to set-theoretic inclusion.

When using classical propositional logic for practical purposes the problem however might occur that different properties might not always be delimitable from each other in practice. This is in particular always the case if the universe of discourse results from an infinite process of stepwise refinement, in which case it has a continuous character. In fact, often we work in metric spaces like the reals. But then it might no longer be convenient to model implicational relationships as strict set-theoretic inclusion; the relation of inclusion is sensitive to arbitrary small changes. This is undesirable if bounds of propositions are given only roughly.

A possible solution is to make the relationship between properties and the pool of possible interpretations in a model more flexible. We may require for instance that inferences should not break down, but be tolerant, with regard to small changes. In this paper, we develop this idea in a specific way, continuing a series of earlier works.

We will actually propose two different ways to proceed and in order to illustrate why we do so, consider the following example, which is actually inappropriate because it refers to first-order logic but might serve for the illustrative purpose. Let a set of objects be distinguished by their sizes. We may consider two different kinds of relations, $\leq$ meaning “smaller or equal than” and $<$ meaning “strictly smaller than”, in order to infer information about one object, say Q , from information about another one, say P . However, a statement like $a \leq b$, where a and b are the sizes of P and Q , respectively, might be practically undecidable if P and Q are similar. The principle of *approximate reasoning* introduces tolerance in the following way: letting $a \leq b$ still hold if a is slightly larger than b , but endowing this statement with a truth-like degree which is 1 if $a \leq b$ actually holds but decreases continuously to 0 when a becomes larger than b . The case of $<$ is, in a sense, dual. Here we may require that a is by a sufficient amount smaller than b so that we are

able to make a clear statement. The principle of *strong reasoning* says that a statement must be tolerant, that is, remain valid, under small changes. It is assigned a truth degree as well, which is 1 if there is no tolerance, and the smaller the greater the tolerance is.

Logics of approximate reasoning have been studied quite intensively. The initial approach is due to Ruspini [6]. Approaches in a logical style have been developed in several papers of three of the authors and others [2, 3, 5, 4]. The most flexible approach is provided by multi-modal logics [3]. Here, the universe of discourse W is a similarity space and a modal operator $\Diamond_\alpha$ is added to the language of classical propositional logic (CPL) for each α from a fixed set C of truth degrees. $\Diamond_\alpha$ is interpreted by the α -neighborhoods U_α in W . Then the implication “from φ it follows ψ to the degree α ” holds according to the principles of approximate reasoning if $[\varphi] \subseteq U_\alpha([\psi])$ where $[\varphi]$ and $[\psi]$ are the sets of interpretations satisfying φ and ψ respectively. In a similar way, this implication holds according to the principles of strong reasoning if $U_\alpha([\varphi]) \subseteq [\psi]$.

We see in particular that both our above concerns can be fulfilled in the multi-modal approach. Here however we first consider what is called the logic of approximate entailment, LAE, developed in [5]. This logic considers graded implications like $\varphi >_\alpha \psi$, which correspond to inclusions $[\varphi] \subseteq U_\alpha([\psi])$ when interpreted in the above logic. Furthermore, in complete analogy we develop the logic of strong entailment, denoted LSE. This logic deals with graded implications $\varphi \succ_\alpha \psi$, which correspond to inclusions $U_\alpha([\varphi]) \subseteq [\psi]$.

This paper is organized as follows. Section 2 contains necessary background on fuzzy similarity measures and their associated implication and consistency measures proposed by Ruspini [6]. In Section 3 we describe the approximate and strong similarity-based entailments and show their basic properties. In Section 4 we recall the logic LAE of approximate entailment, providing its syntax and semantics. Finally, in Section 6, we introduce a new logic for the strong entailment LSE and we propose an axiomatization. We finish with some remarks about future work.

2 Preliminaries: fuzzy similarity relations and Ruspini’s measures

Let us consider a propositional language L built up from a finite set of propositional variables $p_1, \dots, p_n$ and the *constants* $\top$ and $\perp$ by means of the binary operators $\wedge, \vee$ and the unary operator $\neg$. Propositions will be denoted by greek letters $\varphi, \psi, \dots$. The set of propositions will be denoted by $\mathcal{P}$ and the set of

classical interpretations of $\mathcal{P}$ will be denoted by Ω , we will also call them possible worlds. We will use the expressions $w \models \varphi$ and $\varphi \models \psi$ to respectively denote that w satisfies φ (or that ω is a model φ) and that ψ is a logical consequence of φ in classical propositional logic. We will also denote by $[\varphi]$ the subset of interpretations of Ω that satisfy φ , i.e. $[\varphi] = \{\omega \in \Omega \mid \omega \models \varphi\}$.

Following Ruspini [6], the starting point of our framework is to assume that a possible world or state of a system may resemble more to some worlds than to another ones, and this basic fact may help us to evaluate to what extent a partial description (a proposition) may be close or similar to some other. This is modelled by assuming a fuzzy binary relation $S : \Omega \times \Omega \rightarrow [0, 1]$ on the set Ω of classical interpretations of $\mathcal{P}$ capturing a suitable notion of similarity is given. Usual properties that are considered in the literature for such fuzzy binary relations are:

Reflexivity: $S(u, u) = 1$ for all $u \in \Omega$

Separability: $S(u, v) = 1$ iff $u = v$

Symmetry: $S(u, v) = S(v, u)$, for all $u, v \in \Omega$

$\otimes$ -Transitivity: $S(u, v) \otimes S(v, w) \leq S(u, w)$, for all $u, v, w \in \Omega$

where $\otimes$ is a t-norm. The reflexivity property establishes that the similarity degree of any world with itself has the highest value. Separability is a bit stronger since it forbids to have $S(u, v) = 1$ for $u \neq v$. Symmetry has a clear meaning, and $\otimes$ -Transitivity is a relaxed form of transitivity since it establishes $S(u, v) \otimes S(v, w)$ as a lower bound for $S(u, w)$. Note that $S(u, v) = S(v, w) = 1$ implies $S(u, w) = 1$. Reflexive and symmetric fuzzy relations are often called *closeness* or *proximity* relations, while those further satisfying $\otimes$ -transitivity are usually called *$\otimes$ -transitive similarity relations* or *$\otimes$ -indistinguishability operators*. In this paper we will use the term $\otimes$ -similarity relations to refer to these class of relations, and sometimes we will also require them to be separable. Note that Zadeh called simply *similarity relations* to min-transitive similarity relations in the previous sense.

Dually, one can think of $1 - S$ as a kind of metric on worlds, indeed, when $\otimes$ is Lukasiewicz t-norm and S is a separating $\otimes$ -transitive similarity relation, then $1 - S$ is a metric, c.f. [7].

When trying to extend the similarity on worlds to propositions, Ruspini defined in [6] the following two measures

$$I_S(\varphi \mid \psi) = \inf_{\omega \models \psi} \sup_{\omega' \models \varphi} S(\omega, \omega')$$

$$C_S(\varphi \mid \psi) = \sup_{\omega \models \psi} \sup_{\omega' \models \varphi} S(\omega, \omega')$$

which are the lower and upper bounds respectively of the resemblance or proximity degree between φ and ψ . Indeed, I_S is an implication (i.e. inclusion-like) measure, while C_S is a consistency (i.e. intersection-like) measure and thus it is symmetric.

The value of $I_S(\varphi \mid \psi)$ provides the measure to what extent φ is close to be true given ψ for granted and the similarity between worlds represented by S . In particular, when S is separating and the set of worlds is finite then, $I_S(\varphi \mid \psi) = 1$ iff $\psi \models \varphi$. Moreover, if S is $\otimes$ -transitive then I_S is $\otimes$ -transitive as well [6], i.e. the inequality

$$I_S(\chi \mid \varphi) \otimes I_S(\varphi \mid \psi) \leq I_S(\chi \mid \psi)$$

holds for any propositions φ , ψ and χ , capturing a form of generalised modus ponens.

On the other hand, the value of $C_S(\varphi \mid \psi)$ provides the measure of what extent φ can be considered compatible with the available knowledge ψ . In particular, in the finite case and with S satisfying separation property, $C_S(\varphi \mid \psi) = 1$ iff $\psi \not\models \neg\varphi$. Observe that, when the propositional language is finitely generated and ψ is equivalent to a maximal consistent set of propositions, both measures coincide because there is a unique world w such that $w \models \psi$. In addition, it is easy to show that, given a fixed χ , the measure $C_S(\cdot \mid \chi)$ is a possibility measure [2] since the following identities hold true:

1. $C_S(\top \mid \chi) = 1$
2. $C_S(\perp \mid \chi) = 0$
3. $C_S(\varphi \vee \psi \mid \chi) = \max(C_S(\varphi \mid \chi), C_S(\psi \mid \chi))$.

By duality, one can define the following similarity-based measure N_S as follows:

$$N_S(\psi \mid \varphi) = 1 - C_S(\neg\psi \mid \varphi) = \inf_{w \models \neg\psi} \inf_{w' \models \varphi} 1 - S(w, w').$$

The following are relevant properties of N_S measures which will be used in the next section and which can be easily verified:

1. $N_S(\cdot \mid \varphi)$ is a necessity measure for each p such that $\varphi \not\models \perp$, i.e.
 - $N_S(\top \mid \varphi) = 1$
 - $N_S(\perp \mid \varphi) = 0$
 - $N_S(\psi \wedge \chi \mid \varphi) = \min(N_S(\psi \mid \varphi), N_S(\chi \mid \varphi))$
2. $N_S(\psi \mid \varphi \vee \chi) = \min(N_S(\psi \mid \varphi), N_S(\psi \mid \chi))$
3. $N_S(\psi \mid \varphi) = N_S(\neg\varphi \mid \neg\psi)$
4. if $\varphi \models \psi$ then $N_S(\varphi \mid \chi) \leq N_S(\psi \mid \chi)$ and $N_S(\chi \mid \psi) \leq N_S(\chi \mid \varphi)$
5. if $N_S(\psi \mid \varphi) > 0$ then $\varphi \models \psi$

3 Approximate and strong entailment relations

Given a $\otimes$ -similarity relation on Ω , in this section we will first recall from [5] a corresponding family of graded approximate entailment relations $\models_S^\alpha$ and then we will introduce a family of strong entailment relations $\approx_S^\alpha$, both indexed by values $\alpha \in [0, 1]$, and being respectively weaker and stronger notions than the classical logical entailment $\models$.

A (graded) approximate satisfaction relation $\models_S^\alpha \subset \Omega \times \mathcal{P}$, for each $\alpha \in [0, 1]$ by

$$\omega \models_S^\alpha \varphi \quad \text{iff there exists a model } \omega' \text{ of } \varphi \text{ which is } \alpha\text{-similar to } \omega, \text{ i.e. } S(\omega, \omega') \geq \alpha$$

The approximate satisfaction relation can be extended in the natural way over to an *approximate entailment* relation $\models_S^\alpha \subseteq \mathcal{P} \times \mathcal{P}$: a proposition φ entails a proposition ψ at degree α , written $\varphi \models_S^\alpha \psi$, if each model of φ is an approximate model of ψ at level α , that is,

$$\varphi \models_S^\alpha \psi \quad \text{iff } w \models_S^\alpha \psi \text{ for all model } w \text{ of } \varphi$$

It is easy to check that $\varphi \models_S^\alpha \psi$ holds iff any of the following equivalent conditions hold:

- $I_S(\psi \mid \varphi) \geq \alpha$
- $[\varphi] \subseteq U_\alpha([\psi])$

where $U_\alpha([\psi]) \subseteq \Omega$ is the neighborhood of radius α of the set of models of ψ , that is, $U_\alpha([\psi]) = \{w \in \Omega \mid \text{there exists } w' \text{ s.t. } w' \models \psi \text{ and } S(w', w) \geq \alpha\}$.

The characterizing properties of this graded entailment relation are (the reader is referred to [3] for further details):

- (1) **Nestedness:**
if $\varphi \models_S^\alpha \psi$ and $\beta \leq \alpha$ then $\varphi \models_S^\beta \psi$;
- (2) **$\otimes$ -Transitivity:**
if $\varphi \models_S^\alpha \chi$ and $\chi \models_S^\beta \psi$ then $\varphi \models_S^{\alpha \otimes \beta} \psi$;
- (3) **Reflexivity:** $\varphi \models_S^1 \varphi$;
- (4) **Right weakening:**
if $\varphi \models_S^\alpha \psi$ and $\psi \models \chi$ then $\varphi \models_S^\alpha \chi$;
- (5) **Left strengthening:**
if $\varphi \models \chi$ and $\chi \models_S^\alpha \psi$ then $\varphi \models_S^\alpha \psi$;
- (6) **Left OR:**
 $\varphi \vee \chi \models_S^\alpha \psi$ iff $\varphi \models_S^\alpha \psi$ and $\chi \models_S^\alpha \psi$;
- (7) **Right OR:**
if χ has a single model,
 $\chi \models_S^\alpha \varphi \vee \psi$ iff $\chi \models_S^\alpha \varphi$ or $\chi \models_S^\alpha \psi$.

Now we are interested in another kind of graded entailment, a *strong entailment*, which is in a sense dual

to the above approximate entailment. We define a (graded) satisfaction relation $\models_S^\alpha \subseteq \Omega \times \mathcal{P}$, for each $\alpha \in [0, 1]$ by

$$\omega \models_S^\alpha \varphi \quad \text{iff } \omega' \models \varphi \text{ for each } \omega' \text{ s.t. } S(\omega, \omega') \geq \alpha$$

that is, if any model w' in the neighborhood (of radius α) of w is a model of φ . If $\omega \models_S^\alpha \varphi$ we say that w is an *strong model* (at level α) of φ . The natural corresponding entailment relation comes defined in the following way: a proposition φ strongly entails a proposition ψ at degree α , written $\varphi \models_S^\alpha \psi$, if each approximate model of φ at level α is a model of ψ that is,

$$\varphi \models_S^\alpha \psi \text{ iff, for all } w, w \models_S^\alpha \varphi \text{ implies } w \models \psi$$

An equivalent definition, making use of the neighborhoods, would simply be

$$\varphi \models_S^\alpha \psi \text{ iff } U_\alpha([\varphi]) \subseteq [\psi]$$

Moreover, in a similar way the approximate entailment was linked to the implication measure I_S , this strong graded entailment is related to the consistency measure C_S , or equivalently, to the necessity measure N_S . Indeed, the following proposition relate in a precise way the strong entailment $\models_S^\alpha$ and the N_S measure.

Proposition 1 *Assume the language is finitely generated and let $\alpha > 0$. Then $\varphi \models_S^\alpha \psi$ iff $N_S(\psi \mid \varphi) > 1 - \alpha$.*

Proof: $\rightarrow$) If $\varphi \models_S^\alpha \psi$, it means that if w' is such that $S(w, w') \geq \alpha$ for some $w \models \varphi$, then $w' \models \psi$. Therefore, if $w' \not\models \psi$ then $S(w, w') < \alpha$ for all $w \models \varphi$, i.e. $\max_{w \models \varphi} S(w, w') < \alpha$ for all $w' \models \neg\psi$, i.e. $\max_{w' \models \neg\psi} \max_{w \models \varphi} S(w, w') < \alpha$, i.e. $C(\neg\psi \mid \varphi) < \alpha$.

$\leftarrow$) $N_S(\psi \mid \varphi) > 1 - \alpha$ amounts to $C(\neg\psi \mid \varphi) < \alpha$, i.e. for all $w \not\models \psi$ and for all $w' \models \varphi$ we have $S(w, w') < \alpha$. So, if $w \models \varphi$ is such that $S(w, w') \geq \alpha$ for some other world w' it must necessarily be $w' \models \psi$, hence $\varphi \models_S^\alpha \psi$. $\square$

Main properties of this strong graded entailment relation induced by a $\otimes$ -similarity measure S are:

- (1) **Nestedness:**
if $\varphi \models_S^\alpha \psi$ and $\beta \geq \alpha$ then $\varphi \models_S^\beta \psi$;
- (2) $\varphi \models_S^0 \psi$ iff either $\models \neg\varphi$ or $\models \psi$
- (3) $\varphi \models_S^1 \psi$ iff $\varphi \models \psi$
- (4) **min-Transitivity:**
if $\varphi \models_S^\alpha \psi$ and $\psi \models_S^\beta \chi$ then $\varphi \models_S^{\min(\alpha, \beta)} \chi$;
- (5) **Left OR:**
 $\varphi \vee \chi \models_S^\alpha \psi$ iff $\varphi \models_S^\alpha \psi$ and $\chi \models_S^\alpha \psi$;
- (6) **Right AND:**
 $\chi \models_S^\alpha \varphi \wedge \psi$ iff $\chi \models_S^\alpha \varphi$ and $\chi \models_S^\alpha \psi$.

- (7) **Contraposition:**
If $\varphi \models_S^\alpha \psi$ then $\neg\psi \models_S^\alpha \neg\varphi$
- (8) **Restricted $\otimes$ -Transitivity:**
If φ, ψ, χ have a single model then
if $\varphi \models_S^{\alpha \otimes \beta} \neg\psi$ then either $\varphi \models_S^\alpha \neg\chi$ or $\chi \models_S^\beta \neg\psi$

Two interesting derived properties from (3) and (4) are:

- Right weakening:**
if $\varphi \models_S^\alpha \psi$ and $\psi \models \chi$ then $\varphi \models_S^\alpha \chi$;
Left strengthening:
if $\varphi \models \chi$ and $\chi \models_S^\alpha \psi$ then $\varphi \models_S^\alpha \psi$;

Indeed one can show that these properties characterize these graded strong entailments, but the proof is not included for space reasons.

4 A logic of approximate entailment

In [5], it has been developed a graded conditional logic approach, that we will call LAE, to encode in the language syntactical objects representing approximate entailments $\varphi \models_S^\alpha \psi$. To do so, binary (graded) modal operators are introduced (under some restrictions, e. g. nested modal formulas are not allowed, and the language is finitely generated) and appropriate semantics in terms of similarity Kripke structures are given.

Through the rest of the paper, let us fix a countable set $C \subset [0, 1]$ of *similarity degrees*; we require that C contains 0 and 1 and that for any non-zero $\alpha \in C$ there is a largest $\beta < \alpha$ in C . Furthermore, let us fix an operation $\otimes : C \times C \rightarrow C$ which is commutative, associative, in both arguments isotone, and has 1 as its neutral element.

The propositional language of LAE results from extending the propositional language $\mathcal{P}$ introduced in Section 2 with a family $\{>_\alpha\}_{\alpha \in C}$ of binary operators. *Conditional formulas* are built as follows:

- If $\varphi \in \mathcal{P}$ then φ is also a conditional formula.
- If $\varphi, \psi \in \mathcal{P}$ then, for every $\alpha \in C$, $\varphi >_\alpha \psi$ is an *atomic conditional formula*.
- If φ and ψ are conditional formulas then $\neg\varphi$ and $\varphi \rightarrow \psi$ are conditional formulas

Note that in this language, nested conditional formulas are not allowed.

Semantics are specified by $\otimes$ -similarity Kripke models $\mathcal{M} = (W, S, e)$, where W is a set of worlds, $S : W \times W \rightarrow C$ is a $\otimes$ -similarity on W and $e : W \times \mathcal{P} \rightarrow \{0, 1\}$ is such that $e(w, \cdot)$ is a usual Boolean interpretation of propositions of $\mathcal{P}$, with the extra condition that $e(w, \cdot) \neq e(w', \cdot)$ when $w \neq w'$. Then we define the satisfaction relation by stipulating $(\mathcal{M}, w) \models \varphi$ if

$e(w, \varphi) = 1$ for $\varphi \in \mathcal{P}$, and is extended to atomic conditional formulas by defining

$$(\mathcal{M}, \omega) \models \varphi >_{\alpha} \psi \quad \text{if} \quad [\varphi] \subseteq U_{\alpha}([\psi]),$$

or equivalently, if $I_S(\psi \mid \varphi) \geq \alpha$, where now U_{α} refers to neighborhoods in W (i.e. $[\psi] = \{w \in W \mid e(w, \psi) = 1\}$). The rest of the conditions for compound conditional formulas are the usual ones. Note that the notion of satisfiability for $>_{\alpha}$ is independent of any particular world, i.e. it is a global notion. The condition of satisfiability makes clear that in the object language $\varphi >_{\alpha} \psi$ represents lower bounds of $I_S(\psi \mid \varphi)$.

The axioms and rules of LAE are those of Classical Propositional logic (CPL) plus the following ones, where α and β represent any values of C :¹

- (N) $\varphi >_{\alpha} \psi \rightarrow \varphi >_{\beta} \psi$ if $\beta \leq \alpha$
- (CS) $\varphi >_1 \psi \rightarrow (\varphi \rightarrow \psi)$
- (EX) $\varphi >_0 \psi$
- (B) $\chi >_{\alpha} \chi' \rightarrow \chi' >_{\alpha} \chi$, if χ and χ' are m.e.c.'s
- (4) $(\varphi >_{\alpha} \psi) \wedge (\psi >_{\beta} \chi) \rightarrow \varphi >_{\alpha \otimes \beta} \chi$
- (LO) $(\varphi \vee \psi >_{\alpha} \chi) \leftrightarrow (\varphi >_{\alpha} \chi) \wedge (\psi >_{\alpha} \chi)$
- (RO) $(\chi >_{\alpha} \varphi \vee \psi) \leftrightarrow (\chi >_{\alpha} \varphi) \vee (\chi >_{\alpha} \psi)$,
if χ is a m.e.c.

and the following inference rule:

- (RK) From $\varphi \rightarrow \psi$ infer $\varphi >_1 \psi$

In [5], the author proves, among other results, that LAE is complete with respect to the class of similarity models (W, S) where S is a separating $\otimes$ -similarity, when C is finite. Moreover, if the axiom (CS) is dropped, then one gets completeness w.r.t. similarity models where the similarity relation is not necessarily separating.

Notice that a very related approach using metrics instead of similarities has been proposed in [8].

5 Logic of strong entailment

Let us define the logic LSE in a similar way, but not exactly, as we did for the logic LAE. Formulas of LSE are built from propositions of $\mathcal{P}$ by introducing new *graded implications* which are a triples consisting of two propositions $\varphi, \psi \in \mathcal{P}$ and a value $\alpha \in C$, denoted

$$\varphi \succ_{\alpha} \psi.$$

¹In these axioms we use m.e.c. to denote a maximal elementary conjunction, that is, a conjunction where each propositional variable appears either in positive or negative form (remember that we are assuming a finite set of propositional variables).

A *statement* of LSE is built up from graded implications of LSE by means of the binary operators $\wedge, \vee$ and the unary operator $\neg$. The additional operator $\rightarrow$ will be used as an abbreviation for $\neg \cdot \vee \cdot$.

Semantics for LSE are given by Kripke models $\mathcal{M} = (W, S, e)$, where W is a set of worlds, $e : W \times \mathcal{P} \rightarrow \{0, 1\}$ is such that, for all $w \in W$, $e(w, \cdot)$ is a Boolean interpretation of propositional formulas of $\mathcal{P}$, and $S : W \times W \rightarrow C$ is a *separating symmetric* fuzzy relation over possible worlds valued on C . The satisfaction of a graded implication $\varphi \succ_{\alpha} \psi$ in a model $\mathcal{M} = (W, S, e)$, is defined as

$$\mathcal{M} \models \varphi \succ_{\alpha} \psi \quad \text{if} \quad U_{\alpha}([\varphi]) \subseteq [\psi],$$

where again $U_{\alpha}([\varphi]) = \{w \in W \mid S(w', w) \geq \alpha \text{ for some } w' \in W \text{ s.t. } e(w', \varphi) = 1\}$. According to Proposition 1, when $\alpha > 0$, $\mathcal{M} \models \varphi \succ_{\alpha} \psi$ iff $N_S(\psi \mid \varphi) > 1 - \alpha$. The satisfaction of statements is defined classically from the satisfaction of graded implications.

Finally, a *theory* of LSE is a finite set of statements of LSE. We say that a theory $\mathcal{T}$ *semantically entails* a statement Φ , written $\mathcal{T} \models_{\text{LSE}} \Phi$, if Φ is satisfied by any model satisfying every element of $\mathcal{T}$.

We shall axiomatise the logic LSE in the following way.

Definition 1 *The following graded implications are axioms of LSE for any $\varphi, \psi, \chi \in \mathcal{P}$ and $\alpha, \beta \in C$:*

- (A1) $\perp \succ_0 \varphi$
- (A2) $\varphi \succ_1 \psi$ where φ, ψ are such that $\varphi \rightarrow \psi$ is a tautology of CPL
- (A3) $\varphi \succ_0 \top$
- (A4) $(\varphi \succ_{\alpha} \psi) \wedge (\varphi \succ_{\alpha} \chi) \rightarrow (\varphi \succ_{\alpha} \psi \wedge \chi)$
- (A5) $(\varphi \succ_{\alpha} \chi) \wedge (\psi \succ_{\alpha} \chi) \rightarrow (\varphi \vee \psi \succ_{\alpha} \chi)$
- (A6) $(\varphi \succ_{\alpha} \psi) \rightarrow (\neg \psi \succ_{\alpha} \neg \varphi)$
- (A7) $(\varphi \succ_{\beta} \psi) \wedge (\psi \succ_{\alpha} \chi) \rightarrow (\varphi \succ_{\min(\beta, \alpha)} \chi)$
- (A8) $(\varphi \succ_{\beta} \psi) \rightarrow (\varphi \succ_{\alpha} \psi)$ where $\alpha \geq \beta$
- (A9) $(\varphi \succ_0 \psi) \rightarrow (\varphi \succ_1 \perp) \vee (\top \succ_1 \psi)$

In addition, given a tautology of CPL, the statement resulting from a uniform replacement of the atoms by graded implications of LSE is an axiom. Moreover, the following is a rule of LSE for any statements Φ, Ψ :

- (MP) from Φ and $\Phi \rightarrow \Psi$ derive Ψ

A proof of a statement Φ from a theory $\mathcal{T}$ is defined as usual. If it exists, we write $\mathcal{T} \vdash_{\text{LSE}} \Phi$.

Theorem 1 (Completeness) *Let $\mathcal{T}$ be a consistent theory of LSE, and let Φ be a statement. Then $\mathcal{T} \vdash_{\text{LSE}} \Phi$ if and only if $\mathcal{T} \models_{\text{LSE}} \Phi$.*

Proof: The “only if” part is evident. As for the “if” part, notice first that it is easy to check that $\mathcal{T} \vdash_{\text{LSE}} \Phi$ iff $\mathcal{T} \cup AX^*$ proves Φ just using the axioms and rules of classical propositional logic (using graded implications as propositional variables), written $AX^* \cup \mathcal{T} \vdash_{\text{CPL}} \Phi$, where AX^* is the set of instances of the LSE axioms.

Assume that $\mathcal{T} \not\vdash_{\text{LSE}} \Phi$. Then, there is a $\{0,1\}$ -evaluation which is a model of $AX^* \cup \mathcal{T}$ and $v(\Phi) = 0$. We are going to define a similarity-Kripke model $M = (W, S, e)$ such that M is a model of $\mathcal{T}$ but not of Φ . We take $W = \Omega$, the set of propositional models of $\mathcal{P}$ and we define $S : \Omega \times \Omega \rightarrow [0, 1]$ as

$$S(w, w') = \max\{t \in C \mid v(\underline{w} \succ_t \neg \underline{w}') = 0\}$$

where, for each $w \in \Omega$, $\underline{w}$ denotes the proposition whose only model is w . Actually, $\underline{w}$ always exists since we are assuming that the propositional language $\mathcal{P}$ is generated by a finite set of variables. For each $\varphi \in \mathcal{P}$, define $[\varphi] = \{w \in \Omega \mid w \models \varphi\}$.

Claim 1: Let $t > 0$. Then, $v(\varphi \succ_t \psi) = 1$ iff for all $w, w' \in W$ if $w \models \varphi$ and $w' \models \neg\psi$ then $S(w, w') < t$.

Proof: Since φ and ψ are logically equivalent to $\bigvee_{w \models \varphi} \underline{w}$ and to $\bigwedge_{w' \models \neg\psi} \neg \underline{w'}$ and v satisfies all the axiom instances in AX^* , we have that $v(\varphi \succ_t \psi) = 1$ iff $v(\bigvee_{w \models \varphi} \underline{w} \succ_t \bigwedge_{w' \models \neg\psi} \neg \underline{w'}) = 1$ iff $v(\underline{w} \succ_t \neg \underline{w'}) = 1$ for all $w, w' \in W$ such that $w \models \varphi$ and $w' \models \neg\psi$. But according to the definition of S above, $v(\underline{w} \succ_t \neg \underline{w'}) = 1$ iff $S(w, w') < t$. $\square$

Claim 2: $v(\varphi \succ_t \psi) = 1$ iff $U_t([\varphi]) \subseteq [\psi]$, where U_t is defined w.r.t. S .

Proof: Assume $t > 0$. Then, by Claim 1, $v(\varphi \succ_t \psi) = 1$ iff for all $w, w' \in W$ if $w \models \varphi$ and $w' \models \neg\psi$ then $S(w, w') < t$, which is equivalent to the condition that for all $w, w' \in W$ if $w \models \varphi$ and $S(w, w') \geq t$ then $w' \models \psi$, that is, $U_t([\varphi]) \subseteq [\psi]$.

If $t = 0$, by axiom (A9), $v(\varphi \succ_0 \psi) = 1$ iff either $v(\varphi \succ_1 \perp) = 1$ or $v(\top \succ_1 \psi) = 1$, that is, iff either $[\varphi] = U_1([\varphi]) = \emptyset$ or $\Omega = U_1([\top]) \subseteq [\psi]$. So, noticing that $U_0(\emptyset) = \emptyset$, we have that in both cases $U_0([\varphi]) \subseteq [\psi]$. Conversely, if $U_0([\varphi]) \subseteq [\psi]$ and $[\varphi] \neq \emptyset$, then $\Omega = U_0([\varphi]) \subseteq [\psi]$. $\square$

Finally, define $M = (\Omega, S, e)$ where e is defined as $e(w, \varphi) = 1$ if $w \models \varphi$, $e(w, \varphi) = 0$ otherwise, for each $\varphi \in \mathcal{P}$, and the satisfaction condition for graded implications in M as expected:

$$M \models \varphi \succ_t \psi \text{ iff } U_t([\varphi]) \subseteq [\psi] \text{ iff } v(\varphi \succ_t \psi) = 1$$

Therefore, from Claim 2, it is clear that $M \models \Psi$ for all $\Psi \in \mathcal{T}$ but $M \not\models \Phi$, hence $\mathcal{T} \not\vdash_{\text{LSE}} \Phi$. $\square$

6 Conclusions and future work

In this paper we have been concerned with two notions of graded similarity-based entailment arising in the framework of practical reasoning when looking for robust inferences in the presence of small variations either in the premise or in the conclusion². The main contribution of the paper is the study and axiomatization of the so-called logic of strong entailment LSE, which incorporates at the object level constructs as $\varphi \succ_\alpha \psi$ capturing the notion that ψ is a strong consequence (at degree α) of φ . We have shown soundness and completeness results for special kinds of derivations in such a logic but a lot of work remains to be done to fully exploit this logical framework.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful comments and acknowledge partial support of the bilateral Austrian-Spanish project HA2008-0017 and the Eurocores-LogICCC ESF project LoMoReVI. Esteva and Godo also acknowledge partial support of the Spanish project FFI2008-03126-E/FILO and Rodríguez acknowledges the projects CyT-UBA X484 and the research CONICET program PIP 12-200801-02543 2009-2011.

References

- [1] I. Bloch, J. Lang. Towards mathematical “morphologies”, *Technologies for Constructing Intelligent Systems*, Vol. 2, 367-380, Springer-Verlag, 2002.
- [2] D. Dubois, H. Prade, F. Esteva, P. Garcia, L. Godo. A logical approach to interpolation based on similarity relations, *Int. J. Approx. Reasoning* **17** (1997), 1 - 36.
- [3] F. Esteva, P. Garcia, L. Godo, R. O. Rodríguez. A modal account of similarity-based reasoning, *Int. J. Approx. Reasoning* **16** (1997), 235 - 260.
- [4] L. Godo, R. O. Rodríguez. Logical approaches to fuzzy similarity-based reasoning: an overview, in: G. Della Riccia et al. (eds.), *Preferences and similarities*, CISM Courses and Lectures 504, Springer, pp. 75-128, 2008.
- [5] R. O. Rodríguez. *Aspectos formales en el Razonamiento basado en Relaciones de Similitud Borrosas*, Ph. D. Thesis, Technical University of Catalonia, 2002.
- [6] E. Ruspini. On the semantics of fuzzy logic. *International Journal of Approximate Reasoning*, 5:45-88, 1991.
- [7] E. Trillas and L. Valverde. An inquiry into indistinguishability operators. In H.J. Skala et al. (Eds.), *Aspects of Vagueness*. Reidel, Dordrecht Boston, 231-256, 1984.
- [8] T. Vetterlein, K.-P. Adlassnig. T-norm-based fuzzy logics and logics for reasoning under vagueness. *Proc. of IFSA-EUSFLAT 2009*, pp. 1085-1090.

²Analogous notions of approximate and strong entailments correspond in the field of mathematical morphology to the so-called dilation and erosion operators, see e.g. [1].

CONSULTAS CON COMPARADORES DIFUSOS EN ALGORITMOS DE MINERÍA DE DATOS: EN SQL SERVER 2008 *

Angélica Urrutia¹, Claudio Gutiérrez-Soto² y Juan Méndez¹

¹Universidad Católica del Maule, aurrutia@ucm.cl.

²Universidad del Bío-Bío, cogutier@ubiobio.cl

Resumen

En este artículo se presenta una extensión de algoritmos de Minería de Datos existentes en SQL Server 2008, utilizando comparadores difusos de posibilidad que permitirán al usuario final contar una herramienta la cual aporta información útil, oportuna; así como también conocimiento de las variables que influyen de manera directa en los indicadores de gestión. Los aportes principales de este trabajo son; en primer lugar el análisis de los algoritmos que pueden ser extendidos utilizando comparadores difusos, segundo un conjunto de pasos y la aplicación de la propuesta en un caso real.

Palabras claves: Minería de datos, Comparadores difusos, Consultas complejas con algoritmos de Minería de Datos.

1. INTRODUCCIÓN

Nuestra propuesta considera una implementación que utiliza un caso de estudio, que extienden las consultas clásicas a *comparadores difusos de posibilidad* utilizando la lógica difusa, de tal forma de prestar un mejor servicio de información al usuario final que requiere análisis de datos e indicadores de gestión. Las consultas se ejecutan sobre la implementación de los algoritmos Árboles de decisión, Bayes Naive, y Clústeres de Minería de Datos (MD), los cuales están incluidos en SQL Server 2008.

Algunos artículos analizados muestran como la *lógica difusa* se ha utilizado para extender las Bases de Datos Relacionales (BDR) a Bases de Datos Relacionales Difusas (BDRD) [6]. Una propuesta de los comparadores difusos aplicados al resultado de datos obtenidos por un clústeres de Minería de Datos sobre un caso del área de turismo se encuentra en [1], aquí los autores extienden el FSQ de [4] para utilizarlo en consulta difusas potenciando los resultados de los datos obtenidos por un

Clúster. Otras extensiones de conjuntos difusos aplicados a casos de Minería de Datos se encuentran en [2].

Por otro lado, la teoría de conjuntos difusos parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia al conjunto a través de un número real entre 0 y 1 [6, 8]. Así, se introduce el concepto de conjunto o subconjunto difuso asociado a un determinado valor lingüístico, definido por una palabra, adjetivo o etiqueta lingüística A . Para cada conjunto o subconjunto difuso se define una *función de pertenencia o inclusión* $\mu_A(u)$, donde se define un conjunto difuso A sobre un universo de discurso U (dominio ordenado) y es un conjunto de pares dado por: $A = \{\mu_A(u) / u : u \in U, \mu_A(u) \in [0,1]\}$, Donde, μ es la llamada función de pertenencia y $\mu_A(u)$ es el *grado de pertenencia* del elemento u al conjunto difuso A . Este grado oscila entre los extremos 0 y 1, considerando que $\mu_A(u) = 0$, indica que u no pertenece en absoluto al conjunto difuso A y que $\mu_A(u) = 1$, indica que u pertenece totalmente al conjunto difuso A . El mismo concepto se aplica a los comparadores de necesidad y posibilidad expuestos en [3, 4, 6].

Considerando lo anteriormente expuesto, presentamos en los siguientes apartados la propuesta de una arquitectura para procesos de Minería de Datos y consultas con comparadores clásicos y difusos, además de analizar el resultado diferentes escenarios con los algoritmos de Minería de Datos que más se ajusten al problema. La implementación de las consultas tiene el objetivo de proponer una extensión de comparadores, siempre y cuando los resultados entregados por el algoritmo lo permitan. Finalmente se analizan ambos resultados de las consulta y se analizan sus ventajas.

2. COMPARADORES CLÁSICOS Y DIFUSOS

Los operadores de comparación clásicos son; igual, mayor que, menor que, mayor o igual que, menor o igual que y distinto, los cuales se pueden utilizar tanto para comparar números como para comparar textos y fechas.

* Los resultados mostrados en este trabajo son financiados por proyecto interno 2009-2010 de la UCM.

Una extensión de los comparadores clásicos son los comparadores difusos los que han sido definidos como *Comparadores Difuso Generalizado* del modelo GEFRED [4]. Aquí se define un tipo de comparador general basado en *comparador clásico* existente ($=$, $>$, $<$...), el único requisito que se establece es que el *Comparador Difuso* debe respetar los resultados de los comparadores clásicos cuando se comparan distribuciones de posibilidad que expresan valores *crisp*.

También, en [4] se proponen otros comparadores (*Mayor difuso*, *Mayor ó Igual difuso*, *Mucho mayor difuso*) en su versión de posibilidad y necesidad (ver Tabla 1), así como la comparación de distribuciones de posibilidad sobre dominios subyacentes no ordenados. Cabe destacar que al igual que en SQL, los comparadores difusos son implementados en un servidor FSQl que se presentan en [3,4,6] y pueden comparar una columna (o atributo) con una constante o dos columnas del mismo tipo. Los comparadores de necesidad son más restrictivos que los de posibilidad, por lo que su grado de cumplimiento es siempre menor que el grado de cumplimiento obtenido por su correspondiente comparador de posibilidad.

Tabla 1: Comparadores difusos.

Comparador Difuso	Significado
FEQ, NFEQ	Posiblemente, Necesariamente Igual
FGT, NFGT	Posiblemente, Necesariamente Mayor que
FGEQ, NFGEQ	Posiblemente, Necesariamente Mayor o Igual que
FLT, NFLT	Posiblemente, Necesariamente Menor que
FLEQ, NFLEQ	Posiblemente, Necesariamente Menor o Igual que
MGT, NMGT	Posiblemente, Necesariamente Mucho Mayor que
MLT, NMLT	Posiblemente, Necesariamente Mucho Menor que

En general, los comparadores de necesidad exigen que la condición sea satisfecha con cierto grado, de necesidad aunque no sea completamente, mientras los comparadores de posibilidad miden en qué posibilidad de medida (o grado) es posible que la condición se cumpla.

3. COMPARADORES DIFUSOS Y SU IMPLEMENTACIÓN

Este apartado tiene por objetivo presentar la implementación de los comparadores difusos que fueron necesarios para el desarrollo de esta investigación. La implementación de cada comparador difuso está programada en código SQL para SQL Server 2008. Los comparadores difusos de posibilidad implementados son los que muestra la Tabla 1. Al igual que en [4] la sigla de

cada comparador es implementado de la siguiente forma: FEQ posiblemente igual (Fuzzy Equal), FGT posiblemente mayor que (Fuzzy Greater Than), FGEQ posiblemente mayor o igual (Fuzzy Greater Equal), FLT posiblemente menor que (Fuzzy Less Than), FLEQ posiblemente menor o igual (Fuzzy Less Equal), MGT Posiblemente mucho mayor (Much Greater Than) y MLT Posiblemente mucho menor (Much Less Than). A modo de ejemplo, en este trabajo sólo se presenta el comparador FLEQ, el resto de los comparadores se encuentran en [5].

Consideremos que, este grado se utiliza para obtener una medida de compatibilidad de $\mu_R(x)$ en el cual éste es posiblemente *más pequeño o igual* en la distribución de posibilidad $\pi_S(x)$ para todo $x \in U$. En el filtrado difuso la consulta FSQl devolverá las tuplas de la base de datos (BD) con un grado 1 cuando $B_R \leq C_S$; con un grado entre 0 y 1 cuando $B_R > C_S$ y $A_R < D_S$; y con un grado 0 para los otros casos (Ver Figura 1).

El grado de posibilidad del comparador difuso FLEQ [3,4] se define de la siguiente forma:

$$CDEG(R \text{ FLEQ } S) = \begin{cases} 1 & \text{para } B_R \leq C_S \\ \frac{D_R - A_S}{(B_S - A_S) - (C_R - D_R)} & \text{para } B_R > C_S \text{ y } A_R < D_S \\ 0 & \text{para } A_R \geq D_S \end{cases}$$

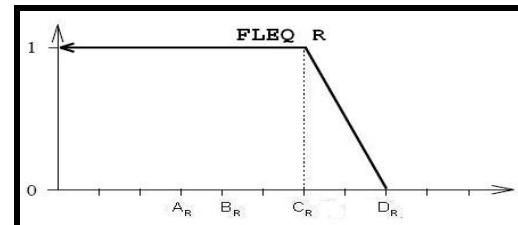


Figura 1: Representación gráfica del comparador FLEQ.

A continuación se presenta el código para SQL Server de la función FLEQ, a modo de ejemplo, la cual implementa el comparador difuso como una función de SQL Server.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION FLEQ
(@crisp FLOAT, @F1 FLOAT, @F4 FLOAT)
RETURNS FLOAT
AS BEGIN
    DECLARE @valor FLOAT
```

```

IF (@crisp <= @F1)
  SET @valor=1;
ELSE IF (@crisp < (@F1+@F4))
  SET @valor=ROUND((@F1 + @F4 -
@crisp)/@F4,4);
ELSE SET @valor=0;
RETURN @valor;
END
GO

```

4. ARQUITECTURA PROPUESTA PARA CONSULTAS FLEXIBLES EN MINERÍA DE DATOS

La arquitectura propuesta para el análisis de información (Ver Figura 2), permite satisfacer *indicadores de Gestión de Información* (GI), obtenidos desde la aplicación de un proceso de Minería de Datos. A dicho resultado, se le aplican consultas utilizando comparadores clásicos y difusos como una extensión.

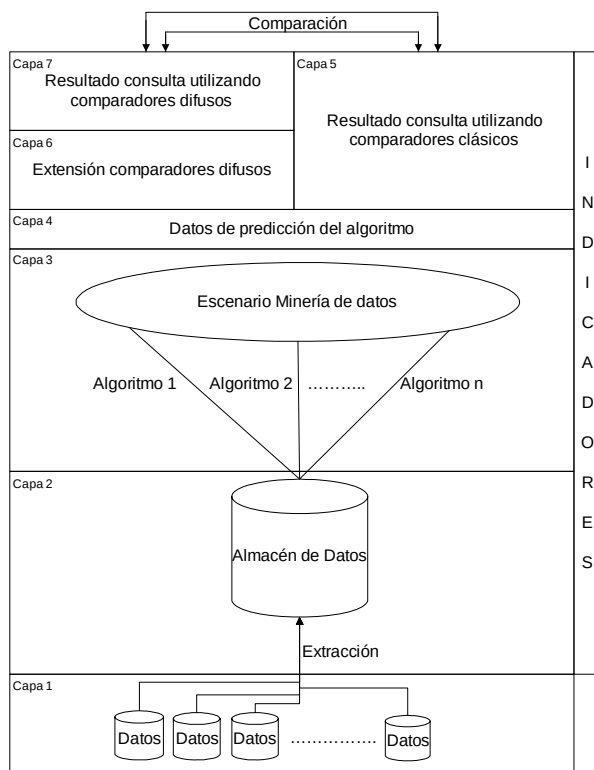


Figura 2: Arquitectura para procesos de minería de datos y consultas con comparadores clásicos y difusos.

La aplicación de Minería de Datos que utiliza comparadores clásicos y difusos, está sustentada en una

taxonomía de 7 capas, lo que permite comparar, qué aportes o en qué casos se obtiene un mejor resultado de consultas. Las capas se inician desde los datos existentes en una base de datos fuente extraídos y almacenados en un Almacén de Datos. A partir de aquí, se pueden implementar distintos escenarios y aplicar uno a más algoritmos de minería de datos, para posteriormente ser consultados con comparadores clásicos y comparadores difusos. Finalmente, ambos resultados son analizados desde el punto de vista descriptivo, lo cual entrega información valiosa al usuario final desde el punto de vista de la gestión de la información.

Cada capa que compone esta arquitectura tiene que realizarse en forma secuencial, sólo las consultas pueden ser tantas como el usuario final las requiera. Como en todo proceso, la principal tarea es recopilar la especificación de requerimientos. En este proceso debe quedar claro, cuáles son las necesidades de gestión del usuario final, para poder comenzar a trabajar cada una de las capas propuestas. A continuación se define cada capa.

Capa 1: En esta primera capa se encuentran todos los orígenes de datos, que pueden ser: un archivo plano, un archivo Excel, una base de datos, entre otros.

Capa 2: Tiene como objetivo la extracción de los datos, los cuales son analizados en la especificación de requerimientos, y estos formarán los datos del almacén de datos y deben ser los datos adecuados para el proceso de Minería de Datos.

Capa 3: En esta etapa se utilizan los algoritmos de Minería de Datos, que dependiendo del caso o escenario se puede aplicar el que más complementa la especificación de requerimientos de la Capa 1. Aquí debe estar definida la problemática y los indicadores de gestión a utilizar.

Capa 4: En esta etapa se definen los datos de predicción para cada algoritmo. Estos datos deben ser coherentes con la predicción de los datos e indicadores de gestión definidos en la Capa 3.

Capa 5: En esta capa se procede a realizar las consultas que solicitan los usuarios finales o que fueron especificadas en los requerimientos del problema, dichas consultas son resultados utilizando comparadores clásicos.

Capa 6: En esta capa se analiza cuál de los comparadores de posibilidad, de los presentados en la Tabla 1, es el más adecuado para la consulta. Estos comparadores han sido implementados en código SQL y se encuentra, explicado en el apartado 3 y en [5].

Capa 7: Aquí se aplican los comparadores de necesidad o posibilidad seleccionados en la Capa 6. Las consultas ejecutadas, deben ser las mismas de la Capa 5 para ver con mayor grado de detalle el aporte de estos comparadores.

La comparación es la última capa, y tiene como objetivo analizar los resultados de las consultas obtenidos de la capa 5 y 7, para especificar por qué puede ser útil uno u otro comparados y cuál es la diferencia de sus resultados.

5. APLICACIÓN DE LA ARQUITECTURA PROPUESTA (CAPAS 1 al 4)

Para validar nuestra propuesta, hemos utilizado un caso de estudio que nos permitió obtener los resultados de la implementación de los comparadores difusos sobre datos obtenidos por minería de datos. A continuación se describe la aplicación de cada una de las capas de la Figura 1.

Capa 1 (Datos): Aquí se utilizó la base de datos Adventure Works Cycles, la cual es un complemento de SQL Server 2008.

Capa 2 (Definición del problema): El modelo de datos con el cual se trabajó, es una parte del Data Warehouse de Adventure Works Cycles utilizando un submodelo y es la entrada para las distintas implementaciones de los procesos de Minería de Datos.

Capa 3 (Escenarios y algoritmos de Minería de Datos): El escenario que se utilizó en este caso, es el de *Correo Directo*, aquí se implementaron tres algoritmos: árboles de decisión, clústeres y Bayes Naive. En el caso de SQL Server da la posibilidad de sugerir cuales serían los mejores algoritmos según sea el caso.

Para el *Escenario de Correo Directo* se implementaron los algoritmos mencionados anteriormente, aunque los algoritmos son completamente diferentes, en este escenario se requiere decidir, *Qué tan probable es que una persona con ciertas características compre algún producto ofrecido*. Por lo cual se debe analizar cuál de los tres algoritmos realiza esta labor de mejor manera. Para esto existe una herramienta llamada “Gráfico de elevación”, la cual se puede encontrar en la pestaña “Gráfico de precisión de Minería de Datos”. Más información de este punto se encuentra en [5].

Capa 4 (Predicción de los escenarios de minería de datos): Para el *Escenario de Correo Directo*, el indicador definido es; *-Cuál es la probabilidad de que un posible cliente compre una bicicleta-*.

Para este caso, se utilizó una herramienta que puede seleccionar los algoritmos que se quieren comparar o seleccionar, algún conjunto de datos específico. La Figura 3, muestra el gráfico de elevación, en el podemos observar que el algoritmo que más se ajusta a nuestro modelo ideal (predicción perfecta) es el algoritmo de *Árboles de Decisión*, por lo cual será este el algoritmo elegido para realizar las predicciones y la posterior consultas clásicas y difusas. Cabe señalar que la extensión a realizar también es válida para los otros dos modelos (Clústeres y Bayes Naive), ya que como se mencionó anteriormente, aunque los algoritmos son completamente diferentes, en esta ocasión se utilizan con el mismo propósito (predecir un atributo).

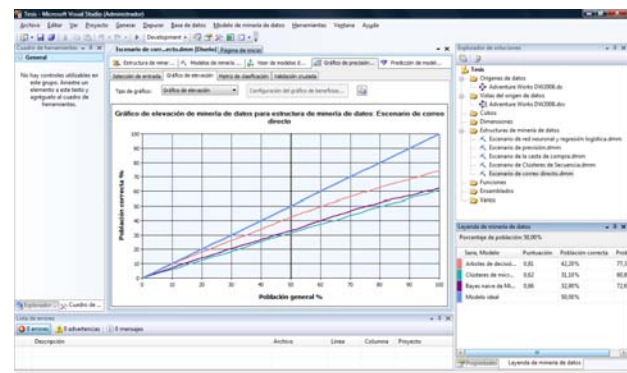


Figura 3: Gráfico de elevación de precisión de algoritmos de Minería de Datos.

Cuando se utiliza el algoritmo de árboles de decisión para realizar predicciones, este genera una consulta de predicción sobre una tabla de casos, dicha consulta entrega la probabilidad de que *-cada persona de la tabla de casos compre un producto-*, dicha tabla de casos contiene perfiles de probables clientes, además los resultados de predicción se almacenan en otra tabla, como resultado se obtiene un porcentaje, que se encuentra en el atributo *Expression* de la tabla en la cual se almacenaron los resultados, el cual dirá *que tan probable es que un potencial cliente se convierta en un cliente expresado de otra forma*, es la probabilidad de que cada posible comprador adquiera un producto (en este caso una bicicleta).

Otros atributos de esta tabla de resultados son: atributo *ProspectAlternateKey* que es identificador del posible cliente, el atributo *Bike Buyer* puede tener dos valores 0 (indica que no es comprador de bicicletas) o 1 (indica que el posible cliente es comprador de bicicletas) y el atributo *Expression* que almacena la probabilidad de que el posible

cliente adquiera un producto (INDICADOR), siendo este campo el que predijo el algoritmo.

6. ANÁLISIS DE RESULTADOS: COMPARACIÓN CONSULTA CLÁSICA Y CONSULTA DIFUSA (CAPAS 5 al 7)

Este apartado tiene por objetivo presentar los resultados obtenidos al utilizar los comparadores clásicos (Capa 5), los comparadores difusos (Capa 7) implementados en la capa 6, y las consultas ejecutadas sobre los resultados de predicción (Capa 4). Cada capa fue efectuada para el caso de *Escenario de Correo Directo*. El análisis de resultados fue obtenido de las predicciones realizadas por el algoritmo de *Árboles de Decisión* sobre una tabla de casos *tabla ProspectiveBuyer*. En la realización de estos pasos se recurrió a SQL Server Management Studio.

Para el análisis de los resultados se utilizó el valor *crisp* 0.51 con el difuminador ± 0.1 ($0.51 - 0.1$, $0.51 + 0.1$), de donde se obtiene el rango [0.41 a 0.61], es decir de un 41% a un 61% de probabilidad, sin incluir el extremo derecho (0.61). El valor *crisp* escogido corresponde a *-los posibles clientes que tienen menor probabilidad de adquirir un producto-*. Este valor *crisp* escogido se debe a que se quiere enfocar una campaña de motivación a los clientes que tienen menos probabilidad de comprar un producto, cabe señalar que la probabilidad del cliente con menos posibilidades de comprar es de 0.506963566619447, es decir un 51% aproximadamente.

A continuación mostramos el resultado de la consulta con el comparador difuso FLEQ y la consulta clásica con comparador menor o igual con todos los pasos. El resto de los comparadores implementados se encuentran en [5].

a) Comparación usando Expression FLEQ 0.51 y Expression ≤ 0.51 .

Para realizar esta comparación se usaron dos consultas, una consulta utilizando el comparador " $\leq$ " y otra consulta usando la función FLEQ (posiblemente menor o igual).

El siguiente código en SQL muestra las sentencias utilizada para la consulta clásica que arroja 119 registros (la consulta corresponde a la Capa 5) y la Tabla 2 muestra el resultado parcial de la ejecución de la consulta clásica, con los atributos correspondientes de la tabla de resultados explicada en el apartado anterior.

```
select * from
[AdventureWorksDW2008].[dbo].[Resultado
CorreoDirecto]
where Expression <= 0.51
```

Tabla 2: Resultados de la consulta usando Expression ≤ 0.51 .

	ProspectAlternateKey	Bike Buyer	Expression
1	3003	0	0,506963566619447
2	54107006788	0	0,506963566619447
...	...	...	...
119	70093423128	0	0,506963566619447

Luego ejecutamos la consulta difusa (corresponde a la Capa 7), para este tipo de consulta se tuvo que utilizar la condición " > 0 " usada al final de la cláusula *where*, ya que la función FLEQ retorna un grado de pertenencia entre 0 y 1, es decir, retorna el grado de pertenencia que posee el valor *crisp* al valor 0.51 difuminado en ± 0.1 , por lo que se debe escoger solo aquellos valores mayores a 0, considérese que el valor 0 indica que no pertenece al conjunto. El resultado parcial de la consulta difusa se muestra en la Tabla 3, con los tres atributos que se obtienen por la tabla de resultado de casos en SQL Server, explicado en el apartado 5.

```
select * from
[AdventureWorksDW2008].[dbo].[Resultado
CorreoDirecto]
where (select
[AdventureWorksDW2008].[dbo].[FLEQ]
(Expression, 0.51, 0.1)) > 0
```

Tabla 3: Resultados de la consulta usando Expression FLEQ 0.51.

	ProspectAlternateKey	Bike Buyer	Expression
1	3003	0	0,506963566619447
2	54107006788	0	0,506963566619447
...	...	...	...
670	37111264500	1	0,550329734038698

El análisis de resultados usando, *Expression FLEQ 0.51* y *Expression ≤ 0.51* , considera que una consulta clásica entrego 119 tuplas de resultados y la consulta difusa entrego 670 tuplas. En la Figura 4 se muestra la gráfica que apoya los resultados obtenidos con la consulta clásica, de color plomo se puede apreciar todo el conjunto de resultados incluidos por el comparador clásico. En la Figura 5 se muestra la gráfica que apoya los resultados obtenidos por la consulta difusa, de color plomo se puede observar todo el conjunto de resultados incluidos por el comparador difuso, aquí se debe recordar que el extremo 0.61 no es incluido.

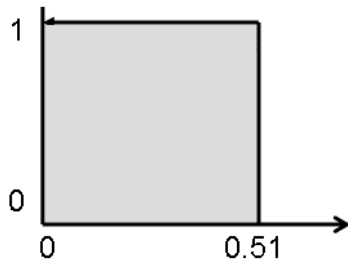


Figura 4: Gráfico de valores para el comparador clásico $\leq$.

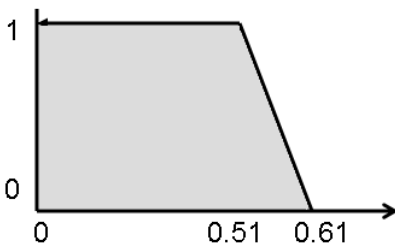


Figura 5: Gráfico de valores para el comparador difuso FLEQ.

El uso de los comparadores EQ y FEQ se encuentran en [7].

7. CONCLUSIONES

La lógica difusa junto con la Minería de Datos son herramientas muy poderosas a la hora de trabajar con indicadores de gestión para el apoyo a la toma de decisiones.

Los comparadores difusos entregan más tuplas, en algunos casos, para el análisis de datos obtenidos por un algoritmo de Minería de Datos, lo que puede apoyar mejor la toma de decisiones.

Los valores a difuminar son importantes de especificar y se deben hacer en conjunto con el usuario final, ya que el resultado de las consultas, deben estar asociadas al indicador de gestión.

Actualmente estamos trabajando en incorporar etiquetas lingüísticas a la consultas de lo que dará otro espectro de análisis a las consultas de algoritmos de Minería de Datos. También estamos trabajando en dar grados de importancia a los datos de cada cluster de Minería de Datos, o grados de pertenencia de los datos al cluster.

Trabajos Futuros: Se propone extender otro tipo de componentes de la lógica difusa a los datos o algoritmos de Minería de Datos. También analizar diferentes tipos de difuminación de las expresiones de las consultas.

Referencias

- [1]. Carrasco R., Araque F., Salguero A., Vila M. A..(2008). Applying Fuzzy Data mining to Tourism Area. Charper XXII. In Handbook of Research on Fuzzy Information Processing in Databases, Vol. II, pp. 563-589. Information Science Reference.
- [2]. Galindo, J. (Ed.), (2008). Section IV Fuzzy Data Mining. Handbook of Research on Fuzzy Information Processing in Databases. Hershey, PA, USA: Information Science Reference. <http://www.info-sci-ref.com>.
- [3]. Galindo, J., Urrutia, A., Piattini, M. (2006). Fuzzy Databases: Modeling, Design and Implementation. Idea Group Publishing Hershey, USA.
- [4]. Galindo, José (1999). "Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del modelo y adaptación de los SGBD Actuales", Tesis Doctoral, universidad de Granada, España, 1999.
- [5]. Mendez J. (2009). "Extensión de algoritmos de minería de datos con comparadores difusos en microsoft SQL Server 2008". Tesis de Licenciatura en Ciencias de la Ingeniería, Universidad Católica del Maule 2009.
- [6]. Urrutia A., Tineo L., & Gonzalez C. (2008). FSQL and SQLf: Towards a Standard in Fuzzy Databases. In Handbook of Research on Fuzzy Information Processing in Databases, Vol. I, pp. 270-298. Information Science Reference.
- [7]. Urrutia A, Mendez J., Gutierrez C. (2009). "Algoritmos de minería de datos extendidos con comparadores difusos y su impacto en los indicadores de gestión". Encuentro de Informática de Gestión Ediciones Universitarias de la Frontera. Universidad de la Frontera Chile, Diciembre de 2009. En edición.
- [8]. Zadeh L.A. (1965); Fuzzy Sets. Information and Control, 8, pp. 338-353.

IMPLEMENTACIÓN DE UNA BASE DE DATOS DIFUSA CON FIRST-2 Y PostgreSQL

Angélica Urrutia¹, José Galindo², Alejandro Sepúlveda¹

¹Universidad Católica del Maule, Chile, aurrutia@ucm.cl

²Universidad de Málaga, España, jgg@lcc.uma.es

Resumen

Se presenta en este trabajo una extensión de una base de datos relacional difusa utilizando como estructura lógica FIRST-2. Esta extensión es más potente y desarrollada sobre software libre, PostgreSQL. Además, se propone una aplicación a un sistema de citas médicas.

Palabras Clave: bases de datos difusas, implementación de datos con imprecisión, Fuzzy SQL, SQL, PostgreSQL.

1 INTRODUCCIÓN

La lógica difusa se ha utilizado para extender las bases de datos relacionales (BDR) a difusas (BDRD) [1][2][7]. Algunas propuestas están implementadas en Oracle, como lo es el servidor FSQ [3] y el motor SQLf [6]. En [3] se propone una extensión teórica de algunos tipos de datos difusos y proponiendo una estructura para su almacenamiento y uso conocida como FIRST-2, una extensión de FIRST [4], implementada bajo Oracle. En el mismo trabajo se presenta FSQ, una completa extensión del lenguaje SQL, para efectuar consultas difusas. FSQ incluye comparadores difusos, control de umbrales de cumplimiento, cuantificadores difusos, visualización del grado de cumplimiento de cada ítem, funciones difusas, diversos tipos de datos difusos (incluyendo fechas), etc. Este trabajo implementa la FIRST-2 y un Servidor FSQ en software libre usando un servidor PostgreSQL [5], el más potente SGBD de código abierto (www.postgresql.org).

En [2] se tiene un trabajo muy completo sobre las bases de datos difusas en general y que también incluye trabajos sobre FSQ.

2 RESUMEN DE FIRST-2

Antes de ver el modelo de representación de información difusa FIRST-2, vamos a hacer un breve resumen de los tipos de datos que incluye el modelo conceptual FuzzyEER, y que son los tratados por nuestro sistema. Se distinguen dos clases de atributos difusos: Atributos

cuyos valores son conjuntos difusos y atributos cuyos valores son grados difusos.

2.1. Atributos Difusos con Conjuntos Difusos

Estos atributos pueden clasificarse en cuatro tipos según [3], de acuerdo el tipo de referencial (dominio subyacente o eje X donde se definen los conjuntos difusos). En todos ellos se incluyen los valores Unknown, Undefined, y Null:

- **Tipo 1:** Son atributos precisos, sin imprecisión, *crisp*. Sin embargo, se permite definir etiquetas lingüísticas en su dominio y podremos usarlas en consultas difusas. Se almacenan igual que un atributo normal, pero puede ser transformado o manipulado usando condiciones difusas. Este tipo es útil para extender bases de datos tradicionales para permitir consultas difusas en sus dominios clásicos. Por ejemplo, preguntas del tipo: “Dame los empleados que ganan *mucho más que* el salario mínimo”.
- **Tipo 2:** Son atributos “imprecisos sobre un referencial ordenado”. Admiten valores *crisp* y difusos, en forma de distribuciones de posibilidad o conjuntos difusos, sobre un dominio subyacente ordenado. Es una extensión del Tipo 1 que sí permite el almacenamiento de información imprecisa, tal como el valor: “aproximadamente 2 metros”. Por simplicidad, estos conjuntos difusos suelen ser una función trapezoidal (Figura 1, donde el eje Y es el grado difuso).
- **Tipo 3:** Son atributos difusos con datos “discretos sobre dominio subyacente no ordenado con analogía”. Aquí se pueden definir etiquetas (Rubio, Moreno...) que son escalares con una relación de similitud o proximidad definida sobre esas etiquetas, para indicar en qué medida se parecen cada par de etiquetas. También se admiten distribuciones de posibilidad (o conjuntos difusos) sobre este dominio, como por ejemplo, el valor {1/Rubio, 0.4/Moreno}, que expresa que cierta persona es más Rubia que Morena. Observe que el dominio subyacente de ese valor difuso es el conjunto de etiquetas y este conjunto carece de orden.
- **Tipo 4:** Estos originales atributos son similares al Tipo 3, pero sin precisar la relación de similitud. En este caso suponemos que no necesitamos tal relación o que no existe.

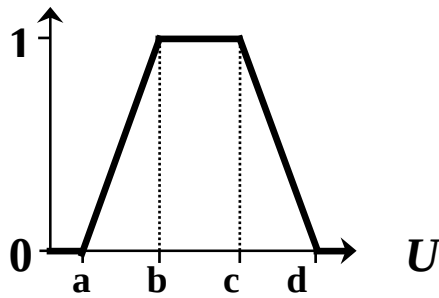


Figura 1: Función Trapezoidal.

2.2. Atributos con Grados Difusos

El dominio de estos grados puede estar en el intervalo $[0,1]$, aunque otros valores podrían también permitirse, tal como una distribución de posibilidad (usualmente sobre este intervalo unidad). Para simplificar, nosotros sólo usaremos grados en dicho intervalo, ya que otras opciones no ofrecen grandes ventajas.

El significado de estos grados es variado y depende de su uso. El procesamiento de los datos será diferente según este significado, por lo que el sistema debe almacenar y considerar este significado. Los significados más importantes de estos grados según distintos autores son [2][3][7]: grado de cumplimiento, grado de incertidumbre, grado de posibilidad y grado de importancia. Por supuesto, podremos definir y usar otros significados.

Estos grados difusos están clasificados en dos familias: grados asociados y no asociados. Los **grados asociados** están asociados a un valor específico al cual se incorpora imprecisión. Estos grados pueden ser asociados a diferentes conceptos [3]:

- **Grado en cada valor de un atributo (Tipo 5):** Algunos atributos pueden tener un grado difuso asociado a ellos. Esto implica que cada valor de este atributo (en cada tupla o instancia) tiene un grado asociado que mide el nivel de “imprecisión” de ese valor/atributo. Para interpretarlo, necesitamos saber el significado de ese grado y el significado del atributo asociado.
- **Grado en un conjunto de valores de diferentes atributos (Tipo 6):** Aquí, el grado está asociado a algunos atributos. Este es un caso poco usual, pero puede ser muy útil en casos específicos. Une la imprecisión de varios atributos en un único grado.
- **Grado en la instancia completa de la relación (Tipo 7):** Este grado está asociado a la tupla completa (u objeto) de la relación y no exclusivamente al valor de un atributo específico de la tupla. Normalmente, puede representar algún “grado de pertenencia” del objeto a la relación o tabla de la base de datos. También suelen usarse para medir el “grado de importancia” de cada objeto.

Por su parte, los **grados no asociados (Tipo 8)** son útiles cuando la información imprecisa que queremos representar, se puede representar usando únicamente el grado, sin asociar este grado a otro valor o valores. Por ejemplo, la peligrosidad de un medicamento puede ser expresado usando un grado difuso de este tipo.

Está fuera del ámbito de este trabajo demostrar la utilidad de estos atributos, grados y de sus distintos significados. Queda ello hecho por los distintos autores que han usado dichos casos de forma individual [1][2][3][6][7].

2.3 Representación de Atributos Difusos

Esta representación depende del tipo de atributo difuso. Los atributos Tipo 1 son representados normalmente, como atributos propios del SGBD, ya que no admiten valores difusos. Los atributos Tipo 2 necesitan cinco atributos clásicos: Uno almacena la clase de valor (Tabla 1) y los otros almacenan los valores *crisp* que configuran el valor difuso. Observe que en la Figura 1 y en la Tabla 1, la función trapezoidal necesita cuatro valores. Un valor aproximado (triangular, aproximadamente d , $d \pm \text{margen}$) se representa con una función triangular centrada en su núcleo d (grado 1) y con grado 0 en $d - \text{margen}$ y $d + \text{margen}$, donde el valor margen depende del contexto (Figura 1 con $b=c$ y $b-a=d-c=\text{margen}$). Para extender a otros tipos de valores distintos de la Tabla 1, podrían ampliarse el número de atributos de la representación.

Los atributos difusos Tipo 3 necesitan un número variable de valores (atributos clásicos): Uno almacena la clase de valor (Tabla 2). Observe que en la Tabla 2, la clase 3 necesita sólo dos valores, pero la número 4 necesita $2n$ valores, donde n es la longitud máxima para distribuciones de posibilidad para cada valor difuso. El valor n debe estar definido para cada atributo Tipo 3 (y se almacena en la FMB como veremos). Los atributos difusos Tipo 4 se representan igual que los Tipo 3. La diferencia se muestra en el siguiente apartado.

Tabla 1: Valores admitidos para atributos difusos Tipo 2.

Número	Tipo de valores
0, 1, 2	UNKNOWN, UNDEFINED, NULL
3	CRISP: d
4	LABEL: label_identifier
5	INTERVAL: $[n,m]$
6	APPROXIMATE VALUE: d
7	TRAPEZOIDAL: $[a,b,c,d]$

Tabla 2: Valores admitidos para atributos Tipo 3 y 4.

Número	Tipo de valores
0, 1, 2	UNKNOWN, UNDEFINED, NULL
3	SIMPLE: Degree/Label
4	POSSIBILITY DISTRIBUTION: Degree ₁ /label ₁ + ... + Degree _n /Label _n

Los grados difusos (Tipos 5, 6, 7 y 8) se representan usando un atributo numérico clásico, cuyo dominio incluya el intervalo $[0,1]$, usualmente con al menos dos decimales.

2.4 Información de la FMB

Los datos de metaconocimiento difuso son el conocimiento necesario sobre la base de datos difusa y especialmente sobre los atributos difusos. Esta información se almacena en formato relacional en la llamada FMB (Fuzzy Metaknowledge Base). Vamos primero a definir la información que se almacena en la FMB y posteriormente explicamos su estructura, sus tablas. Así pues, la FMB almacena esta información:

1. Atributos con capacidades difusas: Qué atributos son de los Tipos difusos 1 a 8.
2. El metaconocimiento requerido para cada atributo depende de su tipo:
 - Tipos 1 y 2: Estos atributos almacenan en la FMB la definición (conjunto difuso) de cada etiqueta lingüística, el “margen” para valores aproximados, y la mínima distancia para considerar dos valores como muy separados (valor “much”). Este último valor se usa en comparadores como “mucho mayor que”, comparador MGT de FSQL [3][6].
 - Tipos 3 y 4: Valor n (explicado antes), nombre de las etiquetas y, para el Tipo 3, la relación de similitud entre cada dos etiquetas.
 - Tipos 5 y 6: Significado del grado, y el atributo (Tipo 5) o atributos (Tipo 6) a los que el grado está asociado.
 - Tipos 7 y 8: Significado del grado.
3. Otros objetos: Éstos incluyen cualificadores difusos (asociados a un atributo y usados para establecer umbrales en consultas), y cuantificadores difusos (asociados a una tabla o a un atributo). Los cuantificadores son usados en consultas, tales como “Dame los empleados que pertenecen a la *mayoría* de los proyectos”, y también en restricciones difusas, como por ejemplo “Un empleado debe trabajar en *muchos* proyectos”.

Si dos atributos difusos de los Tipos 1, 2, 3 ó 4 necesitan las mismas definiciones, podemos registrar tales atributos como compatibles, para simplificar la FMB.

2.5 Tablas de la FMB

Por cuestiones de espacio no podemos explicar todos los atributos de todas las relaciones de la FMB. Sólo tratamos de dar una idea de la utilidad de cada tabla:

- o **FUZZY_COL_LIST:** Describe los atributos difusos de cada tabla. El valor F_TYPE se usa para el tipo de atributo difuso, de 1 a 8. LEN es el valor n .

$CODE\_SIG$ indica el significado del grado si $F\_TYPE \in [5,8]$.

- o **FUZZY_DEGREE_SIG:** Guarda los significados de los grados de la base de datos.
- o **FUZZY_OBJECT_LIST:** Esta tabla contiene las declaraciones de los objetos difusos relacionados con los atributos difusos, tales como, etiquetas lingüísticas, cualificadores y cuantificadores difusos. Los cuantificadores pueden ser absolutos o relativos, y pueden tener uno o dos argumentos [3].
- o **FUZZY_LABEL_DEF:** Define las etiquetas usando funciones trapezoidales (Figura 1).
- o **FUZZY_APPROX_MUCH:** Valores para el “margen” y el “much” para los Tipos 1 y 2.
- o **FUZZY_NEARNESS_DEF:** Relaciones de similitud para los Tipo 3.
- o **FUZZY_COMPATIBLE_COL:** Atributos difusos compatibles, con iguales etiquetas.
- o **FUZZY_QUALIFIERS_DEF:** Definición de cualificadores difusos.
- o **FUZZY_DEGREE_COLS:** Esta tabla establece los atributos (o columnas) asociados a grados difusos (sólo para los Tipos 5 y 6). Observa que un grado Tipo 5 tiene sólo un atributo asociado, mientras que un Tipo 6 puede tener múltiples. Además, un atributo puede tener muchos grados difusos asociados a él, pero todos de los Tipos 5 ó 6. Por supuesto, los Tipos 7 y 8 no usan esta tabla.
- o **FUZZY_ER_LIST:** En palabras del modelo FuzzyEER [3], esta tabla almacena las entidades difusas y las relaciones difusas. $DEGREE\_TYPE$ vale ‘M’ para entidades difusas, ‘C’ para entidades difusas con grados calculados automáticamente, ‘E’ e ‘I’ para entidades débiles difusas (dependencia de existencia o de identificación) y, finalmente, ‘R’ para relaciones difusas representadas por una tabla.
- o **FUZZY_TABLE_QUANTIFIERS:** Definición de cuantificadores asociados a una relación o tabla (no un atributo). Estos cuantificadores son usados en restricciones difusas [3] y pueden ser absolutos o relativos.

3 MODELO CLIENTE-SERVIDOR

Los módulos necesarios para la construcción de la extensión de una base de datos relacional a relacional difusa, usando la estructura lógica de la FIRST-2, según las especificaciones de [3] son, muy resumidamente:

- **Módulo Dinámico del Sistema:** Como parte del análisis de los requerimientos funcionales del sistema que son recogidos por casos de uso, se enmarca el modelo dinámico del sistema.
- **Módulo de Conexión:** Enlaza por defecto al sistema, luego realiza la llamada al servidor. Si el usuario solicita una conexión este módulo es quien llama al servidor, o al Módulo de Control de errores.

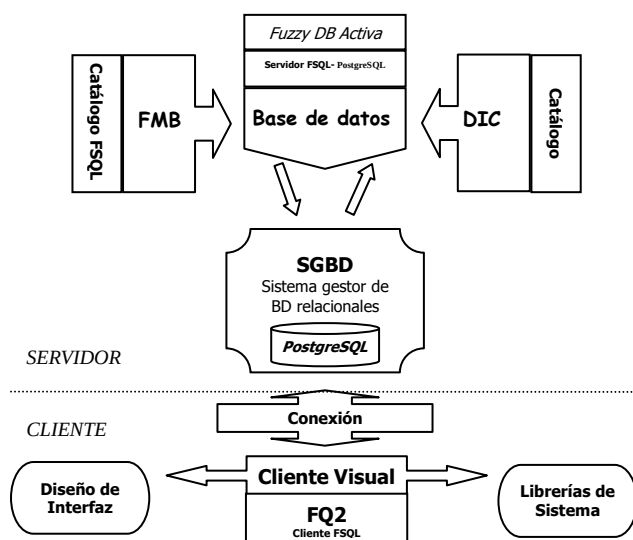


Figura 2: Esquema general de la arquitectura.

- **Módulo de Ejecución de Consulta Clásica:** El proceso de ejecución de una consulta clásica al igual que cualquier llamado al servidor,
- **Módulo de Ejecución de Consulta Difusa:** El proceso de ejecución de una consulta difusa es homólogo, pero con un análisis de la instrucción.
- **Módulo de Traducción:** Aquí el usuario ingresa la consulta difusa en FSQL [6], luego el servidor la analiza léxica, sintáctica y semánticamente. Si no hay errores se efectúa la traducción a SQL.

4 ARQUITECTURA

Para la implementación del cliente que hace de interfaz entre el usuario y el servidor PostgreSQL, se optó por un sistema desarrollado en plataforma Web. Se optó por *PHP*, principalmente por la compatibilidad con una gama de sistemas operativos incluyendo por supuesto Linux y Windows, pero también por la facilidad de instalación de los archivos fuentes, sus características son más conocidas por los usuarios, además de que este lenguaje es de software libre.

El servidor FSQL-PostgreSQL, está construido sobre una arquitectura cliente-servidor, con una interfaz de usuario Fuzzy Query (FQ GNU) que realiza peticiones al *Servidor*, así la carga de los procesos se divide en un par de separaciones lógicas, la de interfaz *Cliente*, pudiendo ser “n” usuarios cargando localmente su FQ, y los procesos del servidor que se acumulan en una sola máquina. En la Figura 2 se presenta un diagrama de la arquitectura, la cuál es explicada a continuación:

- **SGBD (Sistema Gestor de Base de Datos) relacional:** Es PostgreSQL, quien debe recibir todas las operaciones concebidas como extensión del SQL huésped. El SGBD realiza transacciones en SQL tradicional, y al momento de

efectuar una petición se traduce a la extensión del SQL, para que el servidor ejecute y despliegue la transacción con componentes difusas.

- **Base de Datos:** Almacena toda la información, ya sea *crisp* o difusa.
- **FMB y DIC:** Es la Base de Metaconocimiento Difuso ya explicada, y el Diccionario del sistema. Proporciona el diccionario o catálogo del sistema, en este caso de la FIRST-2.
- **Servidor FSQL-PostgreSQL:** Su función principal es capturar las sentencias escritas en lenguaje difuso FSQL [3][6], y traducir y enviar las mismas al SGBD. Utiliza todos los módulos que están soportados sobre PostgreSQL, es decir, base de datos, FMB, DIC y una serie de paquetes de funciones, procedimientos, *triggers*, etc., que se encuentran implementados en el lenguaje procedural PgSQL.

- **Cliente Visual (FQ2 GNU):** Se trata de una interfaz que comunica al usuario con el SGBD PostgreSQL. Aquí el usuario planteará sus operaciones o consultas FSQL y lanzará su ejecución al sistema.

- **Fuzzy DB activa:** Equivalen a una gama de *triggers* difusos que apoyan y activan la tarea del servidor FSQL.

- **Paquetes de uso del servidor FSQL:** Se trata de *scripts* de instalación y desinstalación, así como los paquetes de análisis y conversión de las sentencias en FSQL. También se tiene el paquete de operaciones difusas, que debe dotar al servidor FSQL de las operaciones difusas que generan la extensión.

- **Módulo de Conexión:** Se requieren tres pasos sencillos, primero invocar la librería (*adodb*), luego crear el objeto para finalmente ejecutar la conexión con los parámetros del SGBD.

- **Módulo de errores:** Para que la interfaz implementada sea amigable para el usuario se debe desplegar la información de los posibles errores de forma visual y clara, dicha interfaz de usuario se desarrolló en *Glade*.

- **Módulo de ejecución de consulta SQL:** El sistema deberá desplegar consultas en una pantalla tipo *query* estándar para luego desplegar su ejecución en una tabla.

- **Módulo de ejecución de Consultas** Al igual que para las consultas clásicas, el sistema deberá mostrar gráficamente en una tabla el resultado de una consulta difusa. También se incorporará la información del número de columnas y filas recuperadas. La Figura 3 muestra un ejemplo.

FILA	RUT	HABILIDAD	HABILIDAD1	EXPERIENCIA1	EXPERIENCIA1P1	EXPERIENCIA1P2	EXPERIENCIA2	EXPERIENCIA2P1
1	111	4	33	4	0.10	0	0.30	1
2	222	4	42	4	0.23	0	0.62	0
3	333	4	88	4	0.33	0	0.60	1

Figura 3: Tabla con resultados de una consulta FSQ en el programa cliente FQ2 con herramientas Web.

5 IMPLEMENTACIÓN

Para la implementación de las componentes de FIRST-2 se han seguido las siguientes tres primeras fases, para cada componente:

- **Especificación del componente (Etapa 1):** En esta etapa se especifican detalles y en el caso de que ya exista una versión anterior, se incluirá el algoritmo de ese componente, ya sea módulo, formulario, procedimiento, función o en general algún objeto de PostgreSQL.

- **Algoritmo de implementación (Etapa 2):** En esta etapa se especifica el algoritmo a implementar de la FIRST-2, si ya existe una versión anterior, se incluirá el algoritmo y los cambios que sufrió la versión nueva.

- **Implementación (Etapa 3):** Tecnología Web utilizando herramientas de desarrollo como *PHP*, *javascript*, *Ajax* y plantillas de estilo *CSS*. La gran ventaja de desarrollar el software con estas herramientas es que su licencia es *GNU*, además se puede usar tanto en Linux como en Windows.

- **Pruebas y correcciones parciales (Etapa 4):** Esto incluye la implementación de un completo caso de estudio que estudiamos a continuación.

6 EJEMPLO REAL: CITAS MÉDICAS

Presentamos un caso de estudio con componentes difusas que requieren la implementación de la FIRST-2, con distintos y variados tipos de atributos difusos, entidades difusas, interrelaciones difusas, y distintas restricciones difusas (participación difusa, completitud difusa...), agregación difusa, especialización definida para un atributo difuso, así como también añadir un grado (con

distinto significado posible) a uno o varios atributos, o a la entidad completa.

Este caso de estudio ha sido modelado utilizando el FuzzyEER [3], una extensión del modelo conceptual EER que permite el modelado de atributos difusos.

El objeto del sistema es la gestión de citas médicas de un consultorio de salud. La Figura 4 muestra el modelo del caso de estudio con los principales grados difusos. Las principales entidades se explican a continuación:

- **Entidad Paciente:** Información de las personas que se atienden en el consultorio. Posee datos no difusos, y también un atributo difuso Tipo 2 *Edad* sobre el cual existen cuatro etiquetas lingüísticas predefinidas {*niño*, *joven*, *adulto*, *adulto mayor*}, otro atributo es *Color_piel*, el cual es de Tipo 4 (dominio discreto no ordenado y sin relación de similitud entre sus etiquetas).

- **Entidad Especialista:** Esta entidad guarda información de los profesionales que atienden a los pacientes. Sus atributos son, dos datos no difusos (*Rut* y *Nombre*), uno Tipo 2 (*Habilidad*), un difuso Tipo 3 (*Experiencia*) con etiquetas {*aprendiz*, *normal*, *experto*}, indicando un dominio subyacente discreto con una relación de similitud entre ellos (el dominio real no es discreto ya que este tipo de datos admite conjuntos difusos sobre las etiquetas definidas). El último atributo es un grado difuso Tipo 5 asociado a los atributos *Experiencia* y *Habilidad*. Este grado mediría el grado de *fiabilidad* de sendos atributos.

- **Entidad Asistencia:** Esta entidad almacena las asistencias a las atenciones médicas almacenadas en la base de datos, posee tres atributos, de los cuales *Porcentaje_asistencia* es difuso Tipo 2.

La implementación de los nuevos tipos de datos difusos, permitirán incluir datos y consultas con incertidumbre, ampliando el espectro de información a disposición del proceso de toma de decisiones. Más información de cómo se construyó el sistema, así como diversos tipos de consultas posibles, puede encontrarse en [5].

7 CONCLUSIONES

La definición FIRST-2 [3], permite almacenar información difusa de muy variado tipo usando un SGBD relacional. Su antecesor, FIRST [4], fue implementado bajo Oracle y, entre otras cosas, no permitía tantos tipos de datos difusos. Aquí presentamos FIRST-2 implementado en PostgreSQL.

En cuanto a la implementación en PostgreSQL y componentes de software libre, podemos decir que lo más complejo es la construcción de funciones en código SQL, pero se consiguen las enormes ventajas que ofrece el

software libre, sin renunciar a aspectos de calidad. Este trabajo ofrece una eficaz plataforma para extender, a bajo coste, el uso de BD difusas a aplicaciones reales o científicas. Ambos tipos de aplicaciones han sido estudiadas en [2] mostrando gran potencialidad.

Con respecto a los trabajos futuros cabe mencionar que la tendencia actual es desarrollar aplicaciones OO (Orientadas a Objeto). Sería posible implementar el Servidor FSQl con un Modelo Objeto-Relacional de Base de datos [1] y tecnología Java en el programa cliente, derivándole a él algunas tareas para sacar provecho de su buen desempeño en el uso de la memoria.

También resultará muy interesante aplicar e investigar estas herramientas para diversas aplicaciones de bases de datos difusas, especialmente Data Mining [2].

Agradecimientos

Trabajo parcialmente apoyado por el Gobierno de España (proyectos TIN2006-14285 y TIN2006-07262) y el Gobierno de Andalucía (proyecto P06-TIC-01570).

Referencias

[1] Barranco, C.D., Campaña, J.R., & Medina, J.M. (2008). Towards a Fuzzy Object-Relational Database Model. In Handbook of Research on Fuzzy Information Processing in Databases, Vol. II, pp. 435-461. Information Science Reference (<http://www.info-sci-ref.com>).

[2] Galindo, J. (Ed.), (2008). Handbook of Research on Fuzzy Information Processing in Databases. Hershey, PA, USA: Information Science Reference (<http://www.info-sci-ref.com>).

[3] Galindo, J., Urrutia, A., Piattini, M. (2006). Fuzzy Databases: Modeling, Design and Implementation. Idea Group Publishing Hershey, USA.

[4] Medina J.M., Pons O., Vila A. (1995). FIRST. A Fuzzy Interface for Relational SysTems. VI International Fuzzy Systems Association World Congress (IFSA 1995). Sao Paulo, Brasil.

[5] Sepúlveda, A. (2009). Tratamiento de la imprecisión en BD difusas extensión de la FIRST 2 bajo software Libre. Seminario de Título, Universidad Católica del Maule, Chile.

[6] Urrutia, A., Tineo, L., & Gonzalez, C. (2008). FSQl and SQLf: Towards a Standard in Fuzzy Databases. In Handbook of Research on Fuzzy Information Processing in Databases, Vol. I, pp. 270-298. Information Science Reference (<http://www.info-sci-ref.com>).

[7] Zadrozny, S., de Tré, G., de Caluwe, R., & Kacprzyk, J. (2008). An Overview of Fuzzy Approaches to Flexible Database Querying. In Handbook of Research on Fuzzy Information Processing in Databases, Vol. I, pp. 34-54. Hershey, PA, USA: Information Science Reference (<http://www.info-sci-ref.com>).

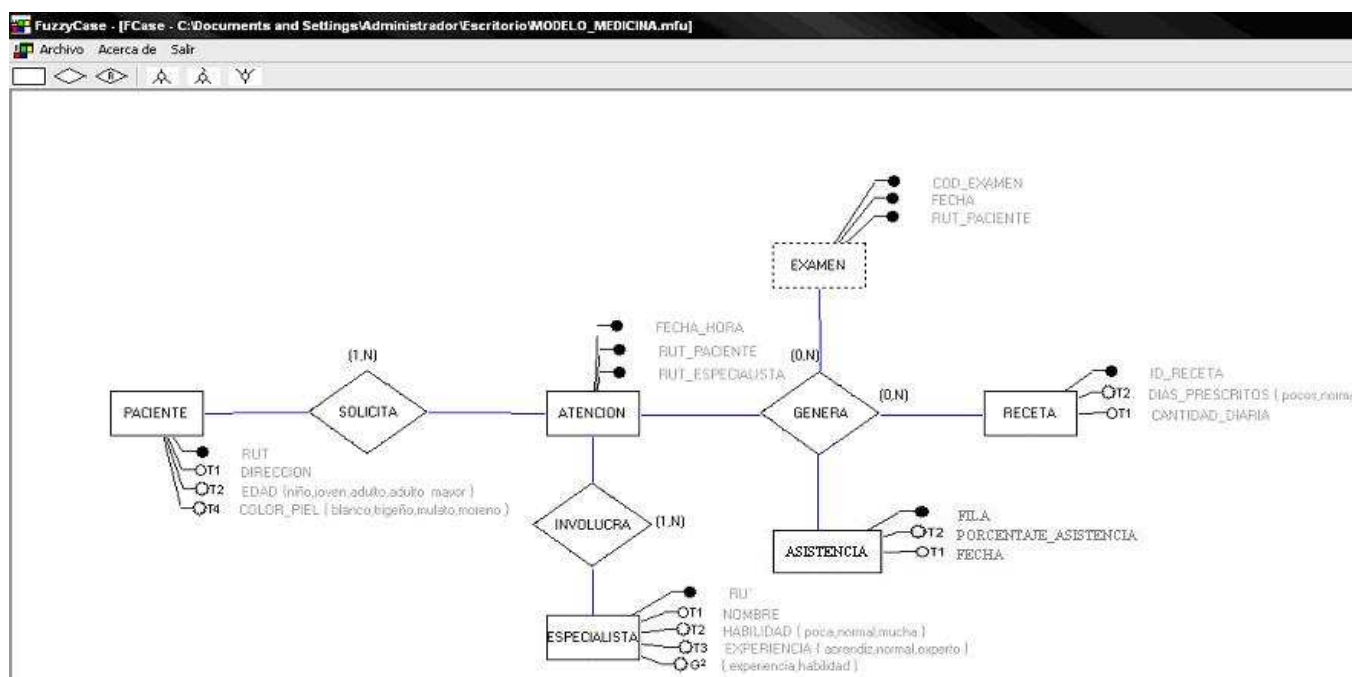


Figura 4: Modelo FuzzyEER del caso de estudio aplicado, usando la herramienta FuzzyCase.

A complete logic for fuzzy functional dependencies over t-norms

P. Cordero¹ M. Enciso¹ A. Mora¹ I. P. de Guzmán¹

¹ Universidad de Málaga, Spain, {pcordero,enciso}@uma.es, {amora,guzman}@ctima.uma.es

Abstract

A fuzzification of the concept of functional dependency based on t-norms is presented. T-norms are equally used in the definition of the similarity relations in the domains and in the fuzziness degree assigned to functional dependencies. A sound and complete logic for this fuzzy functional dependencies is also introduced and, finally, its applicability to redundancy removing is shown.

Keywords: Fuzzy Logic, Fuzzy Functional Dependency, T-norms.

1 Introduction

The developments of Fuzzy Logic during the last three decades have allowed the generalization of the relational model in order to consider imprecise knowledge. L. Zadeh affirms in [25] that the relational facet -FLr- of the Fuzzy Logic is focussed on fuzzy relations and fuzzy dependencies. In FLr facet, L. Zadeh refers to some works devoted to the notion of Fuzzy Attribute Implication [5]. This notion corresponds with the idea of Fuzzy Functional Dependency (FFD)

Three approaches have been used in the literature [22] to consider fuzzy databases: the use of fuzzy membership values [1, 14], the introduction of possibility distributions [18–20] or similarity relations [7–9].

The last approach has been fully tackled. Functional Dependencies have been studied over Domains with Similarity Relations and some definitions of Fuzzy Functional Dependency (FFD) have been introduced [3, 22–24]. In these works, some complete axiomatic systems defined over FFDs with similarity relations are defined too. They are extensions of classical

FD logics and lack of an executable perspective. They were designed to be a formal framework to show the semantics of the functional dependency (either classical or fuzzy). As a matter of fact, they have not been used to reason about functional dependencies and it is not a trivial matter to design an automatic method directly based on their axiomatic system. The main obstacle is that all of them are strongly based on the transitivity axiom, which limits the efficiency of the automatic deduction methods for these logics.

In [13] we have introduced for the first time a sound and complete axiomatic system for Fuzzy Functional Dependencies where the core is a Fuzzy Simplification Rule instead of the usual Fuzzy Transitivity Rule. The importance of our logic comes from the development of executable algorithms to reason automatically and solve the implication problem with FFDs [12], and how the simplification rule for FFDs may be used to remove redundancy from a set of FFDs [13].

A similar approach is tackled by for Belohlavek et.al. in [2, 4]. They show a system of Armstrong-like derivation rules and describe a non-redundant basis of all rules which are true in a data table and present a method for computation of nonredundant bases of attribute implications from data tables with fuzzy attributes. Dependency of attributes and determining redundant attributes in decision tables are some of the important issues in the application of Knowledge Discovery and Data Mining. In [15] the authors introduce the fuzzy functional dependency that satisfies Armstrong Axioms. In addition, they also discuss some interesting applications such as approximate data reduction.

Some applications require the use of new fuzzy operators. They are the triangular-norms (t-norms) used in the context of information retrieval with fuzzy ontologies. The t-norm concept is central in fuzzy logic and its study is very important from the theoretical point of view to the applications [11].

In this work, we present a fuzzification of the concept of functional dependency based on t-norms and we introduce a sound and complete logic for this kind of fuzzy functional dependencies. Our approach (Section 3) is given by considering similarity relations in the domains. These similarity relations are fuzzy equivalence relations in its broadest sense (based on t-norms). Moreover, the concept of FFD introduced is actually fuzzy, that is, they assigns truth values in the $[0, 1]$ interval and generalizes other approaches given in the literature [13, 22] by replacing the minimum operator by arbitrary t-norms. In Section 4, we introduce our logic and prove its soundness and completeness. Finally, in Section 4.3, we show that our logic is executable and its principal inference rule can be seen as a simplification tool because they directly remove redundancy.

2 Preliminaries

We present here the basic concepts about functional dependencies as they appear in database literature. Let $\{D_a \mid a \in \mathcal{A}\}$ be a family of sets indexed in a finite non-empty set of indexes $\mathcal{A}$. We call *attributes* to the indexes and *domain of the attribute a* to the set D_a . We work over the product of these domains, $\mathbb{D} = \prod_{a \in \mathcal{A}} D_a$. The elements in this product $t = (t_a)_{a \in \mathcal{A}} \in \mathbb{D}$ will be named tuples. A relation is a set of tuples $R \subseteq \mathbb{D}$, usually represented as a table.

We introduce here the notation widely accepted in the database community. Given $X, Y \subseteq \mathcal{A}$, XY denotes $X \cup Y$. Given $X \subseteq \mathcal{A}$, D_X denotes $\prod_{a \in X} D_a$. Let $t \in R$ be a tuple, then $t_{/X}$ denotes the *projection* of t to D_X ; that is, if $t = (t_a)_{a \in \mathcal{A}}$ then $t_{/X} = (t_a)_{a \in X}$.

Definition 2.1 Any statement of the type $X \rightarrow Y$, where $X, Y \subseteq \mathcal{A}$, is named a **functional dependency**. We say that a relation $R \in \mathbb{D}$ satisfies $X \rightarrow Y$ if, for all $t_1, t_2 \in R$ we have that: $t_{1/X} = t_{2/X}$ implies that $t_{1/Y} = t_{2/Y}$.

Remark 2.1 The term functional comes from the fact that: the relation R satisfies the FD $X \rightarrow Y$ if R restricted to XY is a (partial) function from D_X to D_Y .

The aim of this paper is to fuzzify this concept and to give a sound and complete fuzzy logic that allows us to manipulate this kind of functional dependencies. Our approach uses the unit interval $[0, 1]$ for the system of truth values, the infimum (denoted by $\wedge$) as the universal quantifier, the supremum (denoted by $\vee$) as the existential quantifier and an arbitrary t-norm (triangular norm denoted by $\otimes$) as the conjunction. That is,

the system of truth values is $([0, 1], \vee, \wedge, 0, 1, \otimes)$ where $([0, 1], \otimes, 1)$ is a commutative monoid and $\otimes$ is monotonic ($a \leq b$ implies $a \otimes c \leq b \otimes c$, for all $a, b, c \in [0, 1]$)

3 Fuzzy Functional Dependencies

The most usual way to fuzzify the concept of functional dependency is by replacing the equality in the definition by fuzzy relations named similarity relations. In this case, we will consider that each domain D_a is endowed with a similarity relation $\rho_a : D_a \times D_a \rightarrow [0, 1]$, that is, a reflexive, symmetric and $\otimes$ -transitive fuzzy relation. We can extend these relations to D_X for all $X \subseteq \mathcal{A}$ as follow:

$$\rho_X(t_1, t_2) = \bigwedge_{a \in X} \rho_a(t_1, t_2)^1$$

Remark 3.1 The definitions of fuzzy functional dependency in the literature [3, 12, 21, 23, 24] are very similar, having slight differences among them. They fuzzify the equality between the attributes value in the following way: A relation $R \subseteq \mathbb{D}$ satisfies the FFD $X \rightarrow Y$ if $\rho_Y(t_1, t_2) \geq \rho_X(t_1, t_2)$ holds, for all $t_1, t_2 \in R$.

Although the definition introduces the fuzzy relations of similarity and generalizes the classical definition, we can say that the functional dependency remains crisp. The inclusion of a degree of fuzzyness in the dependency itself is done in [22]. In this work we adopt the following definition of fuzzy functional dependency.

Definition 3.1 A **fuzzy functional dependency** is an expression $X \xrightarrow{\theta} Y$ where $X, Y \subseteq \mathcal{A}$ and $\theta \in [0, 1]$. A relation $R \subseteq \mathbb{D}$ is said to satisfy $X \xrightarrow{\theta} Y$ if, for all $t_1, t_2 \in R$, the inequation $\theta \otimes \rho_X(t_1, t_2) \leq \rho_Y(t_1, t_2)$ holds.

We remark that, if $\theta = 1$, the previous definition of FFD matches up with the definition of FFD proposed in Remark 3.1. Moreover, if the similarity relations are strongly reflexive then it define a crisp classical functional dependency.

Proposition 3.2 Let $R \subseteq \mathbb{D}$, $X, Y, Y' \subseteq \mathcal{A}$ and $\theta_1, \theta_2 \in [0, 1]$.

1. If $Y \subseteq X$ then R satisfies $X \xrightarrow{1} Y$.
2. If R satisfies $X \xrightarrow{\theta_1} Y$ and $\theta_1 \geq \theta_2$ then R satisfies $X \xrightarrow{\theta_2} Y$.
3. If R satisfies $X \xrightarrow{\theta_1} Y$ and $Y' \subseteq Y$ then R satisfies $X \xrightarrow{\theta_1} Y'$.

¹To simplify the notation, when no confusion arise, we write $\rho_X(t_1, t_2)$ instead of $\rho_X(t_{1/X}, t_{2/X})$.

PROOF: Straightforward from definition. $\square$

Proposition 3.3 (Composition) *Let $R \subseteq \mathbb{D}$, $X, Y, U, V \subseteq \mathcal{A}$ and $\theta_1, \theta_2 \in [0, 1]$. If R satisfies $X \xrightarrow{\theta_1} Y$ and $U \xrightarrow{\theta_2} V$ then R also satisfies $XU \xrightarrow{\theta_1 \wedge \theta_2} YV$.*

PROOF: If R satisfies $X \xrightarrow{\theta_1} Y$ and $U \xrightarrow{\theta_2} V$ then, for all $t_1, t_2 \in R$, $\rho_Y(t_1, t_2) \geq \theta_1 \otimes \rho_X(t_1, t_2)$ and $\rho_V(t_1, t_2) \geq \theta_2 \otimes \rho_U(t_1, t_2)$. Therefore

$$\begin{aligned} \rho_{YV}(t_1, t_2) &= \rho_Y(t_1, t_2) \wedge \rho_V(t_1, t_2) \\ &\geq (\theta_1 \otimes \rho_X(t_1, t_2)) \wedge (\theta_2 \otimes \rho_U(t_1, t_2)) \\ &\stackrel{(1)}{\geq} (\theta_1 \wedge \theta_2) \otimes (\rho_X(t_1, t_2) \wedge \rho_U(t_1, t_2)) \\ &= (\theta_1 \wedge \theta_2) \otimes \rho_{XU}(t_1, t_2) \end{aligned}$$

where in (1) we have used that, for isotonicity of $\otimes$,

$$\begin{aligned} \theta_1 \otimes \rho_X(t_1, t_2) &\geq (\theta_1 \wedge \theta_2) \otimes (\rho_X(t_1, t_2) \wedge \rho_U(t_1, t_2)) \\ \theta_2 \otimes \rho_U(t_1, t_2) &\geq (\theta_1 \wedge \theta_2) \otimes (\rho_X(t_1, t_2) \wedge \rho_U(t_1, t_2)) \end{aligned}$$

$\square$

Proposition 3.4 (Simplification) *Let $R \subseteq \mathbb{D}$, $X, Y, U, V \subseteq \mathcal{A}$ and $\theta_1, \theta_2 \in [0, 1]$. If $X \subseteq U$, $X \cap Y = \emptyset$ and R satisfies $X \xrightarrow{\theta_1} Y$ and $U \xrightarrow{\theta_2} V$ then R also satisfies $U - Y \xrightarrow{\theta_1 \otimes \theta_2} V - Y$.*

PROOF: The set U can be written as disjoint union as $U = X \cup U_Y \cup U_r$ where $U_Y = U \cap Y$ and $U_r = U \setminus (X \cup Y)$.

From hypothesis, for all $t_1, t_2 \in R$,

$$\rho_Y(t_1, t_2) \geq \theta_1 \otimes \rho_X(t_1, t_2) \quad (1)$$

$$\rho_V(t_1, t_2) \geq \theta_2 \otimes \rho_U(t_1, t_2) \quad (2)$$

and, by replacing (1) in (2), we have that

$$\begin{aligned} \rho_{V-Y}(t_1, t_2) &\stackrel{(i)}{\geq} \rho_V(t_1, t_2) \stackrel{(2)}{\geq} \theta_2 \otimes \rho_U(t_1, t_2) \\ &= \theta_2 \otimes (\rho_X(t_1, t_2) \wedge \rho_{U_Y}(t_1, t_2) \wedge \rho_{U_r}(t_1, t_2)) \\ &\stackrel{(ii)}{\geq} \theta_2 \otimes (\rho_X(t_1, t_2) \wedge \rho_Y(t_1, t_2) \wedge \rho_{U_r}(t_1, t_2)) \\ &\stackrel{(1)}{\geq} \theta_2 \otimes (\rho_X(t_1, t_2) \wedge (\theta_1 \otimes \rho_X(t_1, t_2)) \wedge \rho_{U_r}(t_1, t_2)) \end{aligned}$$

where (i) is due to $V \setminus Y \subseteq V$ and (ii) is due to $U_Y = U \cap Y \subseteq Y$ and the isotonicity of $\otimes$ and $\wedge$. Now, since $\rho_X(t_1, t_2) \geq \theta_1 \otimes \rho_X(t_1, t_2)$,

$$\rho_{V-Y}(t_1, t_2) \geq \theta_2 \otimes ((\theta_1 \otimes \rho_X(t_1, t_2)) \wedge \rho_{U_r}(t_1, t_2)) \quad (3)$$

Since $\otimes$ is monotonic, $\theta_1 \otimes \rho_X(t_1, t_2) \geq \theta_1 \otimes (\rho_X(t_1, t_2) \wedge \rho_{U_r}(t_1, t_2))$ and $\rho_{U_r}(t_1, t_2) = 1 \otimes \rho_{U_r}(t_1, t_2) \geq \theta_1 \otimes (\rho_X(t_1, t_2) \wedge \rho_{U_r}(t_1, t_2))$. Therefore,

$$\begin{aligned} &(\theta_1 \otimes \rho_X(t_1, t_2)) \wedge \rho_{U_r}(t_1, t_2) \\ &\geq \theta_1 \otimes (\rho_X(t_1, t_2) \wedge \rho_{U_r}(t_1, t_2)) \\ &= \theta_1 \otimes \rho_{U-Y}(t_1, t_2) \end{aligned}$$

Finally, by replacing in (3), and using commutativity and associativity of $\otimes$,

$$\rho_{V-Y}(t_1, t_2) \geq (\theta_1 \otimes \theta_2) \otimes \rho_{U-Y}(t_1, t_2)$$

$\square$

4 The Simplification Logic for FFDs

We are interested in an axiomatic system that allows us to syntactically derive FFDs. There exists in the literature some complete axiomatic system defined over FFDs with similarity relations [3, 23, 24]. However, there are not many axiomatic systems in the literature to reasoning with fuzzy functional dependencies where the dependency is fuzzy. One of them is given by Sozat and Yazici [22]. It is a fuzzy extension of Armstrong Axiom's and it has the problem inherent in transitivity.

4.1 $\mathbf{SL}_{FFD}$ Logic

We introduce $\mathbf{SL}_{FFD}$, a new logic more adequate for the applications, named Simplification Logic for fuzzy functional dependencies. Its language is the following:

Definition 4.1 *Given a numerable set of attribute symbols $\mathcal{A}$, we define the language $\mathbf{L} = \{X \xrightarrow{\theta} Y \mid \theta \in [0, 1] \text{ and } X, Y \in 2^{\mathcal{A}} \text{ with } X \neq \emptyset\}$.*

The semantic of this logic was outlined in previous sections. The semantic models are pairs made up of a family of domains with its similarities $\{(D_a, \rho_a) \mid a \in \mathcal{A}\}$ and a relation $R \subseteq \mathbb{D}$. However, to simplify the notation, we will only refer to the relation. So $R \models X \xrightarrow{\theta} Y$ denotes that R satisfies the functional dependency $X \xrightarrow{\theta} Y$, $R \models \Gamma$ denotes that R satisfies every fuzzy functional dependency in the set Γ and $\Gamma \models X \xrightarrow{\theta} Y$ denotes that, for all $R \subseteq \mathbb{D}$, $R \models \Gamma$ implies $R \models X \xrightarrow{\theta} Y$. In this point we present the axiomatic system:

Definition 4.2 *The axiomatic system $\mathcal{S}_{FFD}$ on $\mathbf{L}$ has one axiom scheme:²*

²In the literature, the set of attributes Y must be non-empty. In $\mathbf{SL}_{FFD}$, we consider the empty attribute, denoted $\top$. Note that $X \xrightarrow{1} \top$ is an axiom scheme.

Reflexive Axioms (Ax): for all $Y \subseteq X$

$$\vdash_{S_{FFD}} X \xrightarrow{1} Y$$

The inferences rules are the following:

Inclusion Rule (InR): if $\theta_1 \geq \theta_2$

$$X \xrightarrow{\theta_1} Y \vdash_{S_{FFD}} X \xrightarrow{\theta_2} Y$$

Decomposition Rule (DeR): if $Y' \subseteq Y$

$$X \xrightarrow{\theta} Y \vdash_{S_{FFD}} X \xrightarrow{\theta} Y'$$

Composition Rule (CoR):

$$X \xrightarrow{\theta_1} Y, U \xrightarrow{\theta_2} V \vdash_{S_{FFD}} XU \xrightarrow{\theta_1 \wedge \theta_2} YV$$

Simplification Rule (SiR): if $X \subseteq U$ and $X \cap Y = \emptyset$

$$X \xrightarrow{\theta_1} Y, U \xrightarrow{\theta_2} V \vdash_{S_{FFD}} U - Y \xrightarrow{\theta_1 \otimes \theta_2} V - Y$$

The deduction ($\vdash_{S_{FFD}}$) and equivalence ($\equiv_{S_{FFD}}$) concepts are introduced as usual.

4.2 Soundness and Completeness of SL_{FFD} Logic

To study the implication between the syntactic and the semantic level requires several previous definitions.

Definition 4.3 Let Γ be a set of fuzzy functional dependencies over $\mathcal{A}$. The closure of Γ is the set $\Gamma^+ = \{X \xrightarrow{\theta} Y \mid \Gamma \vdash_{S_{FFD}} X \xrightarrow{\theta} Y\}$.

Note that, as a consequence of **Ax** and **InR**, Γ^+ assigns an infinite set of pairs (Y, θ) to every non-empty set X . If the set Y is also fixed then Γ^+ gives an interval (consequence of **InR**) whose supremum will be denoted as $\theta_{X,Y}^+$

$$\theta_{X,Y}^+ = \sup\{\theta \in [0, 1] \mid X \xrightarrow{\theta} Y \in \Gamma^+\}$$

On the other hand, if we fix the value of θ then a subset of $2^{\mathcal{A}}$ is obtained. This set is finite and, by **DeR** and **CoR**, is an ideal of $(2^{\mathcal{A}}, \subseteq)$. The maximum element of this ideal will be denoted by X_{θ}^+ .

$$X_{\theta}^+ = \max\{Y \subseteq \mathcal{A} \mid X \xrightarrow{\theta} Y \in \Gamma^+\}$$

The following theorem ensures the soundness and completeness of the axiomatic system.

Theorem 4.4 Let Γ be a finite set of fuzzy functional dependencies over $\mathcal{A}$, $\emptyset \neq X, Y \subseteq \mathcal{A}$ and $\theta \in [0, 1]$. Then $\Gamma \vdash_{S_{FFD}} X \xrightarrow{\theta} Y$ if and only if $\Gamma \models X \xrightarrow{\theta} Y$.

PROOF: The soundness is a consequence of Propositions 3.2, 3.3 and 3.4. The completeness is proved showing that $\Gamma \not\vdash_{S_{FFD}} X \xrightarrow{\theta} Y$ implies that $\Gamma \not\models X \xrightarrow{\theta} Y$. That is, if $Y \not\subseteq X_{\theta}^+$ then there exists a model for Γ that it is not model for $X \xrightarrow{\theta} Y$.

Let $\Theta = \{\theta_{U,V}^+ \mid U, V \in 2^{\mathcal{A}}, U \neq \emptyset \text{ and } \theta_{U,V}^+ < \theta\}$ and let $\tau \in [0, 1]$ such that $\theta > \tau > \max \Theta$. Let $D_a = \{u, v\}$, $\rho_a(u, u) = \rho_a(v, v) = 1$ and $\rho_a(u, v) = \rho_a(v, u) = \tau$ for all $a \in \mathcal{A}$. We will prove that the relation $R = \{t_1, t_2\} \subseteq \prod_{a \in \mathcal{A}} D_a$ where $t_{1a} = u$, for all $a \in \mathcal{A}$, $t_{2a} = u$ for all $a \in X_{\theta}^+$, and $t_{2a} = v$, for all $a \notin X_{\theta}^+$, is a model for Γ but it is not model for $X \xrightarrow{\theta} Y$.

Let $U \xrightarrow{\theta_1} V \in \Gamma$. If $\theta_{U,V}^+ < \theta$ then

$$\theta_1 \otimes \rho_U(t_1, t_2) \leq \theta_1 \leq \theta_{U,V}^+ < \tau = \rho_V(t_1, t_2)$$

If $\theta_{U,V}^+ \geq \theta$ and $U \not\subseteq X_{\theta}^+$ then $\rho_U(t_1, t_2) = \tau$ and

$$\theta_1 \otimes \rho_U(t_1, t_2) \leq \tau = \rho_V(t_1, t_2)$$

If $\theta_{U,V}^+ \geq \theta$ and $U \subseteq X_{\theta}^+$ then $V \subseteq X_{\theta}^+$ (it is a consequence of the composition rule) and

$$\theta_1 \otimes \rho_U(t_1, t_2) = \theta_1 \otimes 1 = \theta_1 \leq \rho_V(t_1, t_2) = 1$$

Therefore, R is a model for Γ but it is not a model for $X \xrightarrow{\theta} Y$ because

$$\theta \otimes \rho_X(t_1, t_2) = \theta \otimes 1 = \theta > \rho_Y(t_1, t_2) = \tau$$

□

4.3 Redundancy elimination via Simplification

In a database system we look for designs with no redundancy and functional dependencies were defined to capture some semantics of the data strongly connected with the occurrence of redundancy in a database. In the same sense, redundancy is not desirable in the integrity constrain of a database.

The logic that we have introduced is more adequate for the applications. In the following, we illustrate this assertion by showing its good behavior for removing redundancy. The primitive rules allow us to directly eliminate redundancy without use other tools with upper cost. It is possible because our inference rules are really equivalences if redundancy exists, as the following theorem shows. The systematic application of the following equivalences removes redundancy.

Theorem 4.5 Let $X \xrightarrow{\theta_1} Y, U \xrightarrow{\theta_2} V \in \mathbf{L}$.

Decomposition equivalence: If $X \cap Y \neq \emptyset$, then $\{X \xrightarrow{\theta_1} Y\} \equiv_{S_{FFD}} \{X \xrightarrow{\theta_1} Y - X\}$.

Simplification equivalence: If $X \cap Y = \emptyset$, $X \subseteq U$ and $\theta_1 \geq \theta_2$, then

$$\{X \xrightarrow{\theta_1} Y, U \xrightarrow{\theta_2} V\} \equiv_{\mathcal{S}_{FFD}} \{X \xrightarrow{\theta_1} Y, U - Y \xrightarrow{\theta_2} V - Y\}$$

PROOF: The decomposition equivalence is straightforward. In the second equivalence, the left-right implication is due to **SiR** and the proof of the converse implication is the following:

1. $U \xrightarrow{1} \top$	Ax
2. $X \xrightarrow{\theta_1} Y$	Hypothesis
3. $U \xrightarrow{\theta_1} Y$	1, 2, CoR
4. $U - Y \xrightarrow{\theta_2} V - Y$	Hypothesis
5. $U \xrightarrow{\theta_2} VY$	3, 4, CoR
6. $U \xrightarrow{\theta_2} V$	5, DeR

□

5 Conclusions and future works

We have presented the most generalized version of fuzzy functional dependency, considering domain of data with similarity relations in its broadest sense (based on t-norms) and including a threshold for the fuzziness of the dependency itself. We have also introduced logic for this kind of fuzzy functional dependencies proving its soundness and completeness and we show that our logic is executable. Its principal inference rule can be seen as a simplification tool because they directly remove redundancy.

As future works in the use of $\mathbf{SL}_{FFD}$ logic, we are developing an algorithm to automatically remove the redundancy of FFDs. We will also study an extension of the closure algorithm and we will face on with the implication problem using the rules of the $\mathbf{SL}_{FFD}$ logic.

Acknowledgements

This work has been partially supported by projects TIN2007-65819 (Science Ministry of Spain) and P06-FQM-02049 (Junta de Andalucía).

References

- [1] J.F. Baldwin. Knowledge engineering using a fuzzy relational inference language. *Proc. IFAC Conf. on Fuzzy Information, Knowledge Representation, and Decision Processes, Marseille, France*, 15–23, 1983.
- [2] R. Belohlavek, V. Vychodil. Computing non-redundant bases of if-then rules from data tables with graded attributes. *Australian Conference on Artificial Intelligence* 1126–1129 (2005)
- [3] R. Belohlavek, V. Vychodil. Tables with Similarity Relations: Functional Dependencies, Complete Rules and Non-redundant Bases in Database Systems for Advanced Applications. *LNCS*, 3882: 644–658, 2006.
- [4] R. Belohlavek, V. Vychodil. Data Tables with Similarity Relations: Functional Dependencies, Complete Rules and Non-redundant Bases. *DAS-FAA* 644–658 (2006)
- [5] R. Belohlavek, V. Vychodil. Basic Algorithm for Attribute Implications and Functional Dependencies in Graded Setting. *Int. J. Found. Comput. Sci.* 19(2): 297–317 (2008)
- [6] F. Bobillo, U. Straccia. Fuzzy description logics with general t-norms and datatypes. *Fuzzy Sets and Systems*, doi:10.1016/j.fss.2009.03.006.
- [7] B.P. Buckles, F.E. Petry. A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems*, 7: 213–216, 1982.
- [8] B.P. Buckles, F.E. Petry. Fuzzy databases and their applications. *Fuzzy Inform. Decision Process.*: 361–371, 1982.
- [9] B.P. Buckles, F.E. Petry. Uncertainty models in information and database systems. *J. Inform. Sci.*, 11: 77–87, 1985.
- [10] Pablo Cordero, Manuel Enciso, Inmaculada P. de Guzmán, and Angel Mora. Slfd logic: Elimination of data redundancy in knowledge representation. *LNAI 2527, Springer-Verlag*, 141–150, 2002.
- [11] Petr Cintula, Francesc Esteva, Joan Gispert, Lluís Godó, Franco Montagna, Carles Noguera. Distinguished algebraic semantics for t-norm based fuzzy logics: Methods and algebraic equivalencies. *Annals of Pure and Applied Logic*, 160 (1), 53–81, 2009.
- [12] Pablo Cordero, Angel Mora, Manuel Enciso, Gabriel Aguilera, I.P. de Guzmán. Reasoning about Fuzzy Functional Dependencies. *ESTYLF*: 121–126, 2008.

- [13] Pablo Cordero, Manuel Enciso, Angel Mora, Gabriel Aguilera, I.P. de Guzmán. A Complete Logic for Fuzzy Functional Dependencies over Domains with Similarity Relations. *Lecture Notes In Computer Science*: 5517, 261–269, 2009.
- [14] C. Giardina, I. Sack, D. Sinha. Fuzzy Field Relational Database. *Report 8332, Elect. Engng. and Computer Science Dept., Stevens Institute of Technology, Hoboken, NJ.*, 1983.
- [15] Rolly Intan, Masao Mukaidono. Fuzzy Conditional Probability Relations and their Applications in Fuzzy Information Systems *Knowledge and Information Systems*: 6, 345–365, 2004.
- [16] Angel Mora, Manuel Enciso, Pablo Cordero, and Inmaculada P. de Guzmán. The functional dependence implication problem: optimality and minimality. an efficient preprocessing transformation based on the substitution paradigm. *LNAI, Springer-Verlag*, 3040, 2004.
- [17] Angel Mora, Gabriel Aguilera, Manuel Enciso, Pablo Cordero, and Inmaculada P. de Guzmán. A new closure algorithm based in logic: SLFD-Closure versus classical closures. *Inteligencia Artificial, Revista Iberoamericana de IA*, 31 (10): 31–40, 2006
- [18] H. Prade. The connection between Lipskios approach to incomplete information data bases and Zadeh's Possibility Theory. *Proc. Int. Conf. Systems Meth.*: 402–408, 1982.
- [19] H. Prade, C. Testemale. Generalizing database relational algebra for the treatment of incomplete-uncertain information and vague queries. *2nd NAFIP Workshop*, 1983.
- [20] H. Prade, C. Testemale. Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences*, 34: 115–143, 1984.
- [21] H.V.S.V.N. Raju, A.K. Mazumdar. Fuzzy dependencies and losless join descomposition of fuzzy relational database systems. *ACM Transaction of Database Systems*, 13 (2): 129–166, 1988.
- [22] Mustafa Ilker Sozat and Adnan Yazici. A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations. *Fuzzy Sets and Systems*, 117 (2):161–181, 2001.
- [23] B.K. Tyagi, A. Sharfuddin, R.N. Dutta, Devendra K. Tayal. A complete axiomatization of fuzzy functional dependencies using fuzzy function. *Fuzzy Set and Systems*, 151(2): 363–379, 2005.
- [24] Sadok Ben Yahia, Habib Ounalli, Ali Jaoua. An Extension of Classical Functional Dependency: Dynamic Fuzzy Functional Dependency. *Information Science*, 119 (3-4):219–234, 1999.
- [25] Zadeh, Lotfi, Is there a need for fuzzy logic? *Information Science*, 178(13): 2751–2779, 2008.

INCLUSIÓN DE CONJUNTOS BORROSOS EN EL NÚCLEO DEL SISTEMA BOUSI~PROLOG. *

Pascual Julián-Iranzo¹ Clemente Rubio-Manzano¹ Juan Gallardo-Casero²

¹ Paseo de la Universidad, 4; 13071 Ciudad Real, {Pascual.Julian,Clemente.Rubio}@uclm.es

² Paseo de la Universidad, 4; 13071 Ciudad Real, Juan.Gallardo@alu.uclm.es

Resumen

Bousi~Prolog es un lenguaje de programación lógica borrosa cuyo objetivo primordial es flexibilizar el proceso de respuesta a preguntas. Su mecanismo operacional es una extensión del principio de resolución SLD, que denominamos *resolución SLD débil*, donde la unificación clásica se ha sustituido por un algoritmo de unificación borrosa. En el presente trabajo se especifica un método genérico para la unificación de conjuntos borrosos que también es aplicable a otros lenguajes que se apoyen en un mecanismo de resolución débil. La idea es compilar la información relativa a los subconjuntos borrosos generando una relación borrosa entre sus etiquetas lingüísticas asociadas. Posteriormente, esta relación borrosa se emplea de forma estándar y completamente integrada en el mecanismo de unificación borrosa del sistema Bousi~Prolog, lo que permite tratar las etiquetas lingüísticas en pie de igualdad con otras constantes definidas en el código fuente de un programa. Esta aproximación es novedosa ya que es la primera vez que se introducen los conjuntos borrosos en el núcleo de un sistema Prolog mediante el empleo de relaciones borrosas. Además, tiene la ventaja de su sencillez, ya que la inclusión se hace de manera muy natural, sin afectar a la semántica operacional del lenguaje Bousi~Prolog y con apenas modificaciones sintácticas. Esto la convierte en una buena alternativa a la empleada por otros sistemas Prolog borrosos.

Palabras Clave: Programación Lógica Borrosa, Prolog Borroso, Unificación Borrosa, Relaciones de Proximidad, Conjuntos Borrosos.

^{*}This work has been partially supported by FEDER and the Spanish Science and Innovation Ministry under grant TIN 2007-65749 and by the Castilla-La Mancha Regional Administration under grant PII109-0117-4481.

1 INTRODUCCIÓN

La Programación Lógica Borrosa es un área de investigación que estudia como introducir conceptos procedentes de la lógica borrosa en el seno de la programación lógica, con el fin de poder tratar, de manera explícita, la incertidumbre y/o la vaguedad presente en el mundo real. Existen múltiples líneas de trabajo que oscilan entre las que modifican el mecanismo de resolución clásico y lo sustituyen por un mecanismo de resolución borrosa [2, 13, 17] y aquellas que extiende el principio de resolución SLD, modificando su algoritmo de unificación clásico y sustituyéndolo por un algoritmo de unificación borrosa [1, 6, 15, 16].

Bousi~Prolog (BPL, para abreviar) [8, 9, 10] es una extensión del lenguaje Prolog. Su semántica operacional es una adaptación de principio de resolución donde la unificación clásica se reemplaza por un algoritmo de unificación borrosa basado en relaciones de proximidad, lo que nos permite generalizar propuestas como las que aparecen en [6, 15]. Informalmente este algoritmo de *unificación débil* establece que dos términos $f(t_1, \dots, t_n)$ y $g(s_1, \dots, s_n)$ unifican débilmente si f y g son próximos y cada uno de sus argumentos unifican débilmente. Actualmente existen dos implementaciones¹: una implementación de alto nivel [8], que es un meta-intérprete escrito en Prolog, y otra de bajo nivel [9, 11], que incorpora el algoritmo de unificación borrosa en una adaptación de la Máquina Abstracta de Warren y está escrita en Java. El siguiente ejemplo sirve para ilustrar la sintaxis y semántica del lenguaje.

Ejemplo 1 Supongamos una base de datos que almacena información sobre películas y preferencias de los usuarios codificadas mediante ecuaciones de proximidad:

```
%%DIRECTIVAS
:-transitivity(yes).
```

¹Bousi~Prolog esta disponible libremente en la URL <http://www.inf-cr.uclm.es/www/pjulian/bousi.html>

```
%%HECHOS
pelicula(el_senor_de_los_anillos,aventuras).
pelicula(terminator,accion).
pelicula(stargate,ciencia_ficcion).
```

```
%%ECUACIONES DE PROXIMIDAD
aventuras~accion=0.9.
aventuras~ciencia_ficcion=0.8.
```

Ante la pregunta “?-pelicula(stargate,accion)” un sistema Prolog clásico fallaría. Sin embargo Bousi~Prolog permite obtener la respuesta “Yes with 0.8”, ya que este objetivo unifica débilmente con el hecho “pelicula(stargate,ciencia_ficcion)”. Esto es posible debido a que la directiva “transitivity” activa, en tiempo de compilación, la generación automática de la clausura transitiva de la relación inicial, produciendo la entrada “accion~ciencia_ficcion = 0.8”.

El presente trabajo surge ante la necesidad de mejorar las facilidades de que actualmente dispone el sistema BPL para el tratamiento de la vaguedad y el razonamiento aproximado. Se pretende implementar el dominio conjunto borroso en el seno del sistema BPL. Dotar a Bousi~Prolog con esta característica amplía su área de aplicación permitiendo abordar aplicaciones de control borroso, razonamiento aproximado, y en definitiva cualquier aplicación donde se necesite gestionar la vaguedad aprovechando la potencia conjunta de los conjuntos borrosos y los lenguajes declarativos. Este es pues un paso más en el objetivo de afianzar a Bousi~Prolog como un auténtico lenguaje de programación lógica borrosa útil para el *Soft Computing*.

Para llevar a cabo este cometido se parte del concepto formal de variable lingüística [18] y se desarrolla lo que puede verse como un nuevo método para la unificación de conjuntos borrosos. Una *variable lingüística* es, en esencia, una estructura compuesta por un conjunto de términos o etiquetas lingüísticas y cuyo componente semántico son los subconjuntos borrosos asociados a las mismas. El método desarrollado consiste en convertir, en tiempo de compilación, la información almacenada por los subconjuntos borrosos en una relación borrosa sobre el conjunto de etiquetas lingüísticas. De ese modo, las etiquetas lingüísticas pasan a ser tratadas de forma análoga al resto de los símbolos del lenguaje por el mecanismo de unificación débil (que se apoya en el manejo de relaciones borrosas). Esta aproximación es útil para aquellos lenguajes de programación lógica borrosa, como LIKELOG [5], o SiLog [12], que también usan un mecanismo operacional basado en la resolución SLD débil. En este trabajo se ha estudiado con detalle como incluir esta aproximación genérica en el núcleo de la implementación de bajo nivel [9] del sistema BPL.

2 UNA APROXIMACIÓN GENÉRICA.

En este apartado proponemos una aproximación genérica para la inclusión de conjuntos borrosos en el marco de los lenguajes de programación lógica borrosa que emplean un mecanismo operacional basado en algún tipo de resolución débil, como es el caso del propio lenguaje Bousi~Prolog. Formalmente, nuestra aproximación se apoya en los concepto de variable lingüística [18] y relación borrosa. La idea principal de nuestro planteamiento reside en manejar las variables lingüísticas mediante relaciones borrosas, esto es, transformar la información representada por una variable lingüística para que pueda tratarla una relación borrosa. Nuestra propuesta no sólo nos permitirá la inclusión de los conjuntos borrosos de manera sencilla, sino que además tiene en cuenta la distinción entre el conocimiento específico y general planteado en [14] y soluciona problemas de implementación planteados en [16].

Una *variable lingüística* es una estructura formada por la tupla $\langle X, T(X), U, G, M \rangle$, donde: X es el nombre de la variable, $T(X)$ es el conjunto de nombres de valores lingüísticos que puede tomar X , U el dominio o universo de discurso, G es una gramática que permite generar $T(X)$ y M es una regla semántica que asocia a cada nombre de valor lingüístico x de $T(X)$ su significado, esto es, un conjunto borroso sobre U (caracterizado por su función de pertenencia μ_x).

Respecto al concepto de variable lingüística conviene hacer las siguientes puntualizaciones: $T(X)$ constituye el componente sintáctico de X , sus elementos son *términos lingüísticos*; los términos se generan mediante la gramática G (que por el momento dejamos sin especificar); se distingue entre *términos atómicos* (o *primarios*) y *términos compuesto*, que contienen términos atómicos; el dominio U es un conjunto ordinario, no necesariamente numérico; la regla semántica M permite asociar significado a los términos compuesto, mediante cálculos precisos, a partir del significado de sus términos atómicos; el significado de los términos atómicos se establece de manera axiomática asignando a cada término atómico un conjunto borroso sobre U . En otras palabras, los subconjuntos borrosos que M aplica a los términos compuesto son calculados, mientras que los que aplica a los términos atómicos son definidos (de manera subjetiva y dependiente del contexto).

Una relación borrosa binaria, $\mathcal{R}$, sobre un conjunto ordinario U , es un subconjunto borroso sobre $U \times U$ (esto es, una aplicación $U \times U \rightarrow [0, 1]$). En este trabajo estamos interesados en definir relaciones borrosas sobre el conjunto de términos, $T(X)$, asociado

a una variable lingüística, X , de forma que ésta (incluido su componente semántico) pueda tratarse a un nivel puramente sintáctico. Con este fin, procedemos del siguiente modo:

Supongamos que $T(X) = \{x_i \mid i \in I\}$, donde I es un conjunto de índices. Para cada x_i y x_j , con $i, j \in I$, generamos la entrada de una relación borrosa sobre $T(X)$: $\mathcal{R}(x_i, x_j) = \alpha$. El grado de relación α se halla calculando la relación existente entre los subconjuntos borrosos $M(x_i)$ y $M(x_j)$ asociados a estos términos como significado. Para ello, pueden aplicarse técnicas de emparejamiento (*matching*) borroso [3, 4], empleadas con éxito en el sistema **FuzzyClips** [7].

Una vez generada la relación borrosa $\mathcal{R}$, el mecanismo operacional del lenguaje manipula la variable lingüística X y, en particular, los términos de $T(X)$ de manera totalmente estándar. Esto es, como términos (constantes) de un lenguaje de primer orden que son susceptibles de participar en un proceso de unificación débil en pie de igualdad con el resto de los otros símbolos del alfabeto de ese lenguaje.

Es importante destacar que la relación borrosa construida debe cumplir la propiedad reflexiva pero las propiedades simétrica y transitiva no tienen por que cumplirse. El empleo de este tipo de relaciones en un proceso de unificación borrosa contextual (de naturaleza diferente al procedimiento de unificación débil que nosotros empleamos) fue planteado por primera vez en [14], y ampliado en [1], donde se explica que se debe hacer distinción entre el conocimiento general y específico; por ejemplo, dada la variable lingüística *Edad* y dos de sus términos, *joven* y *entre_19_22*, una persona que sea joven no tiene por que estar entre los 19 y los 22 años, pero una persona que esté entre los 19 y los 22 años sí que es una persona joven, por lo tanto, la relación entre *joven* y *entre_19_22* es distinta que la relación entre *entre_19_22* y *joven*, esto es, la relación entre estos términos, no tiene por que cumplir la propiedad simétrica.

3 CONJUNTOS BORROSOS EN EL LENGUAJE BOUSI~PROLOG.

En este apartado nos ocupamos de los detalles concretos de implementación de la estructura variable lingüística en el lenguaje Bousi~Prolog. Comenzamos con los aspectos sintácticos para pasar, después, a explicar las distintas fases de su implementación y las estructuras internas que se generan y facilitan su ejecución.

3.1 Sintaxis.

Todo lenguaje requiere introducir instrucciones específicas que permitan declarar y definir sus estructuras de datos. En el lenguaje Bousi~Prolog se hace uso de dos directivas para declarar y definir la estructura de una variable lingüística X . En realidad solamente se define el componente semántico de la misma: el dominio de discurso U y los conjuntos borrosos que se asocian a los términos primarios de $T(X)$ (el resto, como se verá, se calcula automáticamente). Por otra parte, no haremos distinción entre el componente sintáctico y semántico a efectos de notación, designando con el mismo nombre tanto a la variable lingüística como al dominio de discurso y de igual forma procederemos con los términos y sus valores. Nos remitiremos al contexto para eliminar las ambigüedades que puedan surgir.

La directiva **domain** nos permite definir el dominio o universo de discurso asociado a una variable lingüística. En general utilizamos el mismo nombre tanto para la variable lingüística como para el dominio. Por razones prácticas y porque no es una limitación severa, en esta implementación, solamente tratamos con dominios que son subintervalos de la recta real. La sintaxis concreta de esta directiva es:

```
:-domain(Domain_name(n,m,Magnitude)).
```

donde, **Domain_name** es el nombre del dominio, n y m (con $n < m$) son los límites inferior y superior del subintervalo de los números reales, $[n, m]$, y **Magnitude** es el nombre de la magnitud en la que se miden los elementos del dominio.

Ejemplo 2 La siguiente directiva define un dominio de nombre **age** (*edad*), cuyos valores son los números comprendidos entre 0 y 100, que se miden en **years** (*años*): “:-domain(age(0,100,years))”.

La directiva **fuzzy_set** nos permite definir una lista de subconjuntos borrosos (asociados a términos primarios de una variable lingüística) sobre un dominio determinado (definido previamente). La sintaxis concreta de esta directiva es:

```
:-fuzzy_set(Domain_name, [Subc1(a1,b1,c1[,d1]),
...,
SubcN(an,bn,cn[,dn])]).
```

Los subconjuntos borrosos se definen, dando su nombre y el tipo de función de pertenencia. En estos momentos se pueden definir dos tipos de funciones de pertenencia: la *función trapezoidal*, si usamos cuatro argumentos para definir el subconjunto borroso y la *función triangular* si usamos tres argumentos. Es importante destacar que, en general, la función trapezoidal o triangular se adapta bastante bien a la definición de cualquier concepto, con la ventaja de su fácil representación y simplicidad de cálculos.

Ejemplo 3 La siguiente directiva define una lista de subconjuntos borrosos (asociados a términos primarios de una variable lingüística *Edad*) cuyo universo del discurso es el dominio *edad*, definido anteriormente en el Ejemplo 2.

```
:-fuzzy_set(age, [young(0,0,20,30),
                  middle(20,40,55,70),
                  old(60,80,100,100)]).
```

En concreto define tres subconjuntos borrosos asociados a los términos primarios *young* (joven), *middle* (maduro) y *old* (viejo), cuya función de pertenencia es una función trapezoidal caracterizada por sus argumentos.

Una vez definido un dominio y los conjuntos borrosos asociados a términos primarios, a través de la siguiente gramática se pueden generar los términos lingüísticos compuestos:

```
<Termino> ::= <Term_atómico>
            | <Term_compuesto>.
<Term_compuesto> ::= <ModifT>#<Term_atómico>.
<ModifT> ::= very | somewhat | more_or_less
            | extremely.
```

Ejemplo 4 Para la variable lingüística *Edad* definida en los ejemplos 2 y 3, algunos de los términos compuestos generados a partir de los primarios serían: *very#young*, *more_or_less#middle* o *extremely#old*.

Adicionalmente, en un programa BPL, pueden referenciarse lo que denominamos puntos de dominio y rangos de dominio. Un *punto dominio* es nuestra forma práctica de difuminar valores concretos del dominio de discurso², de forma que puedan ponerse en comparación con el resto de términos lingüísticos. Los puntos de dominio se generan mediante la siguiente gramática:

```
<Punto_dominio> ::= <Nom_dominio>#<Val_dominio>
                  | <P_D_compuesto>.
<P_D_compuesto> ::= <ModifP>#<Punto_dominio>.
<ModifP> ::= about.
```

donde <Val_dominio> es el conjunto de valores del dominio. Nótese que un punto de dominio puede convertirse en la etiqueta lingüística asociada a un subconjunto borroso cuando le precede el modificador *about*.

Un rango de dominio es una etiqueta para un intervalo de elementos de un dominio, comprendidos entre dos valores que lo limitan superior e inferiormente. Las etiquetas de rango de dominio se genera mediante la siguiente gramática:

```
<Rango_dominio> ::= <Ran_dom_compuesto>
                   | <Nom_dominio>#<Val_dominio>#<Val_dominio>.
```

²Avanzamos que, en esta tarea, se utilizarán las funciones de pertenencia definidas para cada subconjunto difuso de un dominio.

```
<Ran_dom_compuesto> ::= <ModifR>#<Rango_dominio>.
<ModifR> ::= about.
```

Como en el caso anterior, un rango de dominio puede difuminarse mediante la adición del modificador *about*.

Ejemplo 5 Siguiendo con el Ejemplo 2 y la variable lingüística *Edad*, ejemplos de puntos de dominio serían: *age#22* y *about#age#22*. Ejemplos de rangos de dominio serían: *age#20#30* y *about#age#20#30*.

3.2 Compilación de Conjuntos Borrosos.

Una de las características más notables del sistema BPL es su capacidad de compilar toda la información relativa a las variables lingüísticas definidas por un programa. En este apartado se detallan las diferentes fases de la compilación de conjuntos borrosos.

Análisis sintáctico.

Durante la fase de análisis sintáctico se comprueba si existe algún error sintáctico en el programa y se crea un árbol sintáctico, que es la base para la generación de código posterior. Adicionalmente, en esta fase, se leen las directivas *domain* y *fuzzy.set* y se crean los dominios y los subconjuntos borrosos asociados a los mismos. La directiva *domain* dispara un procedimiento que crea un objeto de tipo dominio. Un objeto dominio se compone de un nombre, un rango y una magnitud. La directiva *fuzzy.set* lanza un procedimiento que crea objeto de tipo conjunto borroso. Un objeto de tipo conjunto borroso se compone de un dominio (referencia al dominio creado previamente) y de una lista de subconjunto borrosos. Cada subconjunto borroso se compone a su vez de una etiqueta lingüística que lo identifica (y que podrá ser usado posteriormente como una constante) y de una función de pertenencia que queda determinada por los parámetros que se pasan en los argumentos de una función trapezoidal $f(x, a, b, c, d) = \max(\min((x - a)/(b - a), 1, (d - x)/(d - c)), 0)$ (o triangular $f(x, a, b, c) = \max(\min((x - a)/(b - a), (c - x)/(c - b)), 0)$). Conceptualmente, puede considerarse que este proceso desencadena la creación de una tabla de términos lingüísticos, junto con sus significados, como la que ilustra la Figura 1.

También durante la fase de análisis sintáctico se detecta la aparición de términos lingüísticos compuestos, rangos de dominio y puntos de dominio, que se incluyen en la tabla de términos lingüísticos. El significado asociado a los términos lingüísticos compuestos es una función de pertenencia dependiente del modificador empleado sobre un determinado término primario y se resume en la tabla de la

Término	Dominio	Función de pertenencia			
$TermLi_1$	Dom	a_1	b_1	c_1	d_1
$TermLi_2$	Dom	a_1	b_1	c_1	d_1
$\vdots$					
$TermLi_n$	Dom	a_n	b_n	c_n	d_n

Figura 1: Tabla de términos lingüísticos: representación en memoria de un fragmento asociado a la variable Dom.

Modificador	Desc.	Modificador	Desc.
very(y)	y^2	more_or_less(y)	$\sqrt{y}$
somewhat(y)	$y^{0.333}$	extremely(y)	y^3

Figura 2: Significado de términos lingüísticos compuestos.

Figura 2. El significado asociado a un rango de dominio $domain\_name\#a\#b$ es una función de pertenencia μ definida por una función trapezoidal $f(x, a, a, b, b)$. Esto es, $\mu(x) = 1$, si $a \leq x \leq b$, y $\mu(x) = 0$ en otro caso. En otras palabras un rango de dominio es un subconjunto *crisp*. Como se ha indicado en el Apartado 3.1, un rango de dominio admite el modificador *about*. A la etiqueta $about\#domain\_name\#a\#b$ se le asocia un subconjunto borroso definido por la función trapezoidal $f(x, \max(a - F, n), a, b, \min(b + F, m))$, donde $F = (m - n) \times 2.5/100$ es un factor de difuminación del subconjunto crisp del que se parte. Recordamos que n y m son los extremos del intervalo $[n, m]$ que constituyen los valores del dominio de discurso. A un punto de dominio no se le asigna significado, ya que no es propiamente un subconjunto borroso (lo que simbolizaremos con una entrada nula, $\perp$, en el campo de función de pertenencia de la tabla de términos lingüísticos). Esto nos obligará a un tratamiento particularizado de este tipo de términos en la siguiente fase. Un punto de dominio también admite el modificador *about*. A la etiqueta $about\#domain\_name\#a$ se le asocia un subconjunto borroso definido por la función triangular $f(x, \max(a - F, n), a, \min(a + F, m))$, donde F (definido como en el caso anterior) es un factor de difuminación que convierte el valor a en un subconjunto borroso.

Para finalizar este subapartado, hacemos notar que solamente se almacena información de aquellos términos lingüísticos que aparecen en el código fuente de un programa BPL. Esto es, no es necesario generar, por ejemplo, todos los términos compuestos derivados de uno atómico.

Ejemplo 6 Siguiendo con el Ejemplo 2 sobre la variable lingüística *Edad*, Supongamos que tenemos un fragmento de una base de datos que almacena el nom-

Término	Dominio	Función de pertenencia
<i>young</i>	<i>age</i>	μ_{young}
<i>middle</i>	<i>age</i>	μ_{middle}
<i>old</i>	<i>age</i>	μ_{old}
28	<i>age</i>	$\perp$
<i>very-young</i>	<i>age</i>	μ_{very_young}

Figura 3: Tabla generada tras el análisis sintáctico.

bre de las personas y sus edades: `person(john,young)`, `person(mary,age#28)`, `person(paul,about#age#25)`, `person(warren,very#young)`. El resultado de esta fase de la compilación se ilustra en la tabla de la Figura 3.

Generación de relaciones borrosas.

Una vez creada la tabla de términos lingüísticos, la siguiente fase se centra en la generación de relaciones borrosas entre términos lingüísticos con el fin de compilar toda la información semántica asociada a dichos términos. El proceso de generación se implementa mediante el siguiente algoritmo en el que, por simplicidad, nos ceñimos a los términos lingüísticos de una variable lingüística:

Algoritmo 1

Input: El conjunto de términos/significados de un dominio, $\{\langle x_i, \mu_{x_i} \rangle \mid i \in I\}$, donde I es un conjunto de índices.

Output: Un conjunto $\mathcal{R}$ de entradas que definen una relación borrosa entre términos lingüísticos.

Initialization: $\mathcal{R} := \emptyset$

For each $\langle x_i, \mu_{x_i} \rangle$ **&** $\langle x_j, \mu_{x_j} \rangle$, **with** $i, j \in I$, **do**

Case of

- $\mu_{x_i} \neq \perp$ **&** $\mu_{x_j} \neq \perp$:
 $\mathcal{R} := \mathcal{R} \cup \{\mathcal{R}(x_i, x_j) = \text{matching}(\mu_{x_i}, \mu_{x_j})\}.$
- $\mu_{x_i} \neq \perp$ **&** $\mu_{x_j} = \perp$:
Let $x_j = domain\#u_j$ **in**
 $\mathcal{R} := \mathcal{R} \cup \{\mathcal{R}(x_i, x_j) = \mu_{x_i}(u_j)\}.$
- $\mu_{x_i} = \perp$ **&** $\mu_{x_j} \neq \perp$:
Let $x_i = domain\#u_i$ **in**
 $\mathcal{R} := \mathcal{R} \cup \{\mathcal{R}(x_i, x_j) = \mu_{x_j}(u_i)\}$

Return $\mathcal{R}$

En el algoritmo anterior, la función de emparejamiento, *matching*, calcula el grado de relación entre dos subconjuntos borrosos $\mathcal{F}$ y $\mathcal{F}'$, en términos de una medida de su similitud S , que, siguiendo a [3, 4], se apoya en las nociones de posibilidad P y necesidad N . Más precisamente, esta función ha sido definida como sigue: $\text{matching}(\mathcal{F}, \mathcal{F}') =$

$$\begin{cases} P(\mathcal{F}, \mathcal{F}') & \text{si } N(\mathcal{F}, \mathcal{F}') > 0.5; \\ (N(\mathcal{F}, \mathcal{F}') + 0.5) * P(\mathcal{F}, \mathcal{F}') & \text{en otro caso.} \end{cases}$$

Term.1	Term.2	Grado de relación
young	young	1.0
young	very#young	0.9
young	age#28	0.2
young	about#age#25	0.25

Figura 4: Entradas para los términos del Ejemplo 8.

donde, $P(\mathcal{F}, \mathcal{F}') = \max\{\min(\mu_{\mathcal{F}}(u), \mu_{\mathcal{F}'}(u)) \mid u \in U\}$ y $N(\mathcal{F}, \mathcal{F}') = 1 - P(\overline{\mathcal{F}}, \mathcal{F}')$, siendo $\overline{\mathcal{F}}$ el complemento de $\mathcal{F}$ descrito por la función de pertenencia: $\mu_{\overline{\mathcal{F}}}(u) = 1 - \mu_{\mathcal{F}}(u)$, $\forall u \in U$.

Ejemplo 7 A partir de la tabla de la Figura 3, el Algoritmo 1 genera las entradas que definen una relación borrosa, $\mathcal{R}$, que se muestran en la tabla de la Figura 4.

Concluida la fase de compilación, toda la información que contienen los conjuntos borrosos se ha almacenado en las relaciones borrosas definidas sobre el conjunto de sus etiquetas lingüísticas. Gracias a este artificio, la ejecución de un programa transcurre de forma estándar siguiendo el mecanismo de resolución débil.

Ejemplo 8 Siguiendo con el Ejemplo 2 sobre la variable lingüística *Edad* y tomando los predicados del Ejemplo 6, ante la pregunta: “?-person(X,young).”, el sistema BPL respondería: “X = john with 1.0; X = mary with 0.2; X = paul with 0.25; X = warren with 0.9”.

4 CONCLUSIONES

Bousi~Prolog es un lenguaje de programación lógica borrosa cuyo objetivo es el tratamiento de la información imprecisa o vaga mediante técnicas declarativas. Todo sistema Prolog Borroso debería permitir la manipulación de conjuntos borrosos y, además, lo debería hacer de una manera sencilla y eficiente. En este trabajo se ha explicado con detalle como incluir los conjuntos borrosos en el sistema BPL. La idea general ha consistido en definir relaciones borrosas sobre el conjunto de términos, $T(X)$, asociado a una variable lingüística, X , de forma que ésta (incluido su componente semántico i.e., los subconjuntos borrosos) pueda tratarse a un nivel puramente sintáctico en el contexto de un programa lógico. Esta forma de implementación es sencilla, elegante y eficiente, permitiendo el uso de los conjuntos borrosos de manera muy natural (en el contexto de los lenguajes cuya semántica operacional es la resolución SLD débil). Por otra parte, esta aproximación es novedosa ya que es la primera vez que se introducen los conjuntos borrosos en el núcleo de un sistema Prolog mediante el uso de relaciones borrosas.

Referencias

- [1] T. Alsinet and L. Godo. Fuzzy Unification Degree. In *Proc. 2nd Int Workshop on Logic Programming and Soft Computing'98, Manchester (UK)*, page 18 (1998).
- [2] J. F. Baldwin, T. P. Martin, and B. W. Pilsworth. The implementation of FProlog – a fuzzy prolog interpreter. *Fuzzy Sets and Systems*, 23, pp 119–129 (1987).
- [3] M. Cayrol, H. Farenicy, and H. Prade. Fuzzy pattern matching. *Kybernetes*, 11:103-106 (1982).
- [4] D. Dubois, H. Prade, and C. Testemale. Weighted fuzzy pattern matching. *Fuzzy Sets and Systems*, 28:313-331 (1988).
- [5] F.A Fontana and F. Formato. Likelog: A logic programming language for flexible data retrieval. In *Procs of SAC'99*, pages 260–267, (1999).
- [6] F. Formato, G. Gerla, and M.I. Sessa. Similarity-based unification. *Fundam. Inform.*, 41(4):393–414 (2000).
- [7] R.A. Orchard. FuzzyClips Version 6.04A. User's Guide Integrated Reasoning. Institute for Information Technology. Canada (1998).
- [8] P. Julián, C. Rubio and J. Gallardo. Bousi~prolog: a prolog extension language for flexible query answering. In: *ENTCS*, vol 248, pp. 131-147. Elsevier, Amsterdam (2009).
- [9] P. Julián and C. Rubio. A similarity-based WAM for Bousi~Prolog. In: *LNCS*, vol 5517, pp. 245–252. Springer, Heidelberg (2009).
- [10] P. Julián and C. Rubio. A Declarative Semantics for Bousi~Prolog. In: *Proc. of 11th ACM SIGPLAN Symposium on PPDP'09*. pp. 149–160 (2009).
- [11] P. Julián and C. Rubio. UNICORN: A Programming Environment for Bousi~Prolog. In: *Proc. of PROLE'09*. (2009).
- [12] V. Loia, S. Senatore, and M.I. Sessa. Similarity-based SLD resolution and its implementation in an extended prolog system. In *FUZZ-IEEE*, pages 650–653, 2001.
- [13] M. Mukaidono, Z.L. Shen and L. Ding. Fundamentals of fuzzy Prolog. *Intl. J Approximate Reasons*, 3, pp. 1080–1086 (1989).
- [14] L.G. Rios-Filho and S.A. Sandri. Contextual Fuzzy Unification. In: *Proc. of IFSA'95*, pp. 81–84 (1995).
- [15] M.I. Sessa. Approximate reasoning by similarity-based SLD resolution. *Fuzzy Sets and Systems*, 275:389–426 (2002).
- [16] H.E. Virtanen. Linguistic Logic Programming. In T.P. Martin and F.A. Fontana (Eds): *Logic Programming and Soft Computing*, pp. 91–110, Wiley (1998).
- [17] P. Vojtáš. Fuzzy Logic Programming. *Fuzzy Sets and Systems*, 124 (1): 361-370 (2001).
- [18] L. A. Zadeh. The Concept of a Linguistic Variable and its Applications to Approximate Reasoning I, II and III. *J. of Information Sciences* 8 and 9, Elsevier (1975).

EMD: UNA METODOLOGÍA DE RECUPERACIÓN PARA SISTEMAS DE LENGUAJE NATURAL BASADOS EN CASOS

A.Moreo¹ J.L. Castro¹ V.López¹ J.M. Zurita¹

¹ Departamento de Ciencias de la Computación y la Inteligencia Artificial, Universidad de Granada, {moreo, castro, victor, zurita}@correo.ugr.es

Resumen

Muchas de las aplicaciones que trabajan con el Lenguaje Natural, como los Sistemas de FAQ retrieval, los Sistemas de Diálogo, los Asistentes Virtuales, etc. pueden ser modelados mediante un enfoque de Razonamiento Basado en Casos (CBR), transformando el problema del Question Answering al de buscar el caso de la base de casos más parecido a la pregunta introducida por el usuario. Gran parte de los modelos propuestos se valen de Palabras Clave para tal propósito. Sin embargo, la elección de este conjunto de palabras no es una tarea trivial, puesto que depende del conocimiento experto y posiblemente, las palabras que lo compongan no sean suficientes de cara a la diferenciación de casos.

En este trabajo proponemos un algoritmo que generaliza este enfoque aprendiendo automáticamente un conjunto de expresiones (expresiones diferenciadoras) que definen un criterio de comparación entre preguntas. De esta manera, un algoritmo de búsqueda puede utilizarlas de forma eficiente para recuperar el caso más parecido de la base de casos.

Keywords: Question Answering, FAQ retrieval, Case Retrieval

1 INTRODUCCIÓN

El problema del Question Answering cobra especial interés en la Soft Computing por su capacidad para dotar a los computadores de la habilidad de comprensión de preguntas en Lenguaje Natural (LN) y sus posibilidades en el marco de la recuperación automática de información. Sin embargo, el LN presenta numerosos problemas, discutidos en [1, 2], que es necesario abordar para alcanzar esta meta, entre los que destacan el de la ambigüedad y la desestructuración.

Generalmente se clasifican estos sistemas, atendiendo al enfoque que aplican sobre el conocimiento, en sistemas

de Dominio Abierto (buscan la información en la Web) o de Dominio Cerrado (parten de un modelado delimitado del conocimiento). Nuestro algoritmo se enmarca dentro de los sistemas de Dominio Cerrado.

Una de las estrategias propuestas en este tipo de sistemas consiste en aplicar un modelo Basado en Casos, donde cada Caso se constituye por una expresión lingüística y cierta información adicional. En FALLQ [3] encontramos un ejemplo de este enfoque. El problema queda reducido entonces a distinguir cual de las sentencias de la base de casos es la sentencia más parecida a la introducida por el usuario y la respuesta devuelta es la asociada a dicho caso. El problema se conoce como *Case retrieval*.

Este enfoque es especialmente interesante en el problema del *FAQ retrieval* (recuperación de información automática en conjuntos de “Preguntas Frecuentes”), cuyas principales dificultades son comentadas en el trabajo de Sneiders [4]. Se han propuesto muchos enfoques en esta área, algunos de ellos bajo la filosofía de clustering, como [5] y FAQ Finder [6] en el que se utiliza un modelo de espacio vectorial (VSM) para calcular el grado de similitud combinado con WordNet [12] para gestionar el criterio de matching. Otros trabajos se centran en aplicar técnicas sobre Plantillas, como en los sistemas de Sneiders. En la mayoría de estos sistemas, la dependencia con el conjunto de Palabras Clave (términos relevantes en un dominio particular de conocimiento) es muy fuerte.

El problema de esto es que el conjunto de palabras clave, que depende de conocimiento experto y que a menudo se selecciona a mano, no siempre es buen candidato frente a la distinción entre casos. Se han planteado diferentes metodologías para hacer frente a estos problemas. Podemos encontrar un ejemplo de ello en Auto-FAQ [8], en el que se aplican técnicas de procesamiento del lenguaje para mejorar la comparación entre términos clave. En el sistema de Sneiders, las palabras clave se clasifican en *obligatorias*, *opcionales* e *irrelevantes* de forma automática. Ask Jeeves [14] clasifica el conjunto de FAQs en 7 clases distintas para mejorar la búsqueda mediante palabras clave. Otros enfoques pretenden capturar la semántica de los casos elevando el nivel de representación del conocimiento por medio de ontologías, ayudando de esta forma al proceso de gestión del conocimiento en el problema del FAQ retrieval, como se discute en [9, 10]. Esta propuesta tiene como dificultad

añadida el hecho de que la construcción de ontologías sobre dominios grandes puede llegar a ser una tarea compleja.

El interés del problema de la comparación de preguntas como mecanismo previo para responderlas automáticamente es indiscutible, puesto que la forma más natural de obtención de información en los humanos se basa en el esquema de pregunta/respuesta. Es por esto que supone una fuerte motivación en el marco del E-Learning [7, 13].

En este artículo, proponemos un algoritmo para hallar automáticamente conjuntos de *Expresiones* que operan a modo de clasificadores sobre casos en torno a los cuales se construyen, permitiendo distinguirlos de forma eficiente. Dado que este conjunto de expresiones es muy grande, se pretenden seleccionar de forma automática aquellos más representativos.

El artículo está estructurado de la siguiente manera. En la sección 2 se plantea nuestro enfoque del problema. En la sección 3 definimos los conceptos relacionados con las expresiones y se explica el algoritmo de adquisición de los clasificadores. En el apartado 4 presentamos el algoritmo de búsqueda. Los apartados 5 y 6 están dedicados a los experimentos y conclusiones. Finalmente, la sección 7 recoge las propuestas de Trabajos Futuros.

2 ENFOQUE DEL PROBLEMA

2.1. SISTEMAS BASADOS EN LENGUAJE NATURAL

Este tipo de sistemas son entidades virtuales cuya finalidad es la de proporcionar información a consultas en las que tanto la pregunta como la respuesta se formula en lenguaje natural. Ejemplos de este tipo de sistemas son los Asistentes Virtuales, los FAQs inteligentes, los Puntos de Información turística inteligentes o los Sistemas de Diálogo.

Combinan diferentes técnicas de procesamiento de preguntas, búsqueda o inferencia de información y elaboración de la respuesta. La principal dificultad de estos sistemas estriba en la comprensión del lenguaje natural y en la representación del conocimiento.

2.2. ENFOQUE BASADO EN CASOS

En nuestra propuesta, se aborda el problema de la representación del conocimiento mediante un enfoque de Dominio Cerrado Basado en Casos. El razonamiento basado en casos es un modelo en el que se utiliza la experiencia precedente (*casos*) para resolver nuevos problemas por *similitud* y *adaptación*.

El proceso de comprensión consiste en *recuperar* el caso de la base de casos que mayor similitud guarde con la pregunta introducida por el usuario. El problema de la elaboración de la respuesta se reduce entonces a devolver la respuesta asociada a la pregunta seleccionada. Para ello propondremos una métrica de similitud entre preguntas,

en un enfoque similar a [11], con la ventaja de que puede servir como mecanismo para simular la captación de la semántica de la pregunta.

Cada uno de los casos de nuestra base de casos se compone de un conjunto de expresiones lingüísticas (reformulaciones distintas de una misma pregunta) y un conjunto de índices que determinan dónde se ubica dentro de la estructura de la base de casos. Esta estructura es dependiente del dominio y la define el Ingeniero de Conocimiento. Cuanto más rico sea el conjunto de reformulaciones asociadas al caso, mejor quedará capturada su variedad sintáctica.

Las expresiones lingüísticas de los casos pueden estar descritas de forma *abstracta* utilizando variables que puedan referir a más de un objeto del dominio y que se instanciarán en la etapa de *adaptación*. Por motivos de espacio, obviaremos los *casos abstractos* y abordaremos el problema como si de un *Case retrieval* se tratase.

La etapa de *evaluación* de resultados y aprendizaje es realizada manualmente por un experto, que añade nuevos casos a la base de casos especificando el conjunto correcto de índices asociado.

En un dominio sobre información de Sistemas Inteligentes basados en LN, podemos encontrar el siguiente ejemplo, indexado a dos niveles, de la Tabla 1:

Tabla 1: Ejemplo de Base de Casos sobre SI de LN

Propiedad	Objeto	Reformulaciones
Ventajas	AsistenteVirtual	¿Cuáles son las ventajas de un Asistente Virtual?
	FAQ Inteligente	¿Cuáles son las ventajas de un FAQ inteligente?
Precio	AsistenteVirtual	¿Cuánto cuesta un Asistente Virtual?
		¿Cuánto vale un Asistente virtual?
	FAQ Inteligente	¿Cuánto cuesta un FAQ inteligente?
		¿Cuánto vale un FAQ inteligente?
Mantenimiento	AsistenteVirtual	¿Qué tipo de mantenimiento tienen los Asistentes Virtuales?
		¿Cómo se incrementa el conocimiento de los Asistentes Virtuales?
	FAQ Inteligente	¿Qué tipo de mantenimiento tienen los FAQs inteligentes?
		¿Cómo se incrementa el conocimiento de los FAQs inteligentes?

Esta estructura de casos se muestra en la Figura 1:

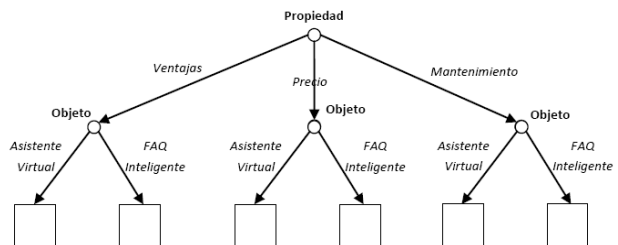


Figura 1: Estructura de la Base de Casos.

Estos índices podrían ser utilizados por un mecanismo de representación del conocimiento más potente como, por ejemplo, las Ontologías.

3 EXPRESIONES MINIMALES DIFERENCIADORAS

En muchos sistemas de Dominio Cerrado, se definen una serie de palabras clave, con la intención de captar la mayor relevancia de éstas frente al resto de términos del dominio. Nuestra intención es deducir conjuntos de palabras extraídos de los casos de forma que permitan distinguirlos al tiempo que representen una reducción semántica y generalicen a otras reformulaciones no contempladas. Existen algunas propuestas en la literatura para conseguir la misma finalidad, como la basada en técnicas estadísticas [15].

El algoritmo busca recubrimientos, que llamaremos expresiones, sobre las sentencias lingüísticas que componen los casos. Se seleccionarán aquellas expresiones que cumplan con las propiedades de diferenciación (subconjunto que únicamente está incluido en el caso del que se extrae) y de minimalidad (es el de menor tamaño).

Supongamos la base de casos a un nivel del ejemplo, con únicamente dos casos, que se muestra en la Tabla 2:

Tabla 2: Base de Casos sencilla.

Caso	Preguntas
Caso1	¿Cuál es el nombre del profesor?
Caso2	¿Cuál es el departamento del profesor?

En este ejemplo, podríamos tomar las expresiones $e_1=\{*\text{nombre}*\}$ y $e_2=\{*\text{departamento}*\}$ para diferenciar los casos. El símbolo '*' representa "cualquier cadena de caracteres". Sin embargo, si existiera un tercer caso "¿Cuál es el nombre de la asignatura que imparte el profesor?", entonces, e_1 dejaría de desambiguar al Caso1. Si redefiniéramos $e_1=\{*\text{nombre}*\text{profesor}*\}$ y añadiéramos $e_3=\{*\text{nombre}*\text{asignatura}*\}$ entonces las expresiones volverían a ser diferenciadoras.

No obstante, existen más expresiones para el Caso 3, como por ejemplo $\{*\text{asignatura}*\}$ o $\{*\text{imparte}*\}$ que podrían ser igualmente válidas, por lo que los conjuntos de expresiones asociadas a los casos contendrán varias expresiones. También podríamos tomar la expresión $\{*\text{nombre}*\text{asignatura}*\text{imparte}*\}$, pero esta expresión no es minimal y en la práctica considerar este tipo de expresiones puede conllevar un gasto computacional inaceptable. Existen otras posibles expresiones diferenciadoras como $\{*\text{la}*\}$, pero ésta no guarda información relevante y debería ser descartada.

Vamos a definir una serie de conceptos que permitirán obtener un conjunto de expresiones útiles.

3.1. DEFINICIONES

3.1.1. Sentencia

Una sentencia S es una lista de palabras no necesariamente únicas. Las palabras pertenecen a una y sólo una de las clases de palabras: K_w (Palabras Clave), S_w (StopWords, conjunto no estándar de palabras que

carecen de significado semántico) o C_w (Comunes o resto de palabras). $W = \{ K_w \cup S_w \cup C_w \}$ es una partición.

$$S=[w_1, w_2, w_3, \dots, w_n], w_i \in W \quad (1)$$

Ejemplo: $\text{dime}_{C_w} \text{las}_{S_w} \text{ventajas}_{K_w} \text{de}_{S_w} \text{un}_{S_w} \text{Asistente}_{S_w}$

3.1.2. Expresión

Una *Expresión* E sobre una *Sentencia* S es una reducción de S compuesta por un subconjunto de sus palabras dispuestas en el mismo orden y que contiene, al menos, una palabra clave si en la sentencia S había al menos una. En el caso de que no hubiera ninguna palabra clave en la sentencia, entonces la única restricción impuesta sobre las palabras que la forman, es que no todas sean StopWords. La notación " $E \exp S$ " representa que E es una expresión de S . El conjunto de todas las expresiones sobre una sentencia es $E(S)=\{e / e \exp S\}$

3.1.3. Expresión Diferenciadora

Una *expresión diferenciadora* e es una expresión que distingue unívocamente la sentencia S de la que ha sido extraída del conjunto de sentencias B del resto de casos de la base de casos (pero no necesariamente de las reformulaciones del caso del que se extrae).

$$e \exp_{\text{dif}} S \text{ wrt } B \leftrightarrow \begin{cases} e \in E(S) \\ \forall S' \in B, e \notin E(S') \end{cases} \quad (2)$$

El conjunto de todas las expresiones diferenciadoras de S con respecto a B es: $ED(S,B) = \{e / e \exp_{\text{dif}} S \text{ wrt } B\}$. Las siglas wrt se corresponden con la expresión del inglés "with respect to".

3.1.4. Expresión Minimal Diferenciadora

Una *Expresión-Minimal-Diferenciadora* (MDE del inglés *Minimal-Differentiator-Expression*) es una Expresión-Diferenciadora no contenida en ninguna otra Expresión-Diferenciadora de la sentencia.

$$e \text{ MDE } S \text{ wrt } B \leftrightarrow \begin{cases} e \in ED(S, B) \\ \forall e' \in (ED(S, B) - \{e\}), e \not\subset e' \end{cases} \quad (3)$$

Nuestro objetivo, es calcular $MDE(S,B)=\{e / e \text{ MDE } S \text{ wrt } B\}$

3.2. OBTENIENDO LAS EMDs

En esta sección, explicamos el algoritmo para obtener las EMDs. Previamente, definimos las etapas de preprocesamiento que se realizan sobre los casos. Por último, mostramos un ejemplo real de ejecución.

3.2.1. Preprocesamiento de la Base de Casos

El preprocesamiento de las sentencias de la base de casos se compone de tres fases. En una primera se eliminan tildes y marcas como los signos de puntuación y se normaliza el caso (todo en minúscula). En la segunda, se reducen *Ngramas* a *tokens*, con el objeto de evitar

interferencias semánticas entre las palabras individuales de conceptos nombrados por más de una palabra. Por ejemplo, “*inteligencia artificial*” podría reducirse a *inteligencia_artificial*. Con esto se consigue, además de reducir el número de palabras, evitar que aparezcan expresiones que contengan únicamente algunas de las palabras de un concepto. Este preprocesamiento se implementa mediante reemplazo por expresiones regulares. En la tercera y última fase, pretendemos capturar el tipo de la pregunta. Suponiendo que éste se deduce de la partícula interrogativa y el verbo más próximo, los unificamos en un token.

Por ejemplo, “¿Qué es la *Inteligencia Artificial*?” quedaría, después de las tres fases anteriores, como “*que_es la inteligencia_artificial*”

3.2.2. Algoritmo

El algoritmo calcula el conjunto MDE(S,B) para todas las sentencias S con respecto a la base B de N casos. Cuenta con un Registro de MDEs calculadas hasta el momento. Este registro permite identificar si una Expresión Diferenciadora engloba a alguna otra ya calculada. Su implementación por medio de Conjuntos Hash hace esta tarea muy eficiente. Las MDEs calculadas para cada Sentencia se van almacenando en el Registro.

Si el conjunto de palabras clave es vacío, o la sentencia no contiene ninguna, entonces el algoritmo trata a todas sus palabras como clave. Por tanto, no es obligatorio definir un conjunto de palabras clave. Este es uno de los puntos fuertes del algoritmo.

El algoritmo utiliza una cola de expresiones candidatas. Para cumplir el requisito de que cada expresión contenga al menos una palabra clave, reordenamos el vector de palabras de la sentencia Q_i^j en un nuevo vector que llamamos KCQ_i^j y que sirve para inicializar la cola únicamente con palabras clave. El número de palabras clave de Q_i^j es z . Cada elemento de KCQ_i^j contiene un índice, consultado por la función $index_{KC}$ que se utiliza para reordenar la expresión al final del algoritmo y antes de incluirla en el Registro:

```

1 Proc MDE( $Q_i^j$ , B, R):
2   MDE:=∅
3   OPENED:=∅
4   For k=1 to z do:
5     Push(OPENED, { $KCQ_{ij}[k]$ })
6   endFor
7   While(OPENED≠∅) do:
8     expr:=pop(OPENED)
9     If (CheckMDE(expr, R, B-{ $Q_i$ }))
10      MDE:=MDE U expr
11    Else
12      If (size(expr)<alpha_bound)
13        For t=indexKC(tail(expr))+1 to k
14          push(OPENED, expr U { $KCQ_{ij}[t]$ })
15        endFor
16      enddo
17   return EMD

```

El factor de ramificación del árbol puede llegar a ser muy elevado, por ello, en la línea 12, podemos el árbol mediante un umbral *alpha* (que establecemos en 5), por lo que no se crearán expresiones con más de 5 palabras.

El procedimiento que calcula el conjunto MDE debe verificar si la expresión candidata está ya registrada en R o está incluida en alguna otra sentencia de otro caso. Si esto ocurre, entonces la expresión no es diferenciadora. *CheckMDE* implementa esta comprobación:

```

1 Proc CheckMDE(expr, R, [ $Q_1, Q_2, \dots, Q_y$ ])
2   expro=QuickShortindex(expr)
3   isMDE:=true
4   If (Registered(R, expro))
5     isMDE:=false
6   Else
7     For (i=1 to y) AND (isMDE) do
8       For (j=1 to size( $Q_i$ )) AND (isMDE) do
9         If (IsSatisfied(expro,  $Q_i^j$ ))
10          isMDE:=false
11        endFor
12      endFor
13   return isMDE

```

3.2.2. Ejemplo

En la tabla 3 se muestra un ejemplo de las EMDs que se obtuvieron de un conjunto de reformulaciones asociadas al punto de información “*Ventajas de un Asistente Virtual*” en nuestros experimentos (sección 5).

Tabla 3: Ejemplo de EMDs obtenidas.

Reformulación	EMDs
¿Qué ventajas tiene un Asistente Virtual?	{*ventajas*}
¿Cuáles son las ventajas de usar un asistente?	{*cuales_son*usar*}, {*las*usar*}
¿Qué ventajas me aportaría comprar un AV?	{*que_aportaria*}, {*me*comprar*}
¿En qué nos puede ayudar un Asistente Virtual?	{*en_que_puede*}, {*ayudar*}
¿Qué aportan vuestros asistentes virtuales?	{*que_aportan*}
¿Qué es lo mejor de los AV?	{*que_es*mejor*}, {*lo*mejor*}
¿Qué supone para mi empresa un Asistente?	{*que_supone*}
¿Qué ventajas me aporta un Asistente Virtual?	{*que_aporta*}
¿Qué ventajas se pueden destacar en los Asistentes Virtuales?	{*destacar*}

4 MATCHING CON EMDs

Nuestro algoritmo de búsqueda opera sobre las EMDs, especialmente diseñadas para mejorar la búsqueda, y no directamente sobre los casos o sobre las palabras clave. El caso ganador, es aquel que maximiza una medida de matching o cercanía entre una sentencia introducida por el usuario y el conjunto de EMDs asociadas al caso. Definiremos primero formalmente las relaciones de inclusión entre sentencias y expresiones para luego describir la búsqueda.

4.1. NIVELES DE COMPARACIÓN Y GRADOS DE INCLUSIÓN

Una sentencia satisface a una expresión si la expresión cubre a la sentencia, es decir, si todos los términos que la forman, están en la sentencia junto con, posiblemente, algunos más. Por ejemplo, la sentencia “¿Qué ventajas tiene un Asistente_Virtual?”, satisface a la expresión $\{ *ventajas *Asistente\_Virtual * \}$. La medida de matching se basa en el grado de inclusión en que esto ocurre, y éste depende de la cercanía entre parejas de palabras y a su relación de orden en ambas listas de palabras.

Para comparar dos palabras w_1 y w_2 usamos un Comparador que devuelve $C(w_1, w_2) = \text{Exacto}$ si son la misma palabra (aceptando diferencias de género, número y caso y distintas conjugaciones para un mismo verbo), $C(w_1, w_2) = \text{Derivado}$ si son sinónimas o están relacionadas jerárquicamente (dos niveles de profundidad), o $C(w_1, w_2) = \text{Diferentes}$, en otro caso. Para implementarlo, hemos utilizado WordNet [13].

Contaremos con un predicado $\text{orden}(S, E)$ que comprueba si las palabras *no Diferentes* contenidas en la sentencia S y la expresión E están dispuestas en el mismo orden.

De acuerdo a la proximidad de las palabras y al orden, la inclusión entre S y E puede ser *Exacta* ($S s_e E$), *Laxa* ($S s_l E$), *Desordenada* ($S s_d E$) o *Desordenada-Laxa* ($S s_{dl} E$), según:

$$S s_e E \leftrightarrow (\forall e \in E, \exists s \in S / C(s, e) = \{\text{Exacto}\}) \wedge \text{orden}(S, E) \quad (4)$$

$$S s_l E \leftrightarrow (\forall e \in E, \exists s \in S / C(s, e) = \{\text{Exacto} \vee \text{Derivado}\}) \wedge (\exists e \in E, \exists s \in S / C(s, e) = \{\text{Exacto}\}) \wedge \text{orden}(S, E) \quad (5)$$

$$S s_d E \leftrightarrow (\forall e \in E, \exists s \in S / C(s, e) = \{\text{Exacto}\}) \wedge \neg \text{orden}(S, E) \quad (6)$$

$$S s_{dl} E \leftrightarrow (\forall e \in E, \exists s \in S / C(s, e) = \{\text{Exacto} \vee \text{Derivado}\}) \wedge (\exists e \in E, \exists s \in S / C(s, e) = \{\text{Exacto}\}) \wedge \neg \text{orden}(S, E) \quad (7)$$

Cuyo valor de matching queda ponderado como sigue:

$$m(S, E) = \begin{cases} 1 & \text{if } S s_e E \\ 0.5 & \text{if } (S s_l E) \vee (S s_d E) \\ 0.25 & \text{if } S s_{dl} E \\ 0 & \text{in other case} \end{cases} \quad (8)$$

4.2. ALGORITMO DE BÚSQUEDA

El matching entre una sentencia S_u introducida por el usuario y un caso C , se calcula de forma aditiva sobre el valor de matching m de S_u con todas las expresiones del conjunto extraídas para cada reformulación incluida en C y que por simplificar llamaremos DME_C .

El valor semántico recogido en una Expresión, es tanto más preciso cuanto mayor sea el número de palabras que la componen. Por esta razón, ponderamos cada valor de matching por la longitud de la expresión.

$$M(S_u, C) = \sum_{e_i \in DME_C} (m(S_u, e_i) \cdot \text{size}(e_i)) \quad (9)$$

Por último, el algoritmo devuelve el caso que maximice la medida M en la base B de casos siempre y cuando éste supere un umbral h de fiabilidad. Es decir:

$$\text{Search}_{MDE}(S_u, B) = \begin{cases} C_i \in B / (M(S_u, C_i) \geq \max_{C_j \in B} \{M(S_u, C_j)\} \wedge M(S_u, C_i) \geq h) \\ \emptyset \text{ en otro caso} \end{cases} \quad (10)$$

La respuesta obtenida ante una pregunta, es la respuesta asociada al caso ganador, o la respuesta vacía si no se infirió ningún caso.

5 EXPERIMENTOS

En esta sección ponemos a prueba nuestro algoritmo enfrentándolo a una implementación del algoritmo de recuperación por conteo de palabras comunes. Si definiéramos un conjunto de palabras clave, entonces el algoritmo de conteo sería un algoritmo de búsqueda básico por palabras clave. Sin embargo, para evitar que los resultados dependan de la calidad de las palabras clave, este conjunto será vacío en ambos algoritmos. Utilizaremos la técnica *10-fold cross-validation*.

5.1. DATOS DE EXPERIMENTACIÓN Y MÉTRICA

Para probar el algoritmo, hemos tomado la base de casos de un Asistente Virtual de una empresa real y hemos establecido un conjunto de pruebas basándonos en los puntos de información del mismo. Partimos el conjunto de casos en dos: por un lado, el *conjunto estático* que utilizamos para aprender el lenguaje y que no intervino en la partición de lotes; por otro lado, tomamos un conjunto de 1500 preguntas realizadas de forma independiente que es el que se usó para crear los lotes.

Se han realizado 10 validaciones completas para tener resultados suficientes para contrastarlos con el *t-test*. En cada *10-fold cross-validation* la base de casos es dividida en 10 subconjuntos independientes y excluyentes entre sí, y todos con idéntica distribución. Cada subconjunto se utiliza una única vez como conjunto de prueba quedando la unión del resto 9 como base. Los resultados se han evaluado mediante la métrica de *precisión* $= \frac{Nc}{Nc+Nf}$ (proporción de los casos clasificados correctamente, Nc , frente a todos los casos clasificados, $Nc+Nf$, donde Nf denota a los erróneamente clasificados).

5.2. RESULTADOS OBTENIDOS

Los resultados que hemos obtenido en media se muestran en la tabla 4, donde *MDE* representa la media de porcentaje de acierto de nuestro algoritmo, y la fila *M*, la media de porcentaje de acierto del algoritmo de conteo. Como puede verse la media de nuestro algoritmo es superior a la del algoritmo de conteo.

Tabla 4: Medias de resultados.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
MDE	90,30	90,57	91,00	91,18	90,36	90,84	90,54	91,49	90,32	91,10
M	70,70	70,89	69,26	68,99	69,75	69,48	70,27	69,71	69,49	68,97

	Media	Desv. Típ.	Error Típ. En Media
EMD	90,774	0,413	0,131
M	69,786	0,669	0,212

Con la intención de verificar que estos resultados son estadísticamente significativos calculamos el *t*-test al nivel de confianza del 95%, y obtuvimos un *p*-value inferior a 0,001. Esto demuestra que existen diferencias significativas en la precisión a favor de nuestro modelo respecto al algoritmo de conteo.

6 CONCLUSIONES

El conjunto de MDEs generadas automáticamente, prescindiendo de palabras clave, supera sustancialmente el porcentaje de acierto de los algoritmos de conteo de palabras clásicos. Al permitir que no se tenga que crear un conjunto de palabras clave, la etapa de modelado del conocimiento, en aquellos problemas en los que esto sea necesario, se agiliza notablemente.

La mejora de porcentaje de acierto que se consigue con el preprocesamiento de *Ngramas* y *Tipo de pregunta* no es estadísticamente mejor. Sin embargo, el tiempo de obtención de MDEs se reduce al bajar el índice de ramificación del algoritmo. Por otro lado, si bien es cierto que el algoritmo depende de las reglas de reducción de *Ngramas* que deben ser definidas a mano, también es cierto que esta dependencia es menor que la que tienen la mayoría de algoritmos con respecto a las palabras clave.

La escalabilidad del algoritmo, en términos de porcentaje de acierto es muy buena. Cuanto mayor sea el número de reformulaciones y de casos, mayor lo será el porcentaje de acierto, ya que las MDEs abstraerán con más precisión. En el límite, considerando una base de casos teórica donde el despliegue de reformulaciones fuese exhaustivo, los clasificadores que se obtendrían serían óptimos, llegando a contener únicamente aquellas palabras que realmente sirvieran para diferenciar unas preguntas de otras.

7 TRABAJOS FUTUROS

La búsqueda automatizada de clasificadores de preguntas que proponemos requiere de más estudio y pruebas. Como tarea futura se considerará un estudio frente a problemas con conjuntos de palabras clave definidas, para evitar que los resultados se vean sesgados por la degeneración de las propiedades del concepto de *Palabra Clave* al asumir que todas lo son.

Nuestros trabajos futuros se centrarán en buscar no solo aquellos elementos que nos informen de qué diferencia unas preguntas de otras sino buscar también aquellos por los que las reformulaciones de un mismo caso se parecen. Podremos utilizar esta información adicional para ampliar el conjunto de reformulaciones en una etapa previa al entrenamiento y capturar algunas relaciones sintácticas que inicialmente no eran accesibles. Para representar esta información utilizaremos *Gramáticas* que generalicen las

reformulaciones de un caso y que nos permitan producir casos nuevos por combinación de estructuras sintácticas. Por otro lado, pretendemos automatizar la creación de las expresiones regulares que dan soporte a la etapa de preprocesamiento de *Ngramas* mediante el análisis de frecuencias de términos consecutivos.

Referencias

- [1] N. Nicolas; B. Kalina; A. Galia; M. Ruslan, IBM T.J. Watson Research Center, Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003
- [2] Advances in Natural Language Processing 5th International Conference on NLP, FinTAL 2006, Proceedings
- [3] M. Lenz, A. Hbner, and M. Kunze, "Question Answering with Textual CBR", Proceedings of the International Conference on Flexible Query Answering Systems (Denmark), pages 236–247, 1998.
- [4] E. Smeiders, "Automated FAQ answering: continued experience with shallow language understanding", AAAI Fall Symposium, 1999, pp.97–107.
- [5] Harksoo Kim, Jungyun Seo, "Cluster-Based FAQ Retrieval Using Latent Term Weights", IEEE INTELLIGENT SYSTEMS,
- [6] K. Hammond, R. Burke, C. Martin and S. Lytinen, "FAQ Finder: A case-based approach to knowledge navigation", Proceedings of the 11th Conference on Artificial Intelligence for Applications, 1995, pp.80–86.
- [7] Deng-Yiv Chiu, Ya-Chen Pan and Wen-Chih Chang, "Using Rough Set Theory to Construct E-Learning FAQ Retrieval Infrastructure"
- [8] S.D. Whitehead, "Auto-FAQ: An Experiment in Cyberspace Leveraging," Computer Networks and ISDN Systems, vol. 28, nos. 1–2, 1995, pp. 137–146.
- [9] S.Y. Yang, "How Does Ontology Help Information Management Processing," WSEAS Transactions on Computers, 5(9), pp. 1843–1850, 2006.
- [10] S.Y. Yang, "How Does Ontology Help User Query Processing for FAQ Services," WSEAS Transactions on Information Science and Applications, 4(5), pp. 1121–1128, 2007.
- [11] Wanpeng Song, Min Feng, Naijie Gu and Liu Wenyin, "Question Similarity Calculation for FAQ Answering", Third International Conference on Semantics, Knowledge and Grid, pp. 298–301
- [12] C. Fellbaum, WordNet: An Electronic Lexical Database, the MIT Press, 1998.
- [13] C.S. Lin "The desing and implementation of network learning environment", Information and Education Volume 67, pp.24–50.
- [14] Ask Jeeves, <http://www.ask.com>
- [15] Frans Coenen, Paul Leng, Robert Sanderson, and Yanbo J. Wang, "Statistical Identification of Key Phrases for Text Classification", Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern

Aplicación de Lógica Difusa en el Proceso de Expansión de Folksonomías

Carmen Martínez-Cruz¹ Ignacio J. Blanco² M. Amparo Vila²

¹ Dpto. Informática, Universidad de Jaén
Campus de Las Lagunillas, 035-A3, 23071, Jaén
cmcruz@ujaen.es

² Dpto. Ciencias de la Computación e I. A., Universidad de Granada
C/ Periodista Daniel Saucedo Aranda S/N, 18071, Granada
{iblanco,vila}@decsai.ugr.es

Resumen

Las folksonomías tienen una naturaleza, carente de semántica que hace muy difícil su utilización y explotación en la Web. Dichas estructuras pueden ser enriquecidas a través de su asociación con términos significativos identificados en la Web Semántica. Sin embargo dicho proceso implica una primera fase de desambiguación y una segunda de expansión del conjunto de etiquetas inicial. En esta propuesta se propone la utilización de la lógica difusa para extender el conjunto de etiquetas pertenecientes a la folksonomía y asociar un grado de pertenencia de cada etiqueta al conjunto en función del grado de similitud de las etiquetas añadidas con el conjunto inicial. Además se permite al usuario, a través del uso de etiquetas lingüísticas, seleccionar el subconjunto de etiquetas que mejor se ajuste a sus necesidades. Por último, también se comparan los dos principales repositorios de información disponibles en la actualidad, *WordNet* y *Wikipedia*, dada su influencia en ambos procesos.

1 Introducción

Las folksonomías[9] juegan un papel muy importante en la Web 2.0 puesto que permiten la descripción de los recursos que en ella se encuentran, p.e. imágenes, URL, noticias, etc. La naturaleza de estas estructuras es completamente opuesta a la de la Web Semántica y las Ontologías, dado que estos conjuntos de etiquetas los especifican los usuarios, normalmente no expertos y no existe ninguna convención de sintaxis o vocabulario. Con estas características, estas estructuras carentes de semántica, son difícilmente procesables y la asociación de semántica a las mismas pasa por establecer relaciones explícitas con ontologías [1, 7].

Una folksonomía puede contener términos que no están

relacionados, p.e. una imagen se puede describir como *amarillo, juguete, transporte, arte, Venezuela*, lo cual provoca que sea realmente complejo entender el verdadero significado de una etiqueta. Incluso un proceso de desambiguación puede conducir a error, a pesar de que los repositorios como *Wordnet* o *Wikipedia* juegan un papel crucial en el mismo dado que proporcionan un gran número de sentidos a cada etiqueta. Desafortunadamente, dichas definiciones son muy similares en muchos casos provocando el aumento de la complejidad del proceso. Por otro lado, las etiquetas usadas pueden ser muy diversas para describir el mismo recurso. El proceso de expansión de una folksonomía puede ayudar a la contextualización de la etiqueta y posteriormente del recurso. Sin embargo este proceso también puede conducir a errores dado que la relación entre los tags iniciales y los expandidos pueden tener ligeras diferencias semánticas, como una relación en hipernimia.

En esta propuesta se propone utilizar la lógica difusa para solventar los problemas descritos: En lugar de descartar los diferentes sentidos de una etiqueta, se conservan los mismos asignándoles un grado de confianza dependiendo de su similitud con el resto del conjunto de etiquetas. Tras esto, el proceso de expansión también asigna un grado de similitud a cada etiqueta añadida al conjunto. Dicho grado se ajusta en función de la relación semántica del término añadido con el conjunto inicial, p.e. sinonimia, hipernimia, meronimia, definiciones, etc. Como resultado, el conjunto inicial expandido puede filtrar los resultados utilizando etiquetas lingüísticas que facilitan al usuario final el proceso de selección de resultados. Además, los repositorios utilizados como fuentes de información (*WordNet* y *Wikipedia*), críticos en este proceso, son utilizados y analizados.

En la sección 2 se describe el proceso de enriquecimiento de folksonomías. A continuación, la definición de la propuesta y un ejemplo analizando los resultados obtenidos se encuentran en las secciones 3

y 4 respectivamente. La definición de las etiquetas lingüísticas se presentan en la sección 5 y finalmente las conclusiones en la sección 6.

2 Fundamentos

2.1 Proceso de Enriquecimiento de Etiquetas

El Enriquecimiento de Folksonomías es una práctica reciente en el campo de la Web Semántica dada la proliferación de este tipo de estructuras en la Web Colaborativa. Dicho proceso, consistente en proporcionar semántica a las etiquetas que componen la folksonomía, conecta la Web 2.0 con la Web Semántica. Existen varias propuestas que desarrollan este *proceso de descubrimiento semántico* en las folksonomías, como la de Specia et al. [7] que agrupa etiquetas relacionadas para relacionarlas con términos de ontologías o la propuesta de Angelou et al. [1] que enriquece una folksonomía limpiada y expandida con los recursos encontradas en diferentes ontologías. Esta última especifica la metodología seguida en este trabajo, resumida en la figura 1.

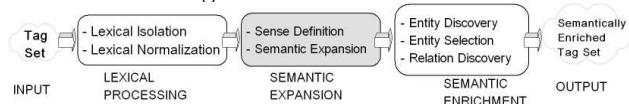


Figura 1: Secuencia del proceso de enriquecimiento de folksonomías en el proyecto *Flor* [1]

2.2 Fuentes de Información

2.2.1 WordNet

Wordnet es un diccionario en línea que modela el idioma inglés. Sus características principales son [6]:

1. Define cada sentido de una palabra, *synset*.
2. Cada *synset* tiene sus sinónimos asociados.
3. Cada *synset* tiene una definición.
4. La hiperonimia se representa para cada *synset* usando una estructura de árbol.
5. Define relaciones de holonimia (*es miembro de*) y meronimia (*tiene miembros*).

El éxito de *WordNet* se debe a la representación tan estructurada que realiza de la información, sin embargo, también presenta las siguientes desventajas [3]:

1. Carece de conexiones entre nombres y verbos y de las relaciones típicas entre las palabras.
2. Carece de relaciones morfológicas.
3. Carece de algunos sentidos y relaciones. También de relaciones temáticas.
4. No tiene una definición uniforme.

5. El primer sentido no es el más lógico, p.e. *tiger*.
6. Tiene una baja cobertura, sobretodo con nombres propios [4].
7. Define demasiados sentidos, y estos suelen tener una granularidad muy fina [4].

2.2.2 Wikipedia

Wikipedia es una enciclopedia en línea desarrollada por una comunidad colaborativa. Sus principales características son [4]:

1. Los artículos representan los sentidos,
2. Los enlaces proporcionan la fuente de sinónimos y otras variaciones de la palabra.
3. Las redirecciones son sinónimos y otras variaciones de la palabra
4. Las categorías representan la jerarquía de conceptos de *Wikipedia* en forma de red.
5. Las páginas de desambiguación se corresponden con los diferentes sentidos de una palabra.
6. La definición está en la primera frase del artículo.
7. Proporciona un repositorio de nombres propios y conceptos especializados [8].
8. Proporciona un gran repositorio de abreviaciones y acrónimos.

Sin embargo, los problemas que presenta son:

1. Las páginas de desambiguación son muy difíciles de procesar automáticamente [4, 5] y existen demasiados sentidos para un término.
2. Los títulos de página incluyen a veces información adicional.
3. Los enlaces incluyen polisemia.
4. Las categorías no representan sólo relaciones ES-UN[8].
5. Los artículos carecen de revisiones formales de expertos.

3 Proceso de Expansión de Etiquetas

3.1 Descripción del Proceso

La *Fase de Expansión Semántica* consiste en expandir un conjunto de etiquetas con otras que tengan un significado similar, para lo cual se necesita un repositorio como *Wordnet* y *Wikipedia*. El proceso tiene dos subfases: i) Definición de los sentidos, la desambiguación

de cada etiqueta se realiza utilizando una función de similitud que compara cada par de etiquetas, ii) Expansión Semántica, en esta fase el conjunto de etiquetas se aumenta con hipéronimos y sinónimos (vase ejemplo en la figura 2).

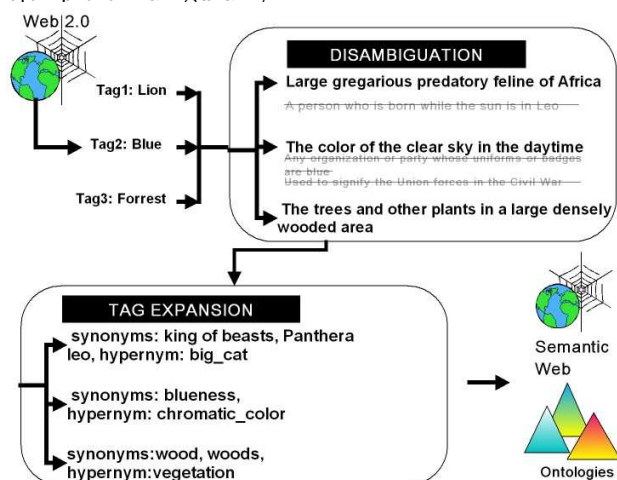


Figura 2: Ejemplo del proceso de expansión [1]

La propuesta descrita en este artículo mejora el proceso anteriormente descrito mediante la definición de un proceso de desambiguación capaz de elegir el número correcto de sentidos del conjunto de etiquetas y una posterior fase de expansión que incluye un grado de similitud en cada etiqueta del conjunto.

Además, el repositorio elegido para desarrollar estos procesos determinará el conjunto de etiquetas resultante, dada su estructura de datos y las relaciones representadas.

3.1.1 Fase 1. Proceso de Desambiguación

El conjunto de sentidos más relacionados con la folksonomía se obtiene en esta fase. Esta tarea presenta los siguientes problemas: i) Cada término tiene numerosos, a veces demasiados sentidos (ejemplo en tabla 3), ii) Las folksonomías se componen por términos que no están relacionados por naturaleza. Por tanto, es muy complicado seleccionar únicamente un único sentido de una palabra. La estructura de datos mostrada en la figura 3 permite gestionar todos los sentidos definidos en el conjunto de etiquetas:

1. Un *Grado de Confianza* que representa la similitud de la etiqueta con el resto del conjunto se define como un conjunto difuso, p.e. *Estrella* en la folksonomía *Estrella, Sol, Azul, Planeta* tiene asociado el conjunto difuso $[0.9, 0.3, 0.8]$ donde 0.9 representa la relación *Estrella-Sol*.
2. Cada sentido tiene asociado un *grado de confianza* que establece la relación de este con el resto del conjunto de etiquetas. Una etiqueta puede tener varios sentidos.

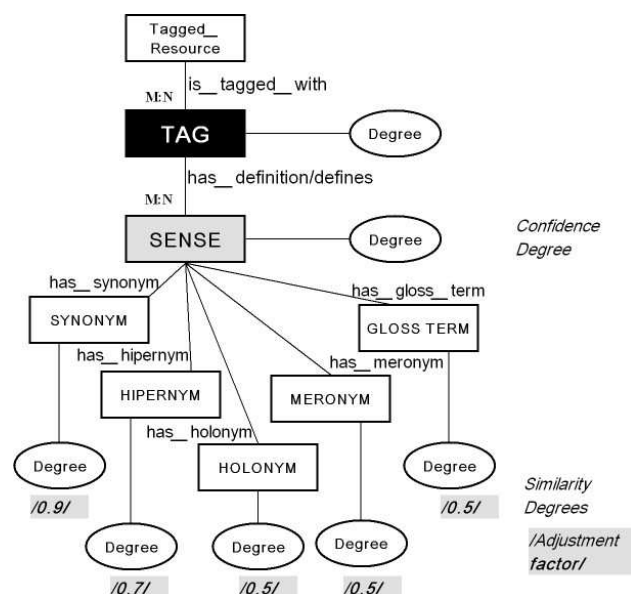


Figura 3: Estructura para extender una folksonomía usando grados difusos

El cálculo del *grado de confianza* consiste en aplicar un proceso de desambiguación donde se valora la adecuación del sentido con la etiqueta. Para ello se calcula la distancia entre cada dos términos dividiendo el camino más corto (hipéronimos) entre su distancia máxima. El *grado de confianza* final de la etiqueta será la *s-norma* (Dubois and Prade [2]) obtenida a partir de los conjuntos difusos de los sentidos que lo describen.

3.1.2 Fase 2: Proceso de Expansión

La fase de expansión incrementa el conjunto de etiquetas con otros términos relacionados. El *grado de similitud* que describe la similitud de las nuevas etiquetas con las originales se asigna de forma siguiente (véase figura 3):

1. Los sinónimos se añaden directamente al conjunto. La *Wikipedia* utiliza redirecciones y enlaces para definirlos.
2. Los hipéronimos presentan dos problemas: i) el número puede ser demasiado grande, ii) el significado puede ser demasiado genérico. En esta propuesta los hipéronimos se ponderan para mantener los mejor valorados.
3. Las definiciones pueden contextualizar una palabra, especialmente en una folksonomía. Sólo se conservan los nombres con una relación mínima con el sentido original. Esta medida se calcula utilizando la distancia Wu and Palmer [10] en *WordNet*.
4. Otros términos como merónimos y homónimos se incluyen. En esta categoría *Wikipedia* incluye en-

laces a otras páginas, sin embargo sólo aquellas con un gran número de referencias se añaden al conjunto.

El *grado de similitud* es el producto del *grado de confianza* del sentido inicial y el correspondiente *Factor de Ajuste*, mostrado en la figura 3. El *Factor de Ajuste* se asigna a cada estructura del sistema de acuerdo con la relación semántica con el sentido inicial.

4 Ejemplo de Aplicación.

4.1 Cálculo de Sentidos y Grados de Confianza

Para describir esta propuesta se muestra el proceso de expansión de una folksonomía de ejemplo compuesta por los términos ya filtrados y descritos en la tabla 1. Una vez el sistema ha obtenido los sentidos para cada etiqueta utilizando un repositorio particular, se calculan sus grados de similitud. Los resultados se muestran en las tablas 2 y 3 para *WordNet* y *Wikipedia* respectivamente.

Tag	WordNet	Wikipedia
Car	5	48
Auto	1	7
Compact	3	14
Environment	2	15

Tabla 1: Número de sentidos de la folksonomía. Para obtener un sentido se compara con todos los demás sentidos de la folksonomía. Este sentido puede guardarse si su grado de similitud es mayor que un umbral de 0.5 y si su relación con alguna de las otras etiquetas del conjunto es mayor que 0.7. Además, el conjunto difuso se obtiene aplicando la s-norma de Dubois and Prade s-norm con el factor de 0.3. Por último, el *grado de confianza final* es la media de los conjuntos difusos respectivos normalizada.

CAR				
sense	Auto	Compact	Environment	Sim.Deg.
1:RoadCar	0.6	0.6	0	0.4/0.8
Snorm	0.6	0.91	0	0.506/1
AUTO				
Sense	Car	Compact	Environment	Sim.Deg.
1:Car, Automobile	0.6	0.9	0	0.5/0.98
Snorm	0.6	0.9	0	0.5/0.98
COMPACT				
Sense	Car	Auto	Environment	Sim.Deg.
1:Economical car	0.5	0.9	0	0.46/0.92
Snorm	0.6	0.9	0	0.5/0.98
ENVIRONMENT				
sense	Car	Auto	Compact	Sim.Deg.
1:Surrounding Cond.	0	0	0	0/0
Snorm	0	0	0	0/0

Tabla 2: Sentidos evaluados usando *WordNet*.

4.1.1 Resultados

WordNet identifica el dominio *Automobiles* y detecta que *Environment* no tiene relación con el conjunto. Sin embargo, *Wikipedia* identifica el dominio de *computer variables*, ya que *Wikipedia* define numerosas categorías relacionadas con cada sentido.

Ambos repositorios identifican correctamente el dominio de *Automobiles* sin embargo *WordNet* ha ex-

CAR				
Sentido	Auto	Compact	Envir.	Sim.Deg.
1:Automobile	0.7	0.7	0.39	0.59/0.8
2:Car and CDR	0.7	0.3	0.7	0.56/0.76
3:CCR and CAR	0.3	0.8	0.6	0.56/0.76
4:Central Apparatus Room	0.39	0.6	0.7	0.56/0.76
Snorm	0.7	0.8	0.7	0.73/1
AUTO				
Sentido	Car	Compact	Envir.	Sim.Deg.
1:Auto Riskshaw	0.7	0.5	0.39	0.53/0.72
2:Automatic Variable	0.7	0.39	0.8	0.63/0.86
Snorm	0.7	0.7	0.8	0.73/1
COMPACT				
Sentido	Car	Auto	Envir.	Sim.Deg.
1:Measure	0.8	0.3	0.39	0.5/0.68
Snorm	0.8	0.5	0.5	0.59/0.80
ENVIRONMENT				
Sentido	Car	Auto	Compact	Sim.Deg.
1:Environment Variable	0.7	0.8	0.3	0.59/0.8
Snorm	0.88	0.8	0.39	0.63/0.86

Tabla 3: Sentidos evaluados usando *Wikipedia*.

cluido el término *Environment* que está actualmente relacionado con el campo *Environmental cars*. Sin embargo, *Wikipedia* ha identificado otro dominio, no correcto pero lógico.

Resumiendo, las categorías de *WordNet* están más especializadas y por tanto, las relaciones definidas entre los términos son mas distantes. Además, en este repositorio el número de definiciones de cada término es menor y por tanto más precisas. *Wikipedia*, al contrario, define un gran número de sentidos que facilita contextualizar los diferentes pares de palabras en el conjunto. Además cada sentido tiene numerosas categorías que provocan la obtención de distancias más cercanas.

4.2 Fase de Expansión

4.2.1 Ejemplo. Etiquetas Resultantes y Grados de Similitud

Una vez evaluadas las etiquetas y sus sentidos, se procede a extender y ponderar el conjunto de etiquetas. Siguiendo con el ejemplo anterior, el conjunto de etiquetas se expande usando *WordNet*, tal y como se muestra en la tabla 4 y con *Wikipedia*, tabla 5.

El cálculo de los grados de similitud se realiza, por ejemplo, para el sinónimo *Motorcar* = Grado de confianza del sentido * factor del sinónimo = $0.98 * 0.9 = 0.88$. Para el hiperónimo es ligeramente distinto, para *Motor-vehicle* el proceso sería: Grado de confianza del sentido * Grado del hiperónimo * Factor del hiperónimo = $0.98 * 0.93 * 0.7 = 0.63$.

4.2.2 Resultados

Wikipedia contiene más datos que *WordNet*. Además, *Wikipedia* no define explícitamente las relaciones que existen entre las palabras, así, los enlaces desde y hacia otros sentidos no tienen un claro significado en el conjunto resultante. Además, la cantidad de sinónimos en la *Wikipedia* es grande dado que este repositorio expresa el mismo sentido de diferente forma, como plurales, acrónimos, abreviaturas, etc.

Por otro lado, los hiperónimos en *Wikipedia* son más

Seq.	Sinónimos	Hiperónimos	Def.	Otros
CAR				
1	[railcar, railway_car, railroad_car]/0.72	wheeled_vehicle/0.50, vehicle/0.44, conveyance/0.39, transport/0.33, instrumentality/0.28]	[wheel]/0.4	[suspension, suspension_system]/0.4 ,[train, railroad_train]/0.4
AUTO				
1	[car, automobile, machine, motorcar]/0.88	motor_vehicle/0.63, self-propelled_vehicle/0.55, wheeled_vehicle/0.40, vehicle/0.35, conveyance/0.30, transport/0.25, instrumentality/0.20	[wheel, engine]/0.4	[accelerator, accelerator_pedal, gas_pedal, gas, throttle, gun, air_bag, auto_accessory, automobile_engine, automobile_horn, car_horn, motor_horn, horn, hooter, buffer, fender, bumper, car_door, car_mirror, car_seat, etc.]/0.49
COMPACT				
1	compact_car/0.82	car/0.61, auto/0.57, automobile/0.59, machine/0.56, motorcar/0.54, motor_vehicle/0.51, automotive_vehicle/0.49, self-propelled_vehicle/0.46, wheeled_vehicle/0.44, vehicle/0.41	[small, economical, car]/0.46	-
ENVIRONMENT				
1		situation/0	condition/0	

Tabla 4: Ejemplo de folksonomía semánticamente expandida y sus etiquetas de similitud usando *WordNet*

Seq.	Sinónimos	Hiperónimos	Def.	Otros
CAR				
1	[Cars, Motorcar, Automobiles, Motor Car, Ottomobile, Automobiles, Motorization, Motorisation, Cardoor, Autos, Automobile, etc.]/ 0.72	Industries/0.18, Travel/0.18, Technology by type/0.18, Transportation/0.34, Marketing/0.18, Manufacturing/0.18, Goods/0.18, Manufactured goods/0.34, Mechanical engineering/0.18, etc.	[wheel, motor, vehicle, engine, motor, transport, goods, university, press, vehicle]/0.4	[wheeled motor vehicle, transporting passengers, engine, transport, vehicle, Russia, Internal combustion, Gustave Trouve, Karl Benz]/0.4
2	[Car (function), Car and cdr] / 0.68	Programming language families/0.17, Lisp programming language family/0.35, Lisp programming language/0.53, etc.	-	[Lisp, Programming, Language]/0.38
3	CCR algebra, CAR algebra, Canonical anticommutation relation, CCR/CAR algebra/0.68	Mathematics/0.17, Mathematical analysis/0.35, Functions and mappings/0.35, Functional analysis/0.53, Physics/0.17, etc.	-	-
4	Central Machine Room, Central machine room, Central apparatus Room, Machine Room/0.68	Electronics/0.17, Communication/0.17, Networks/0.17, Electronic engineering/0.17, Technology by type/0.17, Telecommunications/0.35, etc.	[apparatus, or, or, machine, room, or, facility, gear, house, router, server, etc.]/0.38	[broadcast, television, audio, video routers, compressors, multiplexer]/0.38
AUTO				
1	Auto Rickshaw, Tuk-Tuk, Mototaxi, Auto-wallah, Auto Rickshaws in India, Auto Rickshaws in Pakistan, etc./0.64	South Asian countries, Transport in India, Transport in Pakistan, Transport in Bangkok, Capitals in Asia, Vehicles for hire/0.41, automotive_vehicle/0.5 -Only one level included-	vehicle, hire, chief, mode, transport, transport, south, asia, motorize, etc./0.36	South East Asia, Riskshaw, Bangladesh, motor vehicle, traffic congestion, seabelt, TVS motors, etc./0.36
2	Automatic memory allocation/0.77	Computer programming/0.2, Programming language design/0.2, Programming constructs/0.4, Algebra/0.2, Variables/0.4, Variable (computer programming)/0.6	-	lexally scope, stack grame, processor, registers, Java, C, C++, Perl/0.43
COMPACT				
1		Mathematics/0.15, Mathematical analysis/0.31, Functions and mappings/0.31, Functional analysis/0.6		[volume area, shape, circle]
ENVIRONMENT				
1	SystemRoot, SystemDrive, Environment variables, Env var, LC ALL, Environment (comp. science), etc./0.72	Computing/0.18, System software/0.18, Computer Science/0.18, Operating systems/0.36, Operating system technology/0.56, Computer programming/0.18, etc.	-	vales processes, Unix, Dos, Microsoft Windows, file system, user, command line, bash, computer terminal, etc./0.4

Tabla 5: Ejemplo de folksonomía semánticamente expandida y sus etiquetas de similitud usando *Wikipedia*

específicos que en *WordNet* dado que en *Wikipedia* se define un gran número de categorías. De esta forma alcanzar una categoría "genérica" requeriría obtener un gran número de hiperónimos. Consecuentemente, el número y grado de los hiperónimos en *Wikipedia* será finito y establecido a priori. *Wikipedia*, concretamente, define más categorías para cada sentido y todas ellas son más específicas que las de *WordNet* que son más genéricas y concretas.

Por ltimo, los términos de la definición obtenidos mediante el uso de las categorías, no devuelven buenos resultados dado que los términos de la definición no tienen relación de hiperonimia con los sentidos iniciales.

5 Etiquetas Lingüísticas

El establecimiento de umbrales correctos y flexibles permiten adecuar el número de etiquetas resultantes y sus grados de similitud en el conjunto extendido final

es una tarea compleja. La definición de diferentes etiquetas lingüísticas puede ayudar en este proceso, y así permitir obtener el resultado de manera más intuitiva al usuario. De esta forma se han definido tres tipos de etiquetas: i) *Poco Expandido* dónde los grados de similitud son mayores o iguales que 0.7, ii) *Medianaemente Expandido* donde el umbral está en 0.5, iii) *Muy Expandido* con un umbral de 0.3.

En la 6 se muestra el resultado de haber aplicado la etiqueta *Poco Expandido* sobre el conjunto de etiquetas resultantes mostrado en los ejemplos anteriores.

Thesaurus	Etiquetas Poco Expandidas
<i>WordNet</i>	Car/1, [railcar, railway_car, railroad_car]/0.72, Auto/1, [car, automobile, machine, motorcar]/0.88, automotive_vehicle/0.6, compact/1, compact_car/0.82, environment/1
<i>Wikipedia</i>	Car/1, [Cars, Motor car, Motorcar, Automobiles, Motor Car, Ottomobile, Automobiles, Motorization, Motorisation, Cardoor, Autos, Automobile, Passenger vehicle, A car, Passenger Cars, Car Automobile, Automobilism, Passenger vehicles, Motor-car]/ 0.72, Auto/1, [Automatic memory allocation]/0.77, Compact/1, environment/1, etc./0.72

Tabla 6: Ejemplo de conjunto *Poco Expandido*

6 Conclusiones

El principal objetivo de este trabajo ha sido mejorar el proceso de identificación semántica. Para ello se propone un novedoso proceso que incluye el tratamiento de datos difusos, de tal forma que el conjunto de etiquetas se incremente y cada elemento sea evaluado de acuerdo con su similitud con el conjunto de etiquetas inicial. De esta forma se proporciona al sistema la capacidad de seleccionar aquellas etiquetas que satisfagan las necesidades del usuario sin excluir ningún sentido significativo. Con el uso de esta propuesta, los sentidos ahora no son excluidos sino ponderados y pueden utilizarse en el proceso de identificación de estructuras semánticas en ontologías en las fases posteriores.

La lógica difusa ha proporcionado las herramientas necesarias para representar y ponderar la información relacionada con un conjunto de etiquetas concreto. Cada etiqueta se representa como un conjunto difuso donde se evalúa con precisión la pertenencia de la misma al conjunto de etiquetas inicial.

Por otro lado, *WordNet* proporciona una representación de información muy eficiente, donde se devuelve el tipo de las palabras, las relaciones entre ellas, una finita y bien definida jerarquía de categorías, etc. Sin embargo esta estructura jerárquica de categorías no funciona bien para buscar la relación semántica entre los elementos de la folksonomía, dado que estas estructuras contienen, a menudo, términos que no están relacionados. Como resultado, las folksonomías incluyen términos pertenecientes a diferentes contextos o tienen relaciones de similitud muy bajas.

El catálogo de la *Wikipedia* incluye un gran número de relaciones entre términos, y por lo tanto, grados de similitud elevados. La consecuencia es la opuesta al caso anterior, *Wikipedia* permite al sistema encontrar numerosos contextos donde un conjunto de etiquetas pueden estar relacionadas. Sin embargo, estos resultados son desventajosos dado que el amplio número de sentidos que *Wikipedia* proporciona, hacen el proceso de desambiguación muy difícil.

Por otro lado, el proceso de desambiguación utilizando únicamente hiperónimos no funciona demasiado bien en el caso de las folksonomías dada su naturaleza. Un mecanismo que tenga en cuenta otros términos relacionados como sinónimos, términos de la definición, términos relacionados, etc. pueden mejorar la identificación del contexto y reducir el número de sentidos obtenidos.

El resultante conjunto de etiquetas con sus grados de similitud asociados proporcionan al sistema la posibilidad de elegir aquellas que se ajusten a los requerimientos exigidos por el usuario. Estos gra-

dos también se utilizarán en las fases siguientes donde los grados de similitud se utilizarán para encontrar términos enriquecidos semánticamente. Finalmente, las etiquetas lingüísticas ayudan a escoger el conjunto de etiquetas final dado que dichos conjuntos se representan utilizando expresiones del lenguaje natural.

Agradecimientos

Esta investigación ha sido financiada por los proyectos de la consejería Andaluza de Ciencia y Tecnología TIC1570, TIC1433, TIC03175 y por el proyecto del ministerio con referencia TIN2008-02066. Además se agradece especialmente al profesor Enrico Motta, Sofia Angelotou y al KMI (Knowledge Media Center) en UK, lugar donde se ha realizado esta investigación, por su colaboración en este trabajo.

Referencias

- [1] Sofia Angeletou. Semantic enrichment of folksonomy tagspaces. In *ISWC '08*.
- [2] H. Dubois and H. Prade. Ranking fuzzy numbers in the setting of possibility theory. *Information Sciences*, 30:183–224, 1983.
- [3] Sanda M. Harabagiu, George A. Miller, and Dan I. Moldovan. WordNet 2 – a morphologically and semantically enhanced resource. In *Proc. SIGLEX 1999*, 1999.
- [4] Olena Medelyan, Catherine Legg, David Milne, and Ian H. Witten. Mining meaning from wikipedia, 2008.
- [5] R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In *Proceedings of (NAACL)*, Rochester.
- [6] Ray Richardson, Alan F. Smeaton, and J. Murphy. Using wordnet as a knowledge base for measuring semantic similarity between words. Technical Report CA-1294, Dublin, Ireland, 1994.
- [7] Lucia Specia and Enrico Motta. Integrating folksonomies with the semantic web. pages 624–639. 2007.
- [8] M. Strube and S. Paolo-Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st NCAI*, Boston, USA.
- [9] Thomas Vanderwal. Folksonomy coinage and definition, February 2007.
- [10] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133–138, New Mexico State University, Las Cruces, New Mexico, 1994.

Speeding Up Evolutionary Learning Algorithms using GPUs

Alberto Cano Amelia Zafra Sebastián Ventura

Department of Computing and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain
{i52caroa,azafra,sventura}@uco.es

Abstract

This paper propose a multithreaded Genetic Programming classification evaluation model using NVIDIA CUDA GPUs to reduce the computational time due to the poor performance in large problems. Two different classification algorithms are benchmarked using UCI Machine Learning data sets. Experimental results compare the performance using single and multithreaded Java, C and GPU code and show the efficiency far better obtained by our proposal.

Keywords: GPUs, CUDA, Genetic Programming, Classification.

1 INTRODUCTION

Evolutionary Algorithms (EA) are good method to find a reasonable solution for data mining and knowledge discovery [1], but they can be slow at converging with complex, high dimensional problems. To solve this problem, different ways of parallelization have been studied. Specifically, if we focus on genetic programming (GP), we can find multiple ways to take advantage both of different types of parallel hardware and of different features of particular problem domains. Most of the parallel algorithms during the last two decades deal with the implementation over clusters or Massively Parallel Processing architectures (MPPs). More recently, the work about parallelization using graphics processing units (GPUs) [3], [5] and [10] provide fast parallel hardware for a fraction of the cost of a traditional parallel system.

The purpose of this paper is to analyze the scalability of NVIDIA GPUs using Compute Unified Device Architecture (CUDA) framework by the implementation

of two Genetic Programming (GP) classification models and benchmark them using UCI datasets. Experimental studies compare our proposal with C and Java single and multithreaded code proving the viability of GPUs to speedup EAs solving very large problems.

The remainder of this paper is organized as follows. Section 2 provides an overview of the GP classification algorithms to find out which parts of the algorithms are more susceptible to performance improvement. Section 3 presents information on the GPU computing architecture. Section 4 discusses the parallelization analysis and implementation. Section 5 describes the experimental models benchmarked. Section 6 presents computational experiment results. Conclusions of our investigation and future research tasks are summarized in Section 7.

2 GP CLASSIFICATION ALGORITHMS

GP is a kind of mainstream Evolutionary Algorithms paradigm that presents more variants from EAs and its goal is to develop programs automatically. GP's individuals are expressions or syntax trees that represent the structure of a program that solves the problem, in our case, the correct classification of a set of instances.

There are different proposals using the paradigm of the GP to represent sets of expressions (decision trees, discriminant functions, classification rules, etc). Classification rules are widespread employed for GP and the formalism they use to represent the classifier is IF-THEN rules [8]. The antecedent of the rule (IF) consists of a combination of conditions on the attributes predictors. The consequent (THEN) contains the predicted value for the class. Thus, a rule assigns a pattern to the class if the values of predictor attributes satisfy the conditions assessed in the antecedent.

GP algorithms have the same structure as other EAs. Deb K. [4] propose a model where an EA starts with a

set of individuals named algorithm's population. The initial population is generated randomly. For each iteration, the algorithm evaluates each individual using the fitness function. The algorithm terminates if it finds acceptable solutions or the generation count reaches a limit. Otherwise, it selects several individuals and copies them to replace individuals in the population that were not selected for reproduction so that the population size remains constant. Then, the algorithm manipulates individuals by applying different evolutionary operators such as crossover and mutation. This model initializes the population at first. The consecutive processes of selection, generation, replacement, update and control constitute an iteration of the algorithm.

The most computationally expensive step is generation since it involves the evaluation of all individuals generated. For each individual its expression must be interpreted or translated into an executable format and then it is evaluated for each training pattern. The result of an evaluation can be: true positive (t_p), false positive (f_p), true negative (t_n) or false negative (f_n). With these values the confusion matrix of the individual is constructed to get its quality index, its fitness.

The evaluation process of individuals within the population consists in two loops, where each individual iterates each pattern and checks if the rule covers that pattern. These two loops makes the algorithm really slow when the population or patterns count increases.

The experience using GP algorithms proves that almost 98% of the time is taken by generation step. The execution time of the algorithm is mainly linked to the evaluation of individuals due to the population size and the number of patterns. Thus, the most significant improvement is obtained by accelerating the evaluation phase, and this is what we do in our proposal.

3 CUDA ARCHITECTURE

The CUDA programming model consists of a host that is a traditional CPU and one or more massively data-parallel coprocessing compute devices such as a GPU.

The runtime system executes kernels as batches of parallel threads in a single-instruction, multiple-data (SIMD) programming style. These kernels comprise thousands to millions of lightweight GPU threads per each kernel invocation. To make the most of the hardware, creating enough threads is required; for example, the kernel might compute each array's element result in a separate thread.

CUDA's threads are organized into a two-level hierarchy, at the higher one all the threads in a data-parallel execution phase form a grid. Each call to a kernel

initiates a grid composed of many thread groupings, called thread blocks. All the blocks in a grid have the same number of threads, with a maximum of 512. The maximum number of thread blocks is 65535×65535 , so each device can run up to $65535 \times 65535 \times 512 = 2 \cdot 10^{12}$ threads.

To properly identify threads, each thread in a thread block has a unique ID in the form of a three-dimensional coordinate, and each block in a grid also has a unique two-dimensional coordinate.

Thread blocks are executed in streaming multiprocessors (SM). A SM can perform zero-overhead scheduling to interleave warps and hide the latency of long-latency arithmetic and memory operations.

There are three different main memory spaces: global, shared and local. GPU's memories are specialized and have different access times and output limitations.

Global memory is a large, long-latency memory that exists physically as an off-chip dynamic RAM. The threads can read and write global memory to share data and must write the kernel's output to be readable after the kernel terminates. However, a better way to share data and improve performance is to take advantage of shared memory.

Shared memory is a small, low-latency memory that exists physically as an on-chip registers and its content only maintained during thread block execution and are discarded when the thread block completes. Kernels that read or write a known range of global memory with spatial or temporal locality can employ shared memory as a software-managed cache. Such caching potentially reduces global-memory bandwidth demands and improves overall performance.

Each thread also has its own local memory space as registers, so the number of registers a thread uses determines the number of concurrent threads executed in the multiprocessor, that is called multiprocessor occupancy. To avoid wasting hundreds of cycles while a thread waits for a long-latency global-memory load or store to complete, a common technique is executing batches of global accesses, one per thread, exploiting the hardware's warp scheduling to overlap the threads's access latencies.

4 IMPLEMENTATION

To take advantage of the GPUs architecture we decided to evaluate all individuals over all the patterns simultaneously. An easy way to do that, is to create a grid of thread blocks sized as follows: one dimension is sized as the number of individuals and the other dimension is sized as the number of patterns in the data

set. This organization represents that one thread is the evaluation of one individual over one pattern.

To achieve full performance, we have to maximize the multiprocessor occupancy, so each block represents the evaluation of one individual over 512 patterns. This way, each thread within the block computes one single evaluation, then the size of the second dimension of the grid is the number of patterns divided by 512.

This configuration shown at figure 1 allows up to 65535 individuals and 33.553.920 patterns per GPU, large enough for all tested data sets. Larger populations can be evaluated using several GPUs in the system up to 4 devices and 8 GPU cores per host.

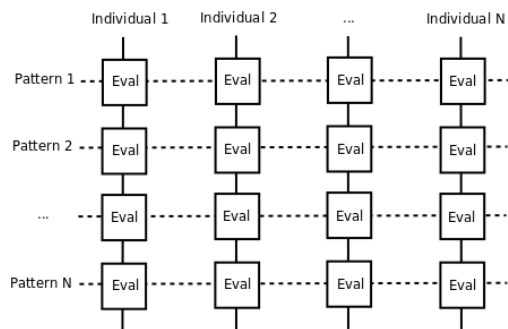


Figure 1: Parallel Evaluation Model.

Before running the evaluator on the GPU, the individuals's rules must be copied to the device memory using PCI-E bus. The CUDA programming model is an extension of the C language, so kernels are easy to write. We use Java Native Interface (JNI) to call Java natives methods and get the individuals's expression tree. JNI, Java's foreign function interface for executing native C code, is used to bridge the Java code with the kernels. In JNI, the programmer declares selected C functions as native external methods that can be invoked by a Java program. The native functions are assumed to have been separately compiled into host-specific binary code using nvcc compiler and also supports callback functions to enable native code to access Java objects.

In the following host code at figure 2, all variables preceded by *h_* are stored at host memory space and *d_* are stored at device memory space. A *base* pointer is used to know which subset of the population must be evaluated by each thread.

The evaluation takes place in two steps. In the first kernel at figure 3, threads check if the rule covers or not the pattern using a stack and stores a value for that thread depending on the classification algorithm. Generally, if the rules covers the pattern and the consequent matches the pattern's class or the rule does not

cover the pattern and the consequent does not match the pattern's class we get a success, otherwise we get a fail.

The second kernel counts the results by subsets of 512, the actual maximum number of threads per block, to get the total number of t_p , f_p , t_n and f_n and build the confusion matrix. This way, the kernel calculates in parallel the fitness of individuals using the confusion matrix and the quality metrics described for each classification model. Finally, results are copied to host memory and set to individuals for the next generation.

The kernel function must analyze the expression working with Polish notation, also known as prefix notation. Its distinguishing feature is that it places operators to the left of their operands. If the arity of the operators is fixed, the result is a syntax lacking parentheses or other brackets.

```
int threadPopulationSize = ceil(
    populationSize/numThreads);
int base = plan->device *
    threadPopulationSize;

for(int i=0; i<threadPopulationSize; i++)
    cudaMemcpy(d_exprTree[i], h_exprTree[
        base + i], sizeof(char)*strlen(
            h_exprTree[base + i]),
        cudaMemcpyHostToDevice);

dim3 threads_eval(1, 512);
dim3 grid_eval(threadPopulationSize, ceil(
    Matrix_H/512.0));

evaluate_kernel
    <<<grid_eval, threads_eval>>>
    (d_fitness, d_exprTree, Matrix_W
    , Matrix_H, classifiedClass);

cudaThreadSynchronize();

dim3 threads_cm(1, 512);
dim3 grid_cm(threadPopulationSize, 1);

confusion_matrix_kernel
    <<< grid_cm, threads_cm >>>
    (d_fitness, Matriz_H);

cudaMemcpy(h_fitness + base, d_fitness,
    threadPopulationSize*sizeof(float),
    cudaMemcpyDeviceToHost );
```

Figure 2: Host code, kernel invocation.

While there are remaining tokens, it checks what it has to do next. A stack is used to store numerical values. Finally, we check the last value of the stack. If this value is true, that means the antecedent was true, so depending on the algorithm used we compare this value to the known class given for the pattern.

```

--global-- void evaluate_kernel()
{
int row = 512*blockIdx.y + threadIdx.y;
int position = blockIdx.x*Matrix.H + row;
int sp=0,bufp=length(expr[blockIdx.x]);

while(1)
{
switch(getop(s, &bufp, expr[blockIdx.x]))
{
case NUMBER:
push(atof(s), stack, &sp);
break;
case VARIABLE:
push(vars[Matrix.W*(row) + atoi(s)],
stack, &sp);
break;
case AND:
if (pop(stack, &sp) * pop(stack, &sp))
push(1, stack, &sp);
else
push(0, stack, &sp);
break;
case OR:
if (pop(stack, &sp) || pop(stack, &sp))
push(1, stack, &sp);
else
push(0, stack, &sp);
break;
case NOT:
if (pop(stack, &sp) == 0)
push(1, stack, &sp);
else
push(0, stack, &sp);
break;
case '>':
op1 = pop(stack, &sp);
if (op1 > pop(stack, &sp))
push(1, stack, &sp);
else
push(0, stack, &sp);
break;
case '<':
op1 = pop(stack, &sp);
if (op1 < pop(stack, &sp))
push(1, stack, &sp);
else
push(0, stack, &sp);
break;
...
case END:
arg = pop(stack, &sp);
if(arg == 1) { // The antecedent is true
if(classifiedClass == knownClass[row])
result[position] = 0; // tp
else
result[position] = 2; // fp
}
else {
if(classifiedClass != knownClass[row])
result[position] = 1; // tn
else
result[position] = 3; // fn
}
return;
default: break;
}}}

```

Figure 3: GPU evaluation kernel.

This implementation allows scalability for different GPUs multiprocessors count. The more SM the device has, the faster it runs because of the parallelized blocks. Future devices will also take advantage of the programming model by the implementation of L1 core and L2 SM cache. The antecedent of the classification rule is the same for all the threads within a block so it can be cached in a single memory access.

5 EXPERIMENTATION

This paper presents an implementation of a GPU GP evaluator for data classification using JCLEC [9]. JCLEC is a software system for Evolutionary Computation (EC) research, developed in the Java programming language. It provides a high-level software environment to do any kind of Evolutionary Algorithm, with support for genetic algorithms, genetic programming and evolutionary programming.

Experiments were run on a PC equipped with an Intel Core i7 processor running at 2.66GHz with one and two NVIDIA GeForce 285 GTX video card equipped with 2GB of GDDR3 video RAM. No overclock was made to any of the hardware.

UCI Machine Learning Repository provides a repository of databases, domain theories and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. We have selected two of them for benchmarks, shuttle and poker hand inverse. The shuttle data set contains 9 attributes, 58000 instances and 7 classes. The poker hand inverse data set contains 11 attributes, 10^6 instances and 10 classes.

Two different GP classification models proposed in the literature are listed. The evaluation from each model is detailed for parallelization purpose. Full description can be obtained from their references.

Falco model

Falco, Della y Tarantino [6] propose a method to get the fitness of the rule by evaluating the antecedent over all the patterns within the data set. The adjustment function calculates the difference between the number of examples where the rule correctly predicts the membership or not of the class and number of examples where the opposite occurs, then the prediction is wrong. Finally fitness is expressed as:

$$fitness = I - (successes - fails) + \alpha * N \quad (1)$$

where I is the total number of examples from all training, α is a value between 0 and 1 and N is the number

Table 1: Execution time results

		Shuttle data set								Poker-I data set							
		Falco Model				Bojarczuk Model				Falco Model				Bojarczuk Model			
Pop		100	200	400	800	100	200	400	800	100	200	400	800	100	200	400	800
Time	Java	102	190	403	822	1832	2315	3652	6193	293	569	1093	2100	1447	2757	5177	9720
	C1	19	23	77	165	97,7	185	328	655	64	115	196	430	261	488	888	1711
	C2	10	12	38	82	51,1	94,1	165	342	33	58	99	216	136	250	446	863
	C4	5	6	20	42	28,1	48,9	89,2	184	17	30	51	111	71,4	133	230	436
	C8	5	6	19	40	27,9	49,4	90,1	177	16	28	47	80	66,4	125	213	408
	GPU	17	23	67	142	3,9	7,2	14,2	28,4	55	99	172	362	20,8	40	∞	∞
	GPU _s	10	13	35	73	2,1	3,7	7,1	14	29	52	88	183	10,5	20	37,5	∞
Speedup	Java	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	C1	5,4	8,1	5,2	5,0	18,8	12,5	11,2	9,5	4,6	5,0	5,6	4,9	5,5	5,7	5,8	4,7
	C2	10,6	15,9	10,5	10,1	35,9	24,6	22,1	18,1	9,0	9,8	11,1	9,7	10,6	11,0	11,6	11,3
	C4	19,7	30,3	20,5	19,8	65,2	47,3	40,1	33,7	16,8	18,9	21,6	18,9	20,3	20,7	22,5	22,3
	C8	19,9	30,1	21,2	20,6	65,7	46,8	40,5	34,6	18,5	20,5	23,3	26,4	21,8	22,0	24,2	23,8
	GPU	349	491	361	347	468	325	256	218	319	343	380	348	69,7	69,6	∞	∞
	GPU _s	636	901	690	672	890	625	513	442	611	662	749	688	138	136	138	∞

of nodes. The closer α to 1 is, the more importance is given to simplicity.

Bojarczuk model

Bojarczuk, Lopes y Freitas [2] proposal presents a method in which each rule is evaluated for all of the classes simultaneously for a pattern. The classifier is formed by taking the best individual for each class generated during the evolutionary process. Instead of successes and fails it measures t_p , f_p , t_n and f_n .

The fitness function used combines two indicators that are commonplace in the domain, namely the sensitivity (Se) and the specificity (Sp), defined as follows:

$$Se = \frac{t_p}{t_p + f_n} \quad Sp = \frac{t_n}{f_p + t_n} \quad (2)$$

GP does not produce simple solutions. The comprehensibility of a rule is inversely proportional to its size. Therefore Bojarczuk defines the simplicity (Sy) of a rule:

$$Sy = \frac{maxnodes - 0.5 * numnodes - 0.5}{maxnodes - 1} \quad (3)$$

where numnodes is the current number of nodes of and individual and maxnodes is the maximum allowed size of a tree. Finally, the fitness function is calculated as the product of the indicators of the predictive accuracy and simplicity:

$$fitness = Se * Sp * Sy \quad (4)$$

6 RESULTS

The results of the two GP classification algorithms benchmarked using UCI data sets are shown at table 1 and figures 4 and 5. Rows in the first half represents the generation time and rows in the second half represents the speedup compared to Java time. Each column is labeled with the algorithm execution configuration from left to right: Population size, Java single CPU thread, C single CPU thread, C two CPU threads, C four CPU threads, C eight CPU threads, 1 GPU device, 2 GPUs devices. Infinite values mean memory requirements exceeded. In Falco table, GPU time is expressed in seconds and CPU time is indicated in minutes. Bojarczuk table times are in seconds.

Benchmarks results prove the ability of GPUs to solve GP evaluation. Intel i7 quadcore performs linear scalability from 1 to 2 and 4 threads, but not any further. To go beyond, GPUs work much better. Its parallelized model allows to speed up classification problem from a month to an hour. Real classification training usually needs dozens of evaluations to get an accurate result, so the absolute time saved is a great deal of time. The highest speedup is obtained with Falco model, and it increased performance by a factor up to 700 over the Java standard CPU solution and up to 130 over the C single threaded CPU implementation.

7 CONCLUSION

Massive parallelization using NVIDIA CUDA framework provides a speedup of hundred of times over Java and C implementation. GPUs are best for thousands

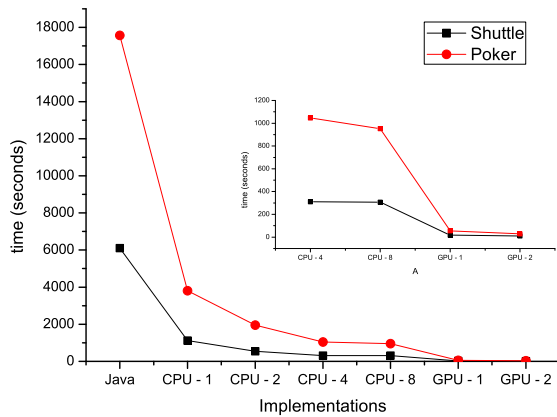


Figure 4: Falco implementation time comparison.

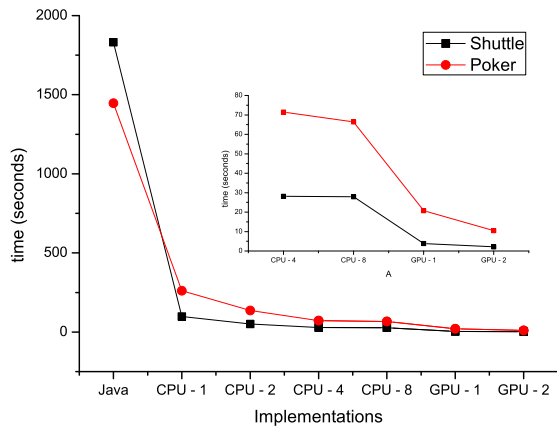


Figure 5: Bojarczuk implementation time comparison.

threads tasks where each thread does its job but all of them collaborate in the execution of the program.

CPU solution is lineal complex. However, the GPU groups the threads in a block, then a grid of blocks is executed in SMs multiprocessors, one per SM. Thus, linearity is approximately given by the number of 30-blocks grid. This implementation allows future scalability for GPUs with more cores. Next NVIDIA GPU codenamed Fermi doubles the number of cores available up to 512. Make note that i7 CPU scores are 2.5 times faster than 3.0 GHz PIV.

Further work will implement the whole algorithm inside the GPU so selection, crossover and mutation steps will be parallelized and data transfers between CPU and GPU memory are expected to minimize.

ACKNOWLEDGMENTS

This work has been financed in part by the TIN2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (CICYT), the P08-TIC-3720 project of the Andalusian Science and Technology Department and FEDER funds.

References

- [1] A. A. Freitas, "Data Mining and Knowledge Discovery with Evolutionary Algorithms", Springer-Verlag, 2002.
- [2] C.C. Bojarczuk, H.S. Lopes, A.A. Freitas, and E.L. Michalkiewicz, "A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets", *Artificial Intelligence in Medicine*, vol. 30, no. 1, pp. 27-48, January 2004.
- [3] Chitty, D., "A data parallel approach to genetic programming using programmable graphics hardware", *GECCO'07: Proceedings of the Conference on Genetic and Evolutionary Computing*, pp. 1566 - 1573, 2007.
- [4] Deb K, "A population-based algorithm-generator for real-parameter optimization", *Soft Computing* 9(4):236253, 2005.
- [5] Harding, S., Banzhaf, W., "Fast genetic programming and artificial developmental systems on gpus", *HPCS'07: Proceedings of the Conference on High Performance Computing and Simulation*, pp. 2 - 2, 2007.
- [6] I. De Falco, A. Della Cioppa, and E. Tarantino, "Discovering interesting classification rules with genetic programming", *Applied Soft Computing Journal*, vol. 1, no.4, pp. 257-269, 2002.
- [7] M.G. Arenas, J.G. Castellano, J. Carpio, P.A. Castillo, M. Cillero, J.J. Merelo, A. Prieto, V.Rivas, G. Romero. "Speedup measurement for a distributed evolutionary algorithm that uses Jini", *In XI Jornadas de Paralelismo*, pp. 247 -252, ISBN:84-699-3003-6, Granada, Spain, September, 2000.
- [8] P.G. Espejo, S. Ventura and F. Herrera, "A Survey on the Application of Genetic Programming to Classification", *IEEE Transactions on Systems, Man and Cybernetics: Part C*, (accepted).
- [9] S.Ventura, C. Romero, A. Zafra, J.A. Delgado and C. Hervás. "JCLEC: a Java framework for evolutionary computation", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 2007.
- [10] William B. Langdon, Andrew P. Harrison, "GP on SPMD parallel graphics hardware for mega Bioinformatics data mining", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 12:1169-1183, 2008.

Determinando Automáticamente los Dominios de Competencia de un Sistema de Clasificación Basado en Reglas Difusas: Un Caso de Estudio con FH-GBML

Julián Luengo¹ Francisco Herrera¹

¹ Departamento Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR, Granada, 18071, España, {julianlm,herrera}@decsai.ugr.es

Resumen

El objetivo de esta contribución es proponer un sistema automático para determinar los dominios de competencia de un Sistema de Clasificación Basado en Reglas Difusas mediante el uso de medidas de complejidad de los datos. Para ello empleamos como caso de estudio el método Fuzzy Hybrid Genetic Based Machine Learning, y examinamos diversas métricas de complejidad de datos sobre un amplio rango de bases de datos extrayendo patrones de comportamiento de los resultados de forma automática. A continuación obtenemos reglas de estos patrones que describen el buen y el mal comportamiento del método FH-GBML.

Gracias a estas reglas es posible predecir el comportamiento del método a partir de los valores de las medidas de complejidad de la base de datos antes de aplicarlo y por tanto establecer sus dominios de competencia.

Keywords: Clasificación, Complejidad de Datos, Sistemas Basados en Reglas Difusas, Sistemas Difusos Genéticos.

1. Introducción

Los Sistemas de Clasificación Basados en Reglas Difusas (SCBRDs) [6, 9] son una herramienta muy útil en el ámbito de Machine Learning dado que son capaces de construir un modelo lingüístico interpretable por seres humanos. Existe una extensa literatura en el campo de los SCBRDs [9], la cual se encuentra muy activa actualmente.

Las capacidades de predicción de los clasificadores dependen enormemente de las características del proble-

ma. Un campo emergente ha aparecido recientemente, el cual usa un conjunto de medidas de complejidad aplicadas al problema para describir su dificultad. Estas medidas cuantifican aspectos particulares del problema que se consideran difíciles para la tarea de la clasificación [2]. Pueden encontrarse estudios de medidas de complejidad aplicadas a algoritmos de clasificación particulares en [2, 4, 3, 11].

La complejidad en los datos puede usarse para caracterizar el rendimiento del SCBRD, y puede considerarse como una nueva rama en el uso de los SCBRDs en Reconocimiento de Patrones. En [10] realizamos una caracterización del método Fuzzy Hybrid Genetic Based Machine Learning (FH-GBML) propuesto por Ishibuchi et al. [8] usando esta metodología.

En el análisis de [10], las relaciones entre las medidas de complejidad de datos y el método FH-GBML eran extraídas ad-hoc por nosotros. En este trabajo estamos interesados en analizar de nuevo la relación entre el método FH-GBML y las mismas medidas de complejidad que en nuestro estudio anterior de forma automática y parametrizada, de forma que no exista un sesgo humano. Con la caracterización de los dominios de competencia de forma automática es posible determinar con cierta confianza cuando el método FH-GBML tendrá buen o mal comportamiento.

Para realizar nuestro estudio, hemos creado 438 bases de datos binarias a partir de problemas reales. Hemos calculado el valor de las 12 métricas de complejidad de datos propuestas por Ho y Basu [5] sobre las bases de datos completas. Hemos evaluado el método FH-GBML sobre ellas con el esquema 10-fcv obteniendo su precisión en entrenamiento y en test. A continuación hemos empleado el método automático propuesto para extraer intervalos de valores de las medidas de complejidad de datos en los cuales el método FH-GBML tiene buen o mal comportamiento. A partir de estos intervalos, obtenemos una serie de reglas que muestran cuantitativamente los diferentes comportamientos del

método FH-GBML.

El resto de la contribución está organizada como sigue. En la Sección 2 describimos el SCBRD que hemos usado, e introducimos las medidas de complejidad consideradas. A continuación, en la Sección 3 desarrollamos la descripción del método automático. En la Sección 4 mostramos el marco experimental y los resultados obtenidos a partir del método automático, junto a las reglas extraídas y su análisis. Finalmente, en la Sección 5 señalamos nuestras conclusiones.

2. Preliminares

En esta sección previa introducimos el SCBRD empleado en nuestro análisis en la Subsección 2.1, y las medidas de complejidad de datos en la Subsección 2.2.

2.1. Algoritmo FH-GBML

El método FH-GBML es un método Pittsburgh en el cual cada conjunto de reglas se maneja como un individuo. Además, contiene un enfoque de aprendizaje Cooperativo-Competitivo (un individuo representa una única regla), el cual se usa como un tipo de mutación heurística para modificar parcialmente cada conjunto de reglas.

Este método usa reglas difusas estándar con pesos [7] en el cual cada variable de entrada x_i está representada por un término lingüístico o etiqueta. El sistema define 14 posibles términos lingüísticos para cada atributo, además de un conjunto “don’t care” especial.

En el proceso de aprendizaje, N_{pop} conjuntos de reglas son creados seleccionando aleatoriamente N_{rule} ejemplos de entrenamiento. Entonces, se genera una regla difusa a partir de cada patrón de entrenamiento eligiendo probabilísticamente un conjunto difuso como antecedente a partir de los 14 candidatos ($P(B_k) = \frac{\mu_{B_k}(x_{pi})}{\sum_{j=1}^{14} \mu_{B_j}(x_{pi})}$); y cada antecedente (conjunto difuso) de la regla difusa generada se reemplaza con *don’t care* usando una probabilidad pre-fijada $P_{don't\ care}$.

$N_{pop} - 1$ conjuntos de reglas se generan por selección, cruce y mutación de la misma manera que los algoritmos de tipo Pittsburgh. A continuación, con una probabilidad pre-fijada, una única iteración del algoritmo Cooperativo-Competitivo se aplica a cada conjunto de reglas generado.

Finalmente, el mejor conjunto de reglas se añade, en la población actual, a los nuevos conjuntos de reglas generados ($N_{pop} - 1$) para formar la siguiente población y, si el criterio de parada no es alcanzado, el proceso genético se repite de nuevo. La clasificación de los ejemplos se realiza mediante método de razonamiento

difuso de la regla ganadora.

En nuestro estudio, se han empleado los parámetros que empleamos en [10].

2.2. Medidas de Complejidad de Datos

En nuestro análisis hemos usado las 12 medidas de complejidad propuestas por Ho y Basu [5] que cuantifican una serie de características de las bases de datos que implican alguna dificultad para la tarea de clasificación. En la Tabla 1 se encuentran enumeradas dichas medidas de complejidad. Una descripción más completa de las mismas puede encontrarse en [2].

Cuadro 1: Medidas de complejidad usadas

Tipo	Identif.	Nombre
Medidas de solapamiento de atributos	F1	Razón discriminante de Fisher
	F2	Volumen de la región de solapamiento
	F3	Máxima eficiencia individual de los atributos
Medidas de separabilidad de las clases	L1	Mínimización de la suma de error distancia por Programación Lineal
	L2	Error del clasificador lineal por Programación Lineal
	N1	Fracción de puntos en los bordes de las clases
	N2	Media de la distancia de Vecinos Más Cercanos intra/inter-clases
	N3	Error del clasificador 1-NN
Medidas de geometría, topología y densidad	L3	No-linealidad del clasificador lineal por Programación Lineal
	N4	No-linealidad del clasificador 1-NN
	T1	Fracción de puntos con subconjuntos adheridos
	T2	Media de puntos por dimensión

3. Descripción del Método Automático

En el trabajo realizado en [10], el proceso de extracción de los intervalos en los cuales se detectaba un buen o mal comportamiento del método FH-GBML era manual. La extracción ad-hoc de los intervalos presenta dos problemas principalmente:

- Los puntos de corte son elegidos arbitrariamente.
- Es posible obviar intervalos con características equivalentes a otros que se han escogido.

Nuestra solución a estos problemas consiste en definir un método automático que es capaz de encontrar los intervalos en los que el método muestra un comportamiento destacado. Mediante esta definición, el método automático decide qué medidas de complejidad son útiles al contener intervalos con estas características, y qué medidas son descartadas (sin dar ningún intervalo para ellas). A continuación, se muestra el esquema del método automático en el Algoritmo 1.

Donde las funciones contenidas en el algoritmo tienen el siguiente comportamiento.

Algoritmo 1 Método automático

Entrada: Conjunto de resultados del clasificador $U = \{u_1, u_2, \dots, u_n\}$ para una serie de bases de datos ordenadas por una medida de complejidad.

Salida: Un conjunto A de intervalos en los que el clasificador muestra buen o mal comportamiento.

Pasos:

$i \leftarrow 1$

$A \leftarrow \{\}$

mientras $i < n$ **hacer**

$pos \leftarrow siguientePuntoDestacado(i, U)$

si $pos \neq -1$ **entonces**

$intervalo \leftarrow extenderIntervalo(pos, U)$

$A \leftarrow A \cup \{intervalo\}$

$i \leftarrow limiteSuperior(intervalo)$

fin si

fin mientras

$A \leftarrow unirIntervalosSolapados(A)$

devolver A

■ *siguientePuntoDestacado*(i, U): Busca el siguiente punto de buen o mal comportamiento en el conjunto U a partir del valor u_i en adelante, incrementando $j = 1, \dots, n - i$.

- Un punto u_{i+j} de buen comportamiento es aquel en que (1) su diferencia de precisión en entrenamiento y test no es superior a 5; y (2) una precisión en entrenamiento superior a 90.
- Un punto u_{i+j} de mal comportamiento es aquel en que (1) su diferencia de precisión en entrenamiento y test es superior a 5; y (2) su precisión en entrenamiento es inferior en 10 puntos a la media de entrenamiento global.

■ *extenderIntervalo*(pos, U): A partir del punto u_{pos} amplía los límites del intervalo (u_{pos-i}, u_{pos+i}) mientras el intervalo siga cumpliendo los criterios correspondientes e $pos - i \geq 0$ y $pos + i \leq n$.

- Un intervalo (u_{pos-i}, u_{pos+i}) de buen comportamiento es aquel en que (1) la precisión media de los puntos contenidos en entrenamiento supere la precisión media en entrenamiento global en 3 puntos; y (2) la precisión en test media de los puntos contenidos supere la precisión media en test global en 6 puntos.
- Un intervalo (u_{pos-i}, u_{pos+i}) de mal comportamiento es aquel en que o bien (1) la precisión media de los puntos contenidos en entrenamiento es inferior a la precisión media en entrenamiento global en 10 puntos; o bien (2) la diferencia de la media de entrenamiento y la media de test del intervalo es inferior en 5 puntos a la diferencia global; o bien (3)

la precisión media en test del intervalo es inferior en 6 puntos a la media en test global.

- *unirIntervalosSolapados*(A): Esta función simplemente comprueba los límites del conjunto de intervalos. Elimina aquellos intervalos completamente cubiertos por otros. Además intenta combinar intervalos con extremos solapados o separados como mucho por 5 puntos si cumplen los criterios de buen o mal comportamiento mencionados.

Es necesario indicar que las definiciones de puntos e intervalos de buen y mal comportamiento están parametrizadas, y pueden ser ajustadas por el usuario. En esta contribución dichas definiciones son análogas a las observadas en las reglas ad-hoc de [10], de manera que es posible realizar una comparación directa.

4. Estudio Experimental: Análisis del Método FH-GBML con el Método Automático

En esta sección mostramos los resultados de la experimentación con el método FH-GBML, y analizamos dichos resultados. En primer lugar, en la Subsección 4.1 se indica el marco experimental con las bases de datos empleadas y el esquema de validación. En la Subsección 4.2 mostramos los intervalos obtenidos con el método automático y formulamos las reglas derivadas de los mismos. Finalmente, en la Subsección 4.3 se analiza la combinación de las reglas simples para determinar los dominios de competencia del método.

4.1. Marco Experimental: Generación de las Bases de Datos

Hemos evaluado el método FH-GBML sobre un conjunto de 438 problemas de clasificación binarios. Las 12 medidas de complejidad se han calculado sobre cada uno de las bases de datos completas. Estas bases de datos son generadas a partir de la combinación por parejas de las clases de 20 problemas del repositorio de la Universidad de California, Irvine (UCI) [1]. En particular son *iris*, *wine*, *new-thyroid*, *solar-flare*, *led7digit*, *zoo*, *yeast*, *tae*, *balanced*, *car*, *contraceptive*, *ecoli*, *hayes-roth*, *shuttle*, *australian*, *pima*, *monks*, *bu-pa*, *glass*, *haberman* y *vehicle*.

Para construir las bases de datos, a partir de cada base de datos original (multiclase) hemos extraído los ejemplos pertenecientes a cada clase. A continuación se crean nuevas bases de datos con los ejemplos pertenecientes a 2 clases diferentes, para cada posible emparejamiento. Si una de las bases de datos obtenidas resulta ser separable linealmente (la medida de complejidad L1 vale 0), es descartada.

Cuadro 4: Reglas simples derivadas a partir de los intervalos

Reglas Buen Comportamiento						
Id.	Rango	% Soporte	% Entrenamiento	Dif. Entrenamiento	% Test	Dif. Test
R1+	If N1 $\in$ [0.00117,0.04865] then <i>buen comportamiento</i>	17.352	99.768	6.199	98.222	9.981
R2+	If N2 $\in$ [0.00883,0.2373] then <i>buen comportamiento</i>	26.941	99.195	5.625	96.654	8.412
R3+	If L1 $\in$ [0.03021,0.2201] then <i>buen comportamiento</i>	22.603	98.764	5.195	95.754	7.512
R4+	If L2 $\in$ [0.0,0.1232] then <i>buen comportamiento</i>	40.411	98.014	4.444	94.890	6.648
Reglas Mal Comportamiento						
Id.	Rango	% Soporte	% Entrenamiento	Dif. Entrenamiento	% Test	Dif. Test
R1-	If F1 $\in$ [0.03355,0.8579] then <i>mal comportamiento</i>	24.201	89.202	-4.367	82.182	-6.060
R2-	If N1 $\in$ [0.2963,1.0] then <i>mal comportamiento</i>	21.005	83.645	-9.924	74.959	-13.283
R3-	If N2 $\in$ [0.5853,1.049] then <i>mal comportamiento</i>	15.982	82.182	-11.387	73.561	-14.681
R4-	If N3 $\in$ [0.1636,0.5426] then <i>mal comportamiento</i>	21.005	83.531	-10.038	74.521	-13.721
R5-	If N4 $\in$ [0.1898,0.4868] then <i>mal comportamiento</i>	21.689	87.069	-6.501	80.847	-7.394
R6-	If L2 $\in$ [0.247,0.5556] then <i>mal comportamiento</i>	30.137	87.911	-5.658	79.299	-8.943
R7-	If T1 $\in$ [0.9635,1.0] then <i>mal comportamiento</i>	27.397	89.572	-3.998	82.201	-6.041
R8-	If T2 $\in$ [0.5625,14.0] then <i>mal comportamiento</i>	23.973	89.878	-3.692	82.235	-6.007

Para poder obtener más bases de datos binarias, también hemos agrupado las clases de dos en dos, esto es, creamos una base de datos binaria y cada una de sus dos clases son la combinación de dos clases de la base de datos original. Para este segundo procedimiento hemos usado las bases de datos *ecoli*, *glass* y *flare*.

Para poder medir el porcentaje de acierto o precisión del método FH-GBML, hemos aplicado un esquema de validación 10-fcv. En la Tabla 2 mostramos la media de precisión global en Entrenamiento y Test obtenidos por el método FH-GBML.

Cuadro 2: Precisión media global de FH-GBML en entrenamiento y test

FH-GBML % precisión media global en entrenamiento	93.570 %
FH-GBML % precisión media global en test	88.242 %

4.2. Formulación de las reglas a partir de los intervalos

En la Tabla 3 resumimos los intervalos de buen y mal comportamiento obtenidos con el método automático para el método FH-GBML para las medidas de complejidad que ha seleccionado.

Una vez definidos los intervalos y sus puntos de corte de la Tabla 3 es posible obtener una serie de reglas a partir de ellos, considerando los valores de las medidas. Las reglas derivadas están resumidas en la Tabla 4. Las columnas que componen la Tabla 4 son las siguientes:

Cuadro 3: Intervalos obtenidos por el método automático

Buen Comportamiento		Mal Comportamiento	
Medida	Rango	Medida	Rango
N1	[0.00117,0.04865]	F1	[0.03355,0.8579]
N2	[0.00883,0.2373]	N1	[0.2963,1.0]
L1	[0.03021,0.2201]	N2	[0.5853,1.049]
L2	[0.0,0.1232]	N3	[0.1636,0.5426]
		N4	[0.1898,0.4868]
		L2	[0.247,0.5556]
		T1	[0.9635,1.0]
		T2	[0.5625,14.0]

- La primera columna corresponde al identificador de la regla.
- La columna Rango indica el dominio de la regla.
- La columna Soporte indica el porcentaje de bases de datos cubiertas del total.
- La columna % Entrenamiento indica el porcentaje medio de precisión en entrenamiento de FH-GBML en las bases de datos cubiertas por la regla.
- La columna Dif. Entrenamiento muestra la diferencia entre el % Entrenamiento y el porcentaje medio de entrenamiento global de FH-GBML.
- La columna % Test indica el porcentaje medio de precisión en test de FH-GBML en las bases de datos cubiertas por la regla.
- La columna Dif. Test muestra la diferencia entre el % Test y el porcentaje medio de test global de FH-GBML.

Las reglas de buen comportamiento (con un símbolo “+” en el identificador) siempre muestran una diferencia positiva respecto a la media en entrenamiento y

Cuadro 5: Reglas obtenidas por la conjunción e intersección de las reglas simples

Id.	Regla	Soporte	% Entrenamiento	Dif. Entrenamiento	% Test	Dif. Test
URP	If R1+ or R2+ or R3+ or R4+ then buen comportamiento	42.237	98.082	4.512	94.958	6.716
URN	If R1- or R2- or R3- or R4- or R5- or R6- or R7- or R8- then mal comportamiento	66.894	91.063	-2.507	84.715	-3.527
$URP \wedge URN$	If URP and URN then buen comportamiento	22.602	97.204	3.634	93.399	5.157
$URP \wedge \neg URN$	If URP and not URN then buen comportamiento	19.634	99.093	5.523	96.752	8.510
$URN \wedge \neg URP$	If URN and not URP then mal comportamiento	44.292	87.929	-5.641	80.284	-7.958
no caracterizados	If not URP and not $(URN \wedge \neg URP)$ then buen comportamiento	13.470	97.965	4.395	93.349	5.107

test global. Las reglas de mal comportamiento (con un símbolo “-” en su identificador) verifican siempre el caso opuesto. El soporte de las reglas muestra que podemos caracterizar un amplio rango de bases de datos y obtener diferencias significativas en precisión.

4.3. Evaluación Conjunta del Conjunto de Reglas

El objetivo de esta sección es analizar la combinación de las reglas de buen comportamiento, y la combinación de las reglas de mal comportamiento. Para ello, realizamos la disyunción de las reglas positivas para obtener una nueva regla (Unión de las Reglas Positivas -URP-), y la disyunción de las reglas negativas para obtener otra (Unión de las Reglas Negativas -URN-). Además de las uniones, consideramos también la intersección y diferencia de éstas: la intersección de URP y URN ($URP \wedge URN$); la diferencia de URP menos URN ($URP \wedge \neg URN$); y la diferencia de URN menos URP ($URN \wedge \neg URP$). En la Tabla 5 se encuentran resumidas las reglas mencionadas, además de una última regla que representa aquellas bases de datos no cubiertas por ninguna de estas reglas.

Analizando las reglas combinadas, podemos observar que el porcentaje de soporte se ha incrementado respecto a las reglas individuales, mientras que la diferencia respecto a la precisión global se mantiene en cada caso (positiva y negativa). Con las reglas URP y $URN \wedge \neg URP$ podemos considerar tres bloques disjuntos de bases de datos con su respectivo soporte, tal y como se ha representado en la Figura 1 (sin orden particular de las bases de datos en cada bloque):

- El primer bloque, a la izquierda, representa las bases de datos cubiertos por la regla URP. Son aquellas bases de datos reconocidas como aquellas en las que FH-GBML presenta un buen comportamiento.
- El segundo bloque, en el centro, contiene las bases de datos para la regla $URN \wedge \neg URP$, las cuales son

malas para el método FH-GBML.

- El tercer bloque, a la derecha, contiene las bases de datos no clasificadas por ninguna de las dos reglas anteriores.

Podemos comprobar que se ha caracterizado el 88.53 % de las bases de datos originales, de forma que aquellas bases de datos para las que el método FH-GBML presenta buen o mal comportamiento han sido caracterizadas en su mayoría. Debemos mencionar que en el estudio con la extracción manual ad-hoc de los intervalos [10], la misma representación de 3 bloques ad-hoc caracterizaba el 75 % de las bases de datos, un 13 % menos. Por tanto, el método automático nos ha permitido realizar una caracterización más exhaustiva.

5. Conclusiones

Hemos realizado un estudio sobre un conjunto de bases de datos binarias con el método FH-GBML y calculado una serie de medidas de complejidad para las bases de datos de manera que se han obtenido un conjunto de intervalos de valores de dichas medidas. A diferencia de estudios anteriores, este proceso de extracción se ha realizado con un método automático que nos permite definir las características de estos intervalos. En estos intervalos extraídos el método presenta una precisión significativamente buena o mala. Además, hemos construido una serie de reglas descriptivas, y estudiado la interacción entre las reglas.

Mediante la combinación de las reglas, hemos obtenido finalmente dos reglas simples y precisas para describir tanto el buen como el mal comportamiento del método FH-GBML, que gracias al método automático caracterizan un mayor número de bases de datos. De esta manera, presentamos la posibilidad de determinar en qué bases de datos el método FH-GBML funcionaría bien o mal antes de ejecutarlo, usando las medidas de complejidad de datos.

Debemos indicar que se trata de un estudio particular

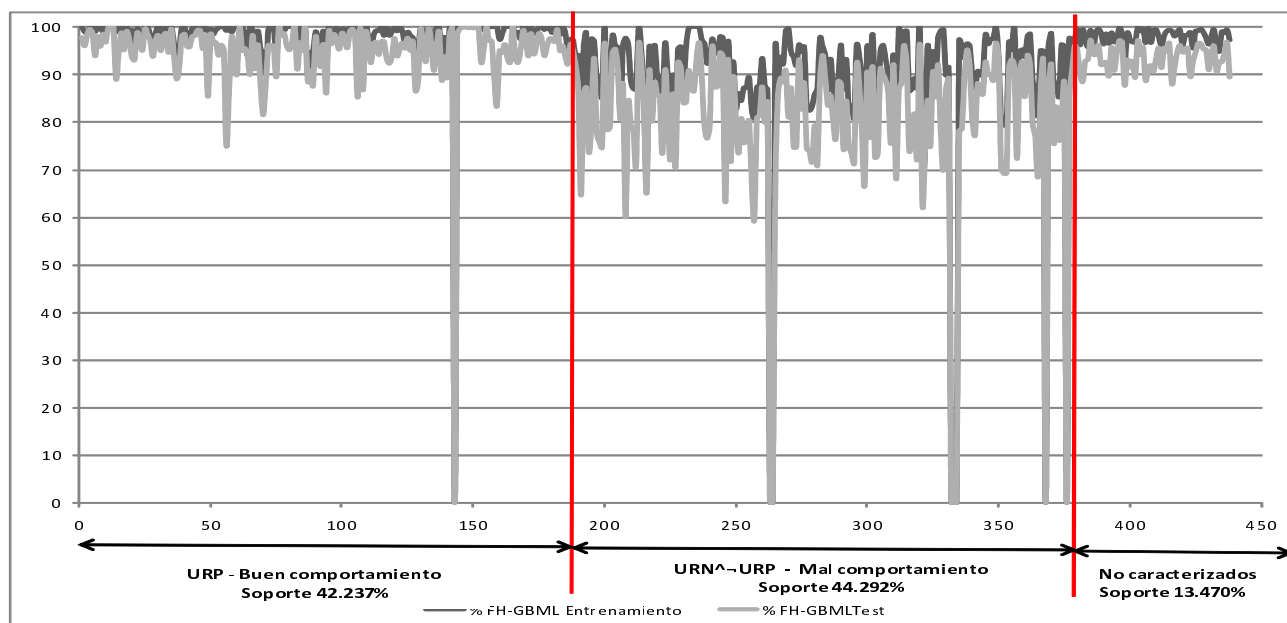


Figura 1: Representación de 3 bloques de la caracterización

para un método específico, el FH-GBML. Como trabajo futuro, y gracias a la automatización de la extracción de los intervalos, es posible extender este estudio a otros métodos no considerados aún para obtener sus dominios de competencia y analizar las relaciones entre diferentes métodos.

Agradecimientos

Este trabajo de investigación ha sido posible gracias a la subvención del proyecto del Ministerio de Ciencia e Innovación TIN2008-06681-C06-01.

Referencias

- [1] A. Asuncion, D. Newman, UCI machine learning repository (2007).
URL <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [2] M. Basu, T. K. Ho, Data Complexity in Pattern Recognition (Advanced Information and Knowledge Processing), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [3] R. Baumgartner, R. L. Somorjai, Data complexity assessment in undersampled classification of high-dimensional biomedical data, Pattern Recognition Letters 12 (2006) 1383–1389.
- [4] E. Bernadó-Mansilla, T. K. Ho, Domain of competence of xcs classifier system in complexity measurement space, IEEE Trans. Evolutionary Computation 9 (1) (2005) 82–104.
- [5] T. K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 289–300.
- [6] H. Ishibuchi, T. Nakashima, M. Nii, Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining, Springer-Verlag, 2004.
- [7] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 13 (2005) 428–435.
- [8] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy gbml approaches for pattern classification problems., IEEE Transactions on Systems, Man, and Cybernetics, Part B 35 (2) (2005) 359–365.
- [9] L. Kuncheva, Fuzzy classifier design, Springer-Verlag, 2000.
- [10] J. Luengo, F. Herrera, Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method, Fuzzy Sets and Systems Artículo Aceptado.
- [11] J. Sánchez, R. Mollineda, J. Sotoca, An analysis of how training data complexity affects the nearest neighbor classifiers, Pattern Anal. Appl. 10 (3) (2007) 189–201.

Miner a de reglas de asociacion con programacion genetica gramatical

José María Luna Juan Luis Olmo José Raúl Romero Sebastián Ventura

Departamento de Informática y Análisis Numérico, Universidad de Córdoba
{i32luarj, juanluisolmo, jrromero, sventura}@uco.es

Resumen

En este trabajo presentamos un algoritmo de programación genética gramatical G3P (*Grammar Guided Genetic Programming*) para la extracción de reglas de asociación sobre conjuntos de datos. Para ello, proponemos dos versiones para la extracción de reglas de asociación: la primera de ellas es un algoritmo generacional simple, mientras que la segunda hace uso de una población auxiliar en la que se insertan individuos que sobrepasen una determinada calidad. Además, comparamos nuestra propuesta con el algoritmo *Apriori* mostrando cómo nuestro algoritmo obtiene unos resultados medios mejores con menor número de reglas.

Palabras Clave: Algoritmos Evolutivos, Programación Genética Gramatical, Minería de datos.

1 INTRODUCCION

Las técnicas de minería de patrones están diseñadas para extraer datos de interés y útiles de las bases de datos. La mayoría de estas técnicas tienen su base en los conjuntos de patrones frecuentes, si bien en ocasiones resulta interesante encontrar asociaciones excepcionales mediante el análisis de los conjuntos de patrones no frecuentes [1]. Este análisis cobra especial relevancia en sectores como la medicina o la biología, así como en ámbitos comerciales, de finanzas o en transacciones económicas a través de la web. Sin embargo, para la mayoría de dominios, los patrones que aparecen con frecuencia son los más útiles y, por tanto, objeto de estudio.

El objetivo de la minería de asociación es la obtención de *reglas útiles* (o de interés) desde las que sea posible derivar nuevo conocimiento. Así pues, el papel de los

sistemas de minería de asociación en este proceso es facilitar el descubrimiento de patrones y permitir la presentación de estas reglas o inferencias para su posterior interpretación por el usuario y así determinar su utilidad.

Uno de los principales algoritmos basados en patrones frecuentes de bases de datos fue el propuesto por *R. Agrawal and R. Srikant* [2] en 1994 y que ha servido como punto de partida de muchas investigaciones [3, 4, 12]. Este algoritmo, denominado *Apriori*, fue el primero utilizado satisfactoriamente para la extracción de reglas de asociación. En él, se extraen los conjuntos de elementos frecuentes y, a partir de estos, se obtiene reglas de asociación. Para reducir el coste computacional, el algoritmo *Apriori* mantiene que si un conjunto de elementos es frecuente, entonces todos sus subconjuntos son frecuentes. Si un conjunto no es frecuente, sus superconjuntos no serán frecuentes, por lo que se puede reducir el coste computacional eliminando dichas producciones. Con posterioridad al algoritmo *Apriori*, se presentaron distintas alternativas para la extracción de patrones frecuentes, como *Eclat* [14], *FP-Growth* [8] y *TM* [11].

Existen investigaciones que proponen algoritmos evolutivos [5] para la extracción de reglas de asociación de datos cuantitativos [9, 10], considerándose este tipo de algoritmos y, particularmente, los algoritmos genéticos [6], como una de las técnicas de búsqueda de mayor éxito para los problemas complejos, demostrando ser una técnica importante para el aprendizaje y la extracción de conocimiento. Los algoritmos genéticos son métodos de búsqueda robustos y flexibles, pues un mismo algoritmo genético puede ser usado con diferentes representaciones. Además, los algoritmos genéticos, permiten obtener soluciones viables dentro de unos límites de tiempo. Es por esto que, tanto los algoritmos evolutivos como los algoritmos genéticos, han experimentado un creciente interés en el campo de la minería de datos.

Regla = Antecedente, Consecuente ;
 Antecedente = Comparación | “Y”, Comparación, Antecedente ;
 Consecuente = Comparación ;
 Comparación = Comparador Categórico, Atributo Categórico ;
 Comparador Categórico = “!=” | “=” ;
 Atributo Categórico = “nombre”, “valor” ;

Figura 1: Gramática de contexto libre.

En el presente trabajo proponemos un algoritmo de programación genética gramatical G3P (*Grammar Guided Genetic Programming*) para la extracción de reglas de asociación. G3P es una extensión de la programación genética que permite la obtención de programas (en este caso, reglas) válidos al definir los individuos utilizando una gramática de contexto libre, con la que estableceremos formalmente las restricciones sintácticas del problema a resolver. Por tanto, cada individuo generado por G3P es un árbol de derivación que genera y representa una solución perteneciente al lenguaje definido por la gramática. Proponemos dos versiones del algoritmo G3P: la primera de ellas es un algoritmo generacional simple, mientras que la segunda hace uso de una población auxiliar en la que se guardan individuos que sobrepasan una determinada calidad. Además, realizamos una primera comparación entre las versiones de nuestro algoritmo para determinar la que mejores resultados nos proporcione, para luego comparar nuestra propuesta con *Apriori*. Los resultados mostrarán que nuestro algoritmo obtiene menor número de reglas de asociación que *Apriori*, lo que implica una mayor eficiencia, y unos resultados medios mejores que *Apriori*.

El presente artículo está organizado como sigue: la Sección 2 describe el modelo desarrollado y sus principales características. En la Sección 3 se describen las bases de datos utilizadas para realizar la experimentación. Posteriormente, en la Sección 4 se describe tanto la ejecución como los resultados obtenidos. Finalmente, mostraremos las conclusiones alcanzadas y comentaremos las líneas de trabajo futuro que pretendemos abordar.

2 DESCRIPCION DEL MODELO

En esta sección presentaremos nuestro modelo, así como sus características más relevantes: cómo se representan los individuos, los operadores genéticos utilizados, así como los evaluadores y mecanismos de control de la población auxiliar.

2.1 REPRESENTACIÓN DE LOS INDIVIDUOS

Cada individuo está formado por dos componentes diferenciados: (a) un *genotipo*, que se codifica mediante G3P con una estructura de árbol con profundidad limitada para evitar las infinitas derivaciones posibles, y (b) un *fenotipo*, que representa la regla completa formada por un antecedente y un consecuente. El antecedente de cada regla se integra por una serie de condiciones relativas a los valores de ciertos atributos que han de ser todas satisfechas. Por el contrario, el consecuente está compuesto por una sola condición. La Figura 1 muestra la gramática de contexto libre que representa las reglas codificadas por los individuos de la población. Hay que indicar que, el símbolo no terminal ‘nombre’ de la gramática vendrá determinado por los atributos del conjunto de datos utilizado en cada momento. Además, para cada atributo de la gramática, se asignará un valor determinado por el rango de valores del mismo atributo en el *dataset*.

2.2 OPERADORES GENÉTICOS

Los nuevos individuos de cada generación se obtienen por medio de dos operadores:

- *Cruce*. Realiza un intercambio de los subárboles de derivación de dos padres a partir de dos puntos compatibles seleccionados aleatoriamente en cada uno de ellos. Dos nodos son compatibles si pertenecen al mismo símbolo no terminal, evitando así producir un individuo erróneo según la gramática definida.
- *Mutación*. Selecciona aleatoriamente un nodo del árbol y, según el tipo de nodo, se actúa. Si el nodo seleccionado es un elemento terminal, se cambia el valor de dicho elemento terminal de forma aleatoria. Si, por el contrario, el nodo seleccionado es un elemento no terminal, se realiza una nueva derivación a partir de dicho nodo. Hay que tener en cuenta que si el elemento seleccionado es un elemento no terminal, y debido a que se podría realizar una derivación diferente a la realizada en una primera instancia, el número de derivaciones necesarias para llegar a un símbolo termi-

nal podría variar, por lo que hay que controlar que no se sobrepase el tamaño máximo de derivación impuesto en el algoritmo.

2.3 EVALUACIÓN

Antes de llevar a cabo la evaluación de un individuo, hemos de llevar a cabo su decodificación, es decir, obtener la regla de asociación que se corresponde con el árbol sintáctico de su genotipo.

El proceso de decodificación consiste en construir una expresión recorriendo el árbol sintáctico en profundidad, y eliminando los símbolos no terminales que aparecen en el mismo. Asimismo, se comprueba que los individuos no utilicen el mismo atributo en el antecedente y consecuente de la regla.

El proceso de evaluación de los individuos se realiza obteniendo el valor de su función de evaluación. Éste será el *soporte*, que se define como la proporción de transacciones en una base de datos D que contiene el conjunto de elementos X :

$$fitness(individuo) = sop(X) = \frac{|X|}{|D|} \quad (1)$$

Otra de las heurísticas que vamos a utilizar es la *confianza* de la regla obtenida mediante el proceso de decodificación. Ésta se define como un estimador de $P(Y/X)$, la probabilidad de encontrar la parte derecha de una regla condicionada a que se encuentre también la parte izquierda:

$$conf(X \Rightarrow Y) = \frac{sop(X \cup Y)}{sop(X)} \quad (2)$$

Si el *fenotipo* del individuo es correcto, se procederá a su evaluación respecto al conjunto de instancias del *dataset*. Dicha evaluación consiste en el cálculo del *soporte*, que se realiza del siguiente modo: para cada atributo del antecedente, se compara con su homónimo de la instancia y se realiza un Y lógico con el valor lógico *verdadero* si es correcto y con el valor lógico *falso* si no lo es. Con el atributo del consecuente actuamos de forma análoga incrementando una variable si el antecedente es correcto y otra si tanto el antecedente como el consecuente son correctos.

2.4 ALGORITMO EVOLUTIVO

En esta sección describiremos las dos versiones del algoritmo evolutivo que hemos desarrollado para resolver el problema de asociación.

2.4.1 Primera versión: algoritmo generacional

Esta primera solución comienza obteniendo la población mediante la generación aleatoria de individuos a partir de la gramática de contexto libre definida en la sección 2.1 y cumpliendo el número máximo de derivaciones posibles. Para llevar a cabo la derivación de los diferentes símbolos que aparecen en la gramática se utiliza el concepto de cardinalidad. El número de cadenas que pueden ser generadas a partir de una gramática de contexto libre es infinito. Sin embargo, podemos agrupar las cadenas en conjuntos generados en d derivaciones, denominando cardinalidad al número de elementos del conjunto generado. Cada símbolo no terminal, tendrá una cardinalidad en base al conjunto generado en d derivaciones. Si un símbolo no terminal se puede derivar de varias maneras, la cardinalidad de dicho símbolo no terminal vendrá determinada por la suma de las cardinalidades de cada una de las derivaciones posibles a partir de dicho símbolo. Si una derivación posee más de un símbolo no terminal, la cardinalidad del conjunto formado por los símbolos vendrá determinada por el producto de las cardinalidades de cada símbolo no terminal de la derivación.

Cada individuo se genera a partir del símbolo inicial de la gramática aplicando reglas de producción de manera aleatoria hasta conseguir una cadena válida. El número de derivaciones vendrá determinado por el número máximo de derivaciones establecido en los parámetros de configuración del algoritmo. Para generar un individuo, el algoritmo se inicia con el símbolo inicial de la gramática y con el número máximo de derivaciones. A partir de este símbolo, se buscan las producciones que se pueden obtener en base al número de derivaciones y se elige una de manera aleatoria, teniendo en cuenta que cuanto mayor es la cardinalidad de un símbolo, mayor es la probabilidad de derivarse a partir de dicho símbolo. El algoritmo continúa de forma recursiva por cada símbolo no terminal, reduciendo, en cada iteración, el número máximo de derivaciones.

En base a la población se seleccionan individuos por medio de un torneo binario. Se seleccionan individuos que actuarán de padres para el cruce, que se realizará en base su probabilidad, siendo más probable que el cruce se realice cuanto mayor sea dicha probabilidad. El siguiente paso es realizar la mutación de los individuos seleccionados. Al igual que el cruce, la mutación dependerá de su probabilidad. Mediante el cruce y la mutación, obtenemos nuevos individuos que pasarán a formar parte de la nueva población mediante reemplazo directo. En la Figura 2 se muestra el pseudocódigo de esta primera versión.

1 Algoritmo generacional.

Entrada: $max\_generaciones, N$

Salida: P

```

1:  $P_0 \leftarrow random(N)$ 
2:  $A_0 \leftarrow \emptyset$ 
3: mientras  $num\_generaciones < max\_generaciones$  hacer
4:   Seleccionar padres ( $P_t \cup A_t$ )
5:   Cruce ( $P'$ )
6:   Mutación ( $P'$ )
7:    $P \leftarrow P'$ 
8:   Actualizar población auxiliar ( $A_t$ )
9:    $num\_generaciones++$ 
10: fin mientras

```

Figura 2: Pseudocódigo del algoritmo generacional.

2.4.2 Segunda versión: uso de una población auxiliar

Proponemos una mejora del algoritmo generacional simple mediante una población auxiliar en la que guardaremos los individuos que cumplan una determinada condición. Esta propuesta comenzará, al igual que la primera versión, mediante la generación aleatoria de individuos a partir de la gramática de contexto libre definida y cumpliendo el número máximo de derivaciones posibles. En la generación inicial, la población auxiliar estará vacía.

Los padres o individuos seleccionados para el cruce se seleccionan de la unión de la población actual y de la población auxiliar. Mediante el cruce, se obtienen nuevos individuos que pasarán a formar la nueva población. El proceso continúa de la misma forma que la primera versión propuesta. Una vez que hemos obtenido la nueva población mediante el cruce y la mutación, pasaremos a actualizar la población auxiliar. Para actualizar dicha población, se realizará la unión de la población auxiliar de la generación anterior y la población actual. A continuación, se procede a la eliminación de los individuos que estén repetidos y con el conjunto resultante, se actúa según las propuestas: seleccionando los mejores individuos en base a su función de evaluación, seleccionando aquellos que sobrepasen un umbral de *soporte*, seleccionando aquellos que sobrepasen un umbral de *confianza* o seleccionando aquellos que sobrepasen un determinado umbral de *soporte* y *confianza*. En las figuras 3 y 4 se muestra el pseudocódigo de esta segunda versión.

3 EXPERIMENTACION

La experimentación se ha realizado con 5 conjuntos de datos¹(véase Tabla 1). Para poder realizar comparaciones con el algoritmo *Apriori*, los datos numéricos

¹UCI Machine Learning Repository.

<http://archive.ics.uci.edu/ml/datasets.html>

2 Algoritmo con población auxiliar.

Entrada: $max\_generaciones, N$

Salida: A

```

1:  $P_0 \leftarrow random(N)$ 
2:  $A_0 \leftarrow \emptyset$ 
3: mientras  $num\_generaciones < max\_generaciones$  hacer
4:   Seleccionar padres ( $P_t \cup A_t$ )
5:   Cruce ( $P'$ )
6:   Mutación ( $P'$ )
7:    $P \leftarrow P'$ 
8:   Actualizar población auxiliar ( $A_t$ )
9:    $num\_generaciones++$ 
10: fin mientras

```

Figura 3: Pseudocódigo del algoritmo con población auxiliar.

3 Actualizar población auxiliar.

Entrada: A

Salida: A

```

1:  $A' \leftarrow P' + A_t$ 
2: Ordenar ( $A'$ )
3: Eliminar duplicados ( $A'$ )
4:  $A_t \leftarrow \text{Umbral}(A')$ 

```

Figura 4: Pseudocódigo de la actualización de la población auxiliar.

han sido preprocesados mediante la técnica de discretización *equal-width binning*² [7] en 10 rangos.

Los *datasets* *WDBC*, *WPBC* y *WDatabaseBC*, se corresponden, respectivamente, con los conjuntos de datos: *Wisconsin Diagnostic Breast Cancer*, *Wisconsin Prognostic Breast Cancer* y *Wisconsin Breast Cancer Database*.

El algoritmo G3P presentado se ha desarrollado usando el software JCLEC (*Java Class Library for Evolutionary Computation*) [13], que sirve como *framework* para el desarrollo de aplicaciones de computación evolutiva. Se ha utilizado para la comparación la implementación de *Apriori* existente en el software WEKA [15].

Los parámetros de configuración empleados son: 100 individuos, 100 generaciones, probabilidad de cruce del 70%, probabilidad de mutación del 10% y número máximo de derivaciones de 24. La propuesta de utilizar una población auxiliar se realiza sobre un tamaño de población auxiliar de 60 individuos. Además, se tendrán en cuenta aquellos individuos por encima de los umbrales de *soporte* 0.5 y *confianza* 0.8. El al-

Mining Association Rules from Data Sets.

<http://www.cs.iastate.edu/~neeraj/cs572.html>

²Este método consiste en dividir en rango de valores en intervalos de tamaño constante.

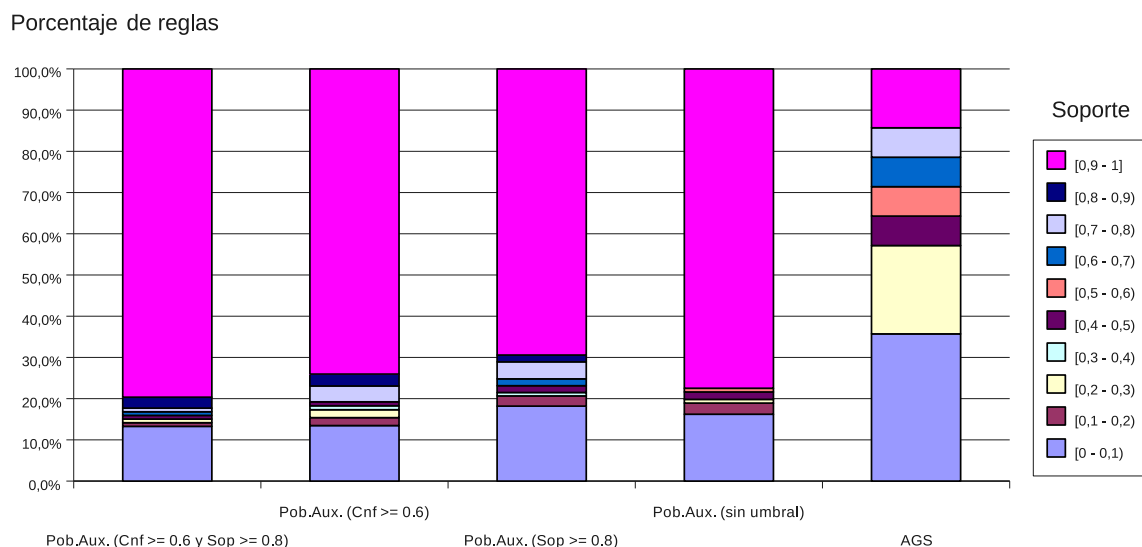


Figura 5: Comparativa entre formas de actualizar la población auxiliar.

Tabla 1: Conjuntos de datos utilizados.

Nombre	Instancias	Atributos	Valores
<i>Soybean</i>	683	36	116
<i>Segment</i>	1500	20	79
<i>WDBC</i>	569	31	302
<i>WPBC</i>	194	34	333
<i>WDatabaseBC</i>	683	11	102

goritmo Apriori, tendrá como umbrales de *soporte* y *confianza* los anteriormente indicados, y como número máximo de reglas a obtener el valor 300000.

4 RESULTADOS

En esta sección mostraremos los resultados obtenidos por las versiones propuestas, realizando una comparación entre las versiones y entre la mejor propuesta y el algoritmo *Apriori*.

4.1 COMPARATIVA ENTRE MODOS DE ACTUALIZAR LA POBLACIÓN AUXILIAR

En este apartado, realizaremos una comparativa entre el algoritmo generacional simple sin población auxiliar y las distintas propuestas de asignación de individuos a la población auxiliar: (a) Población auxiliar con individuos por encima de un umbral de *soporte* de 0.6 y *confianza* 0.8, (b) población auxiliar con individuos por encima de 0.6 como umbral de *confianza*, (c) población auxiliar con individuos por encima de 0.8 como umbral

de *soporte*, (d) población auxiliar con los mejores individuos. Para realizar dicha comparativa, mostraremos el porcentaje de reglas obtenidas para los diferentes intervalos de *soporte*.

En vista de los resultados mostrados en la Figura 5, comprobamos que utilizando una población auxiliar en la que se vayan guardando los mejores individuos que se han ido obteniendo a lo largo de las generaciones, se obtienen mejores resultados que si hacemos uso del algoritmo generacional simple sin población auxiliar. Debido a que estamos buscando *items* frecuentes, entendemos por mejores resultados aquellos que poseen un *soporte* alto. Haciendo uso del algoritmo generacional simple, sólo cerca del 15% de las reglas obtenidas tienen un *soporte* entre el 90% y el 100%. Además, con esta versión del algoritmo, el número de individuos con *soporte* por debajo del 10% es muy elevado.

Descartando esta versión del algoritmo, nos centramos en los diferentes tipos de actualización de la población auxiliar. Comprobamos que la actualización de la población auxiliar con aquellos individuos cuyo *soporte* y *confianza* esté por encima de 0.6 y 0.8 respectivamente, son los que mejores resultados obtiene. Cerca del 80% de las reglas que se obtienen con este tipo de actualización de la población auxiliar tienen un soporte por encima del 90%. Además, el porcentaje de reglas cuyo soporte esté por debajo del 10% es muy bajo en esta versión.

4.2 COMPARATIVA CON APRIORI

En este apartado, realizaremos una comparativa, para cada uno de los *datasets*, entre los resultados obtenidos

con el algoritmo *Apriori* y la mejor propuesta obtenida (uso de una población auxiliar en la que se incluirán aquellos individuos cuyo *soporte* y *confianza* esté por encima de 0.6 y 0.8 respectivamente). Los resultados obtenidos por el algoritmo *Apriori* y por nuestra propuesta son los que se muestran en las tablas 2 y 3 respectivamente, donde *n_reglas* representa el número de reglas obtenido cuyo *soporte* es mayor de 0.5 y *confianza* mayor de 0.8; *Sop_global* representa el *soporte* del conjunto de reglas obtenido; *Sop_med* representa la media del *soporte* del conjunto de reglas; y *Conf_med* la media de la *confianza* del conjunto de reglas obtenido.

Tabla 2: Resultados obtenidos por *Apriori*.

<i>Nombre</i>	<i>n_reglas</i>	<i>Sop_global</i>	<i>Sop_med</i>	<i>Conf_med</i>
<i>Soybean</i>	300000	100%	0.6409	0.9280
<i>Segment</i>	169945	100%	0.5479	0.9615
<i>WDBC</i>	75	84.71%	0.5458	0.8957
<i>WPBC</i>	1	59.27%	0.5927	0.8273
<i>WDatabaseBC</i>	237	88.29%	0.5322	0.9204

Tabla 3: Resultados obtenidos por la mejor propuesta.

<i>Nombre</i>	<i>n_reglas</i>	<i>Sop_global</i>	<i>Sop_med</i>	<i>Conf_med</i>
<i>Soybean</i>	36	100%	0.9171	0.9795
<i>Segment</i>	33	100%	0.9955	0.9993
<i>WDBC</i>	54	100%	0.9997	0.9998
<i>WPBC</i>	51	100%	0.9892	0.9929
<i>WDatabaseBC</i>	50	100%	0.9884	0.9999

Analizando los resultados presentados podemos comprobar cómo, con mayor número de reglas obtenidas, *Apriori* obtiene un *soporte* global menor que el obtenido por nuestra propuesta. Además, nuestra propuesta obtiene un *soporte* y *confianza* promedios mejores que los del algoritmo *Apriori*.

5 CONCLUSIONES Y TRABAJOS FUTUROS

El presente artículo describe un primer intento de aplicar técnicas G3P para descubrir reglas de asociación en un conjunto de datos. Se han utilizado diversas formas de llevar a cabo el algoritmo: algoritmo generacional simple sin hacer uso de ninguna población auxiliar, y mejora del algoritmo generacional simple haciendo uso de una población auxiliar (así como diferentes requisitos que deben satisfacer los individuos para formar parte de la población auxiliar).

Como hemos mostrado, utilizar una población auxiliar en la que se guardan los mejores individuos obtenidos hasta el momento permite obtener un porcentaje mayor de individuos con *soporte* alto que en el caso de un algoritmo generacional simple sin población

auxiliar. Además, de las distintas versiones propuestas para la asignación de individuos a esta población auxiliar, ordenar los individuos candidatos a formar parte de la población auxiliar por orden de *soporte* y que superen un umbral de *soporte* y *confianza* es la que mejores resultados ofrece. Así, en vista de los resultados obtenidos en la Sección 4, nuestra propuesta obtiene un *soporte* global mayor que el algoritmo *Apriori* para cada uno de los 5 conjuntos de datos utilizados. Además, obtenemos menos reglas, por lo que la eficiencia es mayor. A todo esto, añadimos que el *soporte* y *confianza* medios obtenidos son mayores que los conseguidos con el algoritmo *Apriori*.

En futuras investigaciones se abordarán nuevas formas de mutación o de cruce que nos permitan introducir mayor diversidad en el algoritmo y de esta forma ampliar el espacio de búsqueda. Además, se explorará sobre el uso de algoritmos multiobjetivo, en los que podemos maximizar el *soporte* y la *confianza*. Así, comprobamos si este tipo de algoritmos es eficaz en nuestro problema y si supone una mejora a las propuestas presentadas en el presente artículo. Para ello, podemos hacer uso de dos algoritmos multiobjetivo conocidos como son SPEA2 (*Strength Pareto Evolutionary Algorithm*) y NSGA2 (*Non-dominated Sorting Generational Algorithm*). También trataremos de estudiar qué formas de sustitución de individuos son más adecuadas. Obsérvese que para formar parte de la población auxiliar, los individuos deberán compararse según su genotipo y su cadena de derivación. Individuos con el mismo *soporte* o *confianza* no tienen por qué ser igual de ‘importantes’ y podría sustituirse un individuo por otro siendo, a priori, iguales; sin embargo, en generaciones sucesivas podría ocurrir que se demostrase que hubiese sido mejor no sustituirlos.

Finalmente, en futuros trabajos pretendemos explorar el uso de bases de datos multi-relacionales. La mayoría de las investigaciones de minería de datos están enfocadas a buscar patrones sobre simples tablas de datos, pero hoy en día, la mayoría de los datos comerciales están almacenados en bases de datos relacionales usando múltiples tablas conectadas por un identificador. Además, centraremos nuestra investigación en realizar una versión de nuestro algoritmo que trate la incertidumbre y permita obtener reglas de asociación difusas. El enfoque cuantitativo permite a un elemento formar o no parte de un intervalo, lo que nos lleva a una subestimación o sobreestimación de los valores que están cerca de las fronteras de estos conjuntos. Este hecho lleva al desarrollo de reglas de asociación difusas.

Agradecimientos. Este trabajo ha sido financiado por los proyectos del Ministerio de Ciencia y Tec-

nología y de la Junta de Andalucía, TIN2008-06681-C06-03 y TIC-3720, respectivamente.

Referencias

- [1] M. Adda, L. Wu and Y. Feng. *Rare Itemset Mining*. Sixth International Conference on Machine Learning and Applications. IEEE Computer Society (2007).
- [2] R. Agrawal and R. Srikant. *Fast algorithms for mining association rules*, The International Conference on Very Large Databases (1994), 487-499.
- [3] F. Bodon. *A tire-based APRIORI implementation for mining frequent item sequences*. In 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, Chicago, Illinois, USA. ACM Press, New York (2005), 56-65.
- [4] C. Borgelt. *Efficient Implementations of Apriori and Eclat*. In Workshop on Frequent Itemset Mining Implementations. CEUR Workshop Proc. 90, Florida, USA (2003).
- [5] A.E. Eiben, J.E. Smith. *Introduction to Evolutionary Computing*. In 1st Natural Computing Series. Springer, Heidelberg (2003).
- [6] D.E. Golberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York (1998).
- [7] J. Han and M. Kamber. *Data Mining - Concepts and Techniques*. Morgan Kaufmann, 2nd (2006).
- [8] J. Han, J. Pei and Y. Yin. *Mining frequent patterns without candidate generation*. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (2000), 1-12.
- [9] J. Mata, J.L. Alvarez and J.C. Riquelme. *Mining numeric association rules via Evolutionary Algorithm*. In 5th International Conference on Artificial Neural Networks and Genetic Algorithms, Prague (2001), 264-267.
- [10] J. Mata, J.L. Alvarez and J.C. Riquelme. *Discovering Numeric Association Rules via Evolutionary Algorithm*. In Chen, M.-S., P.S., Liu, B. PAKDD 2002. Springer, Heidelberg (2002), 40-51.
- [11] M. Song and R. Sanguthevar. *A transaction mapping algorithm for frequent itemsets mining*, IEEE Transactions on Knowledge and Data Engineering (2006), 472-481.
- [12] R. Srikant and R. Agrawal. *Mining quantitative association rules in large relational tables*. ACM SIGMOD International Conference on Management of Data, Canada (1996).
- [13] S. Ventura, C. Romero, A. Zafra, J.A. Delgado and C. Hervás. *JCLEC: A Java Framework for Evolutionary Computation*. SoftComputing (2008), 381-392.
- [14] M. J. Zaki, S. Parthasarathy, M. Ogihara and W. Li. *New algorithms for fast discovery of association rules*. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (1997), 283-296.
- [15] *Weka Machine Learning Project*. <http://www.cs.waikato.ac.nz/~ml/index.html>

Un Modelo de Clasificación basado en Reglas de Asociación Difusas para Problemas de Alta Dimensionalidad y Selección Evolutiva de Reglas

Jesús Alcalá-Fdez Rafael Alcalá Francisco Herrera

Dept. de Ciencias de la Computación e I.A., CITIC-UGR, Universidad de Granada, {jalcala,alcala,herrera}@decsai.ugr.es

Resumen

El aprendizaje de Sistemas de Clasificación Basados en Reglas Difusas en problemas de alta dimensionalidad se ve dificultado por el crecimiento exponencial que experimenta el espacio de búsqueda cuando el número de ejemplos y/o variables del problema es elevado. En este trabajo presentamos un método de clasificación basado en reglas de asociación difusas y selección evolutiva de reglas para problemas de alta dimensionalidad que nos permite obtener un clasificador preciso y compacto con un bajo coste computacional. Los resultados obtenidos sobre 26 problemas de distintas características muestran la efectividad del método propuesto.

Palabras Clave: Algoritmos Genéticos, Clasificación, Problemas de Alta Dimensionalidad, Reglas de Asociación Difusas.

años se han realizado muchos esfuerzos para hacer uso de sus ventajas en clasificación, dando lugar a un nuevo enfoque denominado clasificación asociativa [5, 7, 16, 20].

Un sistema de clasificación asociativa típico se construye en dos etapas:

- Extracción de las reglas de asociación a partir de la base de datos.
- Seleccionar un pequeño conjunto de reglas de asociación relevantes para construir el clasificador.

Con el objetivo de mejorar la interpretabilidad de las reglas obtenidas y eludir las fronteras no naturales en el particionamiento de los dominios de las variables, algunos investigadores han propuesto métodos para obtener clasificadores basados en reglas de asociación difusas [4, 12, 13].

En este trabajo nosotros presentamos un método de clasificación basado en reglas de asociación difusas y selección evolutiva de reglas para problemas de alta dimensionalidad, denominado FARC-HD, que permite obtener un clasificador preciso y compacto con un bajo coste computacional. Este método consta de 3 etapas:

1. Extracción de reglas de asociación difusas para clasificación.
2. Filtrado de las reglas candidatas.
3. Selección evolutiva de reglas.

Los resultados obtenidos de la comparación con un método reciente para obtener de un clasificador basado en reglas de asociación difusas, CFAR [4], sobre 26 problemas de distintas características muestran la efectividad del método propuesto. Además, este estudio ha sido contrastado a través de técnicas estadísticas no paramétricas.

1 INTRODUCCIÓN

Los Sistemas de Clasificación Basados en Reglas Difusas (SCBRDs) han sido empleados en muchas aplicaciones del mundo real, tales como detección de intrusiones [19], etc. Sin embargo, en muchas de estas áreas los datos de los que disponemos tienen un número de elevado de ejemplos y/o variables. Esto dificulta el aprendizaje de los SCBRDs debido al crecimiento exponencial que experimenta el espacio de búsqueda, dando lugar a problemas de escalabilidad y complejidad.

El descubrimiento de asociaciones es una de las técnicas de la minería de datos que más ha sido utilizada para extraer conocimiento interesante a partir de bases de datos grandes [10]. Durante los últimos

La organización del trabajo es como sigue. En la siguiente sección se introduce brevemente las reglas de asociación difusas para clasificación. En la Sección 3 se presenta el método propuesto, describiendo en detalle cada una de sus etapas. La Sección 4 muestra el comportamiento del método propuesto en 26 problemas reales. Por último, en la Sección 5, se muestran algunas conclusiones.

2 PRELIMINARES: REGLAS DE ASOCIACIÓN DIFUSAS PARA CLASIFICACIÓN

Últimamente, la teoría de los conjuntos difusos ha sido utilizada más y más frecuentemente para describir relaciones entre los datos [14]. El uso de conjunto difusos nos permite extender los tipos de relaciones que se pueden representar [18], por lo que en los últimos años muchos investigadores han propuesto métodos para extraer reglas de asociación difusas a partir de datos cuantitativos [1, 6, 11].

Las medidas más comunes para medir el interés de una regla de asociación son el *soporte* y la *confianza*. Estas medidas pueden ser definidas para reglas de asociación difusas como:

$$\text{Soporte}(A \rightarrow B) = \frac{\sum_{x_p \in N} \mu_{AB}(x_p)}{|N|} \quad (1)$$

$$\text{Confianza}(A \rightarrow B) = \frac{\sum_{x_p \in N} \mu_{AB}(x_p)}{\sum_{x_p \in N} \mu_A(x_p)} \quad (2)$$

donde $|N|$ es el número total de ejemplos, $\mu_A(x_p)$ es el grado de emparejamiento del ejemplo x_p con el antecedente de la regla y $\mu_{AB}(x_p)$ es el grado de emparejamiento del ejemplo x_p con la regla completa.

Una regla de asociación difusa podría ser considerada una regla de clasificación difusa si el antecedente de la regla contiene los ítems difusos (cada ítem difuso es una tupla formada por una variable y un término lingüístico) y el consecuente una clase. Así, una regla de clasificación asociativa difusa como $A \rightarrow \text{Clas}_j$ podría ser medida en términos de soporte y confianza como:

$$\text{Soporte}(A \rightarrow \text{Clas}_j) = \frac{\sum_{x_p \in \text{Clas}_j} \mu_A(x_p)}{|N|} \quad (3)$$

$$\text{Confianza}(A \rightarrow \text{Clas}_j) = \frac{\sum_{x_p \in \text{Clas}_j} \mu_A(x_p)}{\sum_{x_p \in N} \mu_A(x_p)} \quad (4)$$

3 FARC-HD: UN CLASIFICADOR BASADO EN REGLAS DE ASOCIACIÓN DIFUSAS PARA PROBLEMAS DE ALTA DIMENSIONALIDAD

En esta sección se describe nuestra propuesta para obtener un clasificador basado en reglas de asociación difusas para problemas de alta dimensionalidad. Este método está basado en 3 etapas:

1. Extracción de reglas de asociación difusas para clasificación.
2. Filtrado de las reglas candidatas.
3. Selección evolutiva de reglas.

En las siguientes subsecciones introduciremos cada una de estas etapas, describiendo en detalle todas sus características.

3.1 EXTRACCIÓN DE REGLAS DE ASOCIACIÓN DIFUSAS PARA CLASIFICACIÓN

Para generar la base de reglas (BR) inicial empleamos un árbol de búsqueda para obtener los conjuntos de ítems difusos frecuentes para cada una de las clases. El nivel 0 de cada árbol estará formado por el conjunto vacío, mientras que el nivel 1 estará compuesto por todos los ítems difusos según un orden preestablecido. Cada nodo hijo de un nodo A será el resultado de unir dicho nodo con otro nodo del mismo nivel con el que se diferencia en un único ítem. La figura 1 muestra un ejemplo de un árbol obtenido a partir de dos variables (V_1 y V_2) con dos términos lingüísticos (B y A)

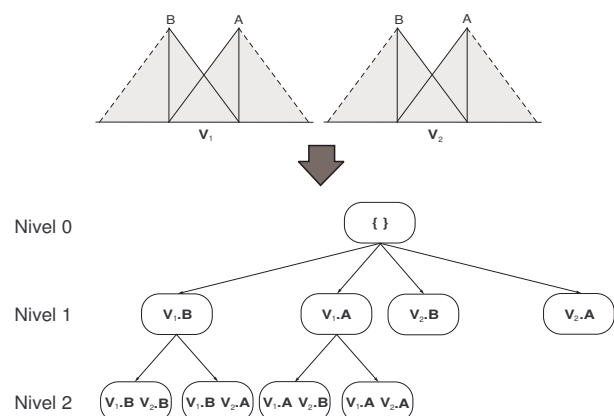


Figura 1: Árbol para las variables V_1 y V_2 con los términos lingüísticos B y A

Los conjuntos de ítems que presentan un soporte mayor que el soporte mínimo indicado por el experto son considerados frecuentes. Si el soporte del conjunto de ítems asociado a un nodo del árbol es menor que el soporte mínimo, este nodo no será extendido más debido a que ninguno de sus descendientes podrá tener un soporte mayor. De igual forma, si la confianza de un nodo es mayor que la confianza mínima indicada no será necesario extender más dicho nodo.

Una vez extraídos todos los conjuntos de ítems frecuentes generamos las reglas de clasificación utilizando los conjuntos de ítems frecuentes como antecedentes de las reglas y la clase como consecuente. Este proceso será repetido para cada una de las clases.

El número de conjuntos de ítems frecuentes depende directamente del soporte mínimo seleccionado. Este es calculado considerando el número total de ejemplos. Sin embargo, el número de ejemplos del que disponemos para cada una de las clases puede ser distinto. Por esta razón, el soporte mínimo es recalculado para cada una de las clases teniendo en cuenta el ratio de ejemplos disponible para cada clase:

$$\min Sop_{Clas_i} = \min Sop * f_{Clas_i} \quad (5)$$

En esta etapa el número de reglas generadas puede ser muy elevado. Sin embargo, es muy difícil para un ser humano manejar una gran cantidad de reglas. Además, si estas reglas tienen muchas variables en el antecedente son difíciles de interpretar. Por esta razón, la profundidad de los árboles será limitada a un valor fijo ($Prof_{max}$) determinado por el experto.

3.2 FILTRADO DE LAS REGLAS CANDIDATAS

Para reducir el coste computacional, en esta etapa hacemos uso del descubrimiento de conjuntos para seleccionar las reglas más interesantes entre las reglas generadas en la etapa anterior por medio de un esquema de ejemplos ponderados [15]. Este esquema trata cada ejemplo de forma que cuando es cubierto por una regla seleccionada este no es eliminado. En su lugar el método almacena para cada ejemplo un contador i con el número de veces que ha sido cubierto por una regla seleccionada y decrementa su peso según la fórmula $p(e_j, i) = \frac{1}{i+1}$ (inicialmente el peso de cada ejemplo es 1,0). Un ejemplo será completamente eliminado cuando haya sido cubierto k_j veces.

Así, en cada iteración del método ordena las reglas según una medida de calidad y reduce el peso de los ejemplos cubiertos. Este proceso es repetido hasta que todos los ejemplos hayan sido cubiertos k_j veces o hasta que no queden más reglas para seleccionar.

En este trabajo nosotros hemos modificado la medida de calidad utilizada en [15] (wWRAcc') para poder manejar reglas difusas. Así, la medida de calidad para una regla $A \rightarrow Clas_j$ se define como:

$$\frac{n''(A \cdot Clas_j)}{n'(Clas_j)} \cdot \left(\frac{n''(A \cdot Clas_j)}{n''(A)} - \frac{n(Clas_j)}{N} \right) \quad (6)$$

donde $n''(A)$ representa el resultado de sumar el peso de cada ejemplo cubierto por su grado de emparejamiento con el antecedente de la regla, $n''(A \cdot Clas_j)$ el resultado de sumar el peso de cada ejemplo cubierto de la clase $Clas_j$ por su grado de emparejamiento con el antecedente de la regla y $n'(Clas_j)$ la suma de los pesos de los ejemplos de la clase $Clas_j$. Además, el primer término de la definición de wWRAcc' ha sido modificado por $\frac{n''(A \cdot Clas_j)}{n(Clas_j)}$ para premiar a aquellas reglas que cubran ejemplos no eliminados de la clase $Clas_j$.

3.3 SELECCIÓN EVOLUTIVA DE REGLAS

En esta etapa hacemos uso de un AG para seleccionar un subconjunto compacto de reglas con una elevada precisión de clasificación a partir de la BR generada en la etapa anterior. En este caso utilizamos el modelo genético del método CHC [8] que ha presentado buenos resultados en problemas de selección binaria [3]. A continuación se explicarán los componentes de este modelo: modelo evolutivo, codificación y población inicial, evaluación, operador de cruce y reinicialización.

3.3.1 Modelo Evolutivo

CHC hace uso de un enfoque de *Selección basado en Poblaciones*. P padres y sus hijos son combinados para seleccionar los P mejores individuos que formarán parte de la siguiente población. Este enfoque utiliza un mecanismo de prevención de incesto y de reinicialización para introducir diversidad en la población, en lugar del operador de mutación.

El mecanismo de prevención de incesto es utilizado para indicar cuando aplicar el operador de cruce, es decir, dos padres son cruzados sólo si su distancia de Hamming dividida por 2 es superior a un determinado umbral, L . Este valor umbral es inicializado como:

$$L = (\#Genes)/4.0 \quad (7)$$

donde $\#Genes$ es el número de genes que tiene un cromosoma. Cuando en una generación no entra ningún

individuo nuevo en la población, L es decrementado. En nuestro caso, L será decrementado cada vez un $\varphi\%$ de su valor inicial, siendo $\varphi\%$ determinado por el usuario (normalmente un 10%).

3.3.2 Codificación y población inicial

Cada cromosoma es un vector binario que determina cuando una regla es seleccionada o no ('1' y '0' respectivamente). Así, un cromosoma C tendrá la siguiente estructura:

Si $c_i = 1$ Entonces $(R_i \in BR)$ sino $(R_i \notin BR)$,

donde R_i es la i -ésima regla de la BR inicial. Para utilizar la información disponible, todas las reglas candidatas son incluidas en la población como una solución inicial. La población inicial se obtiene generando un individuo con todos los genes a '1', y el resto de los individuos son generados aleatoriamente dentro del conjunto $\{0, 1\}$.

3.3.3 Evaluación

Para evaluar un cromosoma calculamos el porcentaje de clasificación y minimizamos la siguiente función:

$$Funcion_{eval}(C) = clas_{por} - w_1 \cdot NR \quad (8)$$

donde NR es el número de reglas seleccionadas (para penalizar subconjuntos de reglas grandes) y w_1 es calculado utilizando el porcentaje de clasificación considerando todas las reglas de la BR inicial,

$$w_1 = \alpha \cdot (clas_{por}_{inicial} / NR_{inicial}) \quad (9)$$

donde α es un peso proporcionado por el experto que determina el equilibrio entre complejidad y precisión (una buena elección es 1,0). Un cromosoma será valorado con un valor $-w_1$ si hay alguna clase para la que no se ha seleccionado al menos una regla.

3.3.4 Operador de cruce

Nosotros utilizamos el operador de cruce HUX [9]. Este operador intercambia exactamente la mitad de los genes que son distintos en los padres, asegurando la máxima distancia entre los hijos y sus padres (exploración).

3.3.5 Reinicialización

Para salir de óptimos locales, CHC utiliza un mecanismo de reinicialización. En este caso, el mejor cromosoma se mantiene en la población y el resto son generados aleatoriamente dentro del conjunto $\{0, 1\}$. Este mecanismo será aplicado cuando el valor umbral L alcance un valor menor que 0.

4 EXPERIMENTOS Y ANÁLISIS DE RESULTADOS

Para analizar el funcionamiento del método propuesto hemos seleccionado 26 problemas reales con distintos tamaños. La tabla 1 resume las principales características de los 26 problemas y muestra el enlace al proyecto KEEL [2] desde el que podemos descargarlos. Para cada problema se muestra el número de Variables (Var), el número de ejemplos (Eje) y el número de clases (Cla).

Tabla 1: Problemas considerados en el análisis

Nombre	Var	Eje	Cla	Nombre	Var	Eje	Cla
Iris	4	150	3	Crx	15	690	2
Phoneme	5	5404	2	Pen-based	16	10992	10
Monks	6	432	2	Segment	19	2310	7
Ecoli	7	336	8	German	20	1000	2
Pima	8	768	2	Twonorm	20	7400	2
Yeast	8	1484	10	Ringnorm	20	7400	2
Glass	9	214	7	Thyroid	21	7200	3
Contracep.	9	1473	3	Wdbc	30	569	2
Page-blocks	10	5472	5	Ionos.	34	351	2
Magic	10	19020	2	Dermato.	34	366	6
Wine	13	178	3	SatImage	36	6435	6
Heart	13	270	2	Spambase	57	4597	2
Cleveland	13	303	5	Sonar	60	208	2

Disponibles en <http://sci2s.ugr.es/keel/datasets.php>

En este estudio nosotros comparamos el método propuesto con el método CFAR [4]. El método está basado en 2 etapas. En la primera genera todas las reglas de asociación difusas posibles mediante una extensión del método Apriori y, en la segunda selecciona las mejores reglas para construir el clasificador.

En los distintos experimentos, se trabaja con un modelo de validación cruzada con 10 particiones de datos, esto es, 10 particiones aleatorias al 10%, y la combinación de cuatro particiones (90%) como entrenamiento y una partición como test. Así se tienen 10 particiones al 90% y 10% en entrenamiento y test. Para cada partición cada método ha sido ejecutado 3 veces y en las tablas se muestran los resultados medios de las 30 ejecuciones de cada algoritmo. Además, hemos aplicado el test no paramétrico de Rangos y Signos de Wilcoxon [17] con un nivel de confianza del 95% ($\alpha = 0,05$) para asegurar que existen diferencias significativas entre los métodos.

Las particiones lingüísticas iniciales consideradas tendrán cinco términos lingüísticos con forma triangular. Los valores de los parámetros utilizados en todos los experimentos son:

Tabla 2: Resultados obtenidos por los métodos CFAR y FARC-HD

Nombre	CFAR				FARC-HD			
	#R	#L	Ent	Test	#R	#L	Ent	Tst
Iris	9,1	1,5	91,3	90,7	4,0	1,0	97,4	96,4
Phoneme	2,0	1,0	70,7	70,7	6,2	1,7	80,2	80,1
Monks	12,1	1,8	100,0	99,6	4,9	1,2	99,7	99,8
Ecoli	4,0	3,1	48,5	47,6	26,0	2,4	86,1	78,0
Pima	2,0	1,9	65,1	65,1	6,6	2,1	77,0	75,9
Yeast	2,0	1,0	16,4	16,4	20,8	2,6	59,0	55,5
Glass	3,4	2,1	44,2	44,7	21,0	2,5	77,5	64,0
Contracep.	2,0	1,0	42,7	42,7	22,8	2,6	58,1	52,0
Page-blocks	2,0	1,0	89,8	89,8	7,9	2,0	94,0	93,8
Magic	2,0	1,5	64,8	64,8	8,9	1,9	80,4	80,4
Wine	115,7	3,0	99,6	93,2	10,2	1,8	98,9	92,1
Heart	45,3	3,3	86,0	82,2	17,4	2,5	92,0	83,2
Cleveland	2,0	2,1	53,9	53,9	39,0	2,9	81,8	56,5
Crx	89,9	5,1	89,5	86,8	9,8	2,3	89,4	85,5
Pen-based	6,9	2,9	36,5	36,4	53,6	2,7	94,2	93,6
Segment	43,8	5,1	56,9	56,9	22,2	2,4	91,1	89,9
German	2,0	2,7	70,0	70,0	23,0	2,5	80,5	71,4
Twonorm	1981,7	3,7	92,0	91,7	11,9	2,1	88,4	87,5
Ringnorm	-	-	-	-	9,8	1,5	88,2	87,9
Thyroid	67,6	5,8	92,6	92,6	3,7	2,5	93,6	93,4
Wdbc	-	-	-	-	6,3	1,2	95,4	94,4
Ionos.	-	-	-	-	15,2	1,9	95,4	90,0
Dermato.	-	-	-	-	23,5	2,4	99,3	90,2
SatImage	-	-	-	-	19,0	2,7	75,6	74,9
Spambase	-	-	-	-	9,8	1,7	84,8	84,8
Sonar	37,3	3,0	83,0	72,5	11,6	2,4	93,8	78,9
Media	121,6	2,6	69,7	68,4	16,0	2,1	86,6	81,9

- FARC-HD: $MinSop = 0,05$, $MinConf = 0,8$, $Prof_{max} = 3$, $k_t = 2$, $Pob = 50$, $Eval = 5.000$, $\alpha = 0,02$
- CFAR: $Minpsop = \{0,05,0,1\}$, $Minpconf = 0,85$, $MS = 0,15$

Destacar que los autores del método CFAR utilizaron 0,1 como soporte mínimo, lo que podría ser muy elevado para algunos de los problemas utilizados en este estudio. Para evitar este problema este método ha sido ejecutado utilizando dos soportes mínimos (0,05 y 0,1), mostrando en las tablas el mejor resultado obtenido con cada uno de ellos en cada problema.

Los resultados obtenidos se muestran en la Tabla 2, donde $\#R$ representa el número medio de reglas, $\#L$ el número medio de términos lingüísticos en el antecedente de las reglas y, Ent y $Test$ el porcentaje medio de acierto sobre las particiones de entrenamiento y test. Hay que destacar que el método

CFAR no puede ejecutarse en 6 de los 26 problemas utilizados en el estudio.

Como podemos observar, el método propuesto obtiene la mejor precisión en media. Además, el número medio de reglas obtenidas es reducido y con menos de 3 términos lingüísticos en el antecedente, lo que facilita su interpretabilidad desde el punto de vista del usuario.

La tabla 3 muestra los resultados de aplicar el test de Wilcoxon sobre los resultados obtenidos en test. Como podemos observar la hipótesis nula es rechazada ($valor-p < \alpha$) y nuestro método es el que obtiene mayor rango, por lo que podemos concluir que nuestro método es el que presenta un mejor funcionamiento.

Tabla 3: Test de Wilcoxon. Rango positivo para FARC-HD (R^+). Rango negativo para CFAR (R^-).

Comparación	R^+	R^-	Hipótesis ($\alpha = 0,05$)	valor-p
FARC-HD vs. CFAR	192	18	Rechazada	0,001

En la tabla 4 se muestran los tiempos de ejecución obtenidos en un Pentium Corel 2 Quad a 2.5GHz, con 4Gb de memoria y con sistema operativo linux. Si analizamos estos tiempos podemos ver como el método CFAR emplea una gran cantidad de tiempo cuando se incrementa el tamaño del problema, mientras que el método propuesto presenta un bajo coste computacional en todos los problemas.

Tabla 4: Tiempos de ejecución (hh:mm:ss)

Nombre	CFAR	FARC-HD	Nombre	CFAR	FARC-HD
Iris	00:00:00	00:00:00	Crx	00:02:02	00:00:02
Phoneme	00:00:05	00:00:13	Pen-based	00:00:07	00:03:15
Monks	00:00:01	00:00:01	Segment	03:08:57	00:00:26
Ecoli	00:00:01	00:00:01	German	00:02:17	00:00:06
Pima	00:00:06	00:00:02	Twonorm	05:03:33	00:01:06
Yeast	00:00:01	00:00:08	Ringnorm	-	00:01:06
Glass	00:00:04	00:00:01	Thyroid	08:57:26	00:00:13
Contracep.	00:00:01	00:00:05	Wdbc	-	00:00:09
Page-blocks	00:01:08	00:00:15	Ionos.	-	00:00:04
Magic	00:16:30	00:01:00	Dermato.	-	00:00:04
Wine	00:00:45	00:00:01	SatImage	-	00:04:51
Heart	00:00:12	00:00:01	Spambase	-	00:03:29
Cleveland	00:00:09	00:00:02	Sonar	00:00:20	00:00:36

5 CONCLUSIONES

En este trabajo hemos presentado un método de clasificación basado en reglas de asociación difusas y selección evolutiva de reglas para problemas de alta di-

mensionalidad, FARC-HD, que nos permite obtener un clasificador preciso y compacto con bajo coste computacional. Si observamos los resultados obtenidos podemos ver como el método propuesto obtiene en media los mejores resultados en precisión y presenta un bajo coste computacional en todos los problemas, obteniendo una buena escalabilidad. Además, este método obtiene modelos con un número reducido de reglas y que incluyen pocas variables en el antecedente, por lo que facilita su interpretabilidad desde el punto de vista de usuario.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio Español de Educación y Ciencia bajo el proyecto TIN2008-06681-C06-01.

Referencias

- [1] J. Alcalá-Fdez, R. Alcalá, M. Gacto, F. Herrera. Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems* 160:7, Pág.905-921, 2009.
- [2] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, F. Herrera. KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing* 13:3, Pág.307-318, 2009.
- [3] J. Cano, F. Herrera, M. Lozano. Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. *IEEE Transactions on Evolutionary Computation* 7:6, Pág.561-575, 2003.
- [4] Z. Chen, G. Chen. Building an associative classifier based on fuzzy association rules. *International Journal of Computational Intelligence Systems* 1:3, Pág.262-273, 2008.
- [5] G. Chen, H. Liu, L. Yu, Q. Wei, X. Zhang. A new approach to classification based on association rule mining. *Decision Support Systems* 42:2, Pág.674-689, 2006.
- [6] M. Delgado, N. Marín, D. Sánchez, M. Vila. Fuzzy association rules: General model and applications. *IEEE Transactions on Fuzzy Systems* 11:2, Pág.214-225, 2003.
- [7] G. Dong, X. Zhang, L. Wong, J. Li. Caep: Classification by aggregating emerging patterns. *Proceedings of the Second International Conference on Discovery Science (DS)*, Lecture Notes in Artificial Intelligence 1721, Pág.30-42, 1999.
- [8] L.J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of genetic Algorithms* 1, Pág.265-283, 1991.
- [9] L.J. Eshelman, J.D. Schaffer. Real-coded genetic algorithms and interval-schemata. *Foundations of Genetic Algorithms* 2, Pág.187-202, 1993.
- [10] J. Han, M. Kamber. Data Mining: Concepts and Techniques. Second Edition. *Morgan Kaufmann*, 2006.
- [11] T. Hong, Y. Lee. An overview of mining fuzzy association rules. *Studies in Fuzziness and Soft Computing* 220, Pág.397-410, 2008.
- [12] Y.-C. Hua, G.-H. Tzeng. Elicitation of classification rules by fuzzy data mining. *Engineering Applications of Artificial Intelligence* 16:7-8, Pág.709-716, 2003.
- [13] Y.-C. Hua. Determining membership functions and minimum fuzzy support in finding fuzzy association rules for classification problems. *Knowledge-Based Systems* 19:1, Pág.57-66, 2006.
- [14] H. Ishibuchi, T. Nakashima, M. Nii. Classification and Modeling with Linguistic Information Granules. *Advanced Approaches to Linguistic Data Mining*, Springer-Verlag, 2004.
- [15] B. Kavsek, N. Lavrac. Apriori-sd: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence* 20:7, Pág.543-583, 2006.
- [16] B. Liu, Y. Ma, C. Wong. Classification using association rules: weaknesses and enhancements. *Data Mining for Scientific and Engineering Applications* 2, Pág.591-602, 2001.
- [17] D. Sheskin. Handbook of parametric and non-parametric statistical procedures. *Chapman & Hall/CRC*, 2003.
- [18] T. Sudkamp. Examples, counterexamples, and measuring fuzzy associations. *Fuzzy Sets and Systems* 149:1, Pág.57-71, 2005.
- [19] C. Tsang, S. Kwong, H. Wang. Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition* 40:9, Pág.2373-2391, 2007.
- [20] B. Liu, W. Hsu, Y. Ma. Integrating classification and association rule mining. *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Pág.80-86, 1998.

Linguistic summarization of data with probabilistic fuzzy quantifiers

F. Díaz-Hermida, A. Bugarín.

Department of Electronics and Computer Science
University of Santiago de Compostela
15782 Santiago de Compostela. Spain.
{felix.diaz,alberto.bugarin.diz}@usc.es

Abstract

In this work we study how a probabilistic based model of fuzzy quantification can be used to build quantified fuzzy summaries. Given a particular set of data, we describe how a set of expressions endowed with a semantics that is convenient and comprehensive for human consumption can be generated for linguistically summarizing data.

Keywords: fuzzy quantification, linguistic summaries, determiner fuzzification schemes.

1 Introduction

Although fuzzy quantification has been presented in the literature as a powerful mechanism for summarizing data [7, 8, 9], building of summaries is not always based on plausible models of fuzzy quantification [1, 2, 3, 6].

In [6] three main issues in fuzzy quantification are identified: interpretation, reasoning and summarization. Interpretation is the main basic goal, as the other issues depend on a correct modeling of fuzzy quantifiers. Both reasoning and summarization depend on the first issue, since a plausible behavior cannot be expected for these tasks when non-plausible models are used for quantified fuzzy reasoning. Therefore, building of linguistic summaries based on fuzzy quantifiers demands using plausible fuzzy quantification models in order to be successful.

In this paper, we show that probabilistic fuzzy quantification models [2, 3, 4] fulfill a number of properties that make them appropriate for the summarization problem. We focus our research in the use of the F^A model [2, 4]. The F^A model is a (non-standard) *Determiner Fuzzification Scheme*¹ (DFS) [6] that has

¹Determiner fuzzification schemes are fuzzy quantifica-

a very solid theoretical behavior and is much more plausible than most of the models defined in the literature. Moreover, the model fulfills other adequacy criteria that makes it very useful for practical applications. As a key point, the fuzzy operators induced by the model are the product tnorm and the probabilistic sum tconorm. This fact makes the F^A model essentially different from the standard models defined in [6].

We can identify two main issues in summarizing data with fuzzy quantifiers. The first one is to discover the best set of quantified expressions that is capable to adequately describe a given set of data, and to build summaries based on them. In the second problem, fuzzy quantifiers are used to guide the retrieval or ranking of fuzzy quantified patterns.

In this paper we deal with the first summarization problem. Given a set of data, we analyze the key concepts to build a set of summaries involving quantifiers to describe the data using the F^A quantification model. An algorithm that accomplishes this task is proposed following these criteria.

2 The F^A quantifier fuzzification mechanisms

To overcome Zadeh's framework to fuzzy quantification Glöckner [6] reconsiders the problem of fuzzy quantification as the problem of looking for adequate procedures to transform semi-fuzzy quantifiers (specification means) into fuzzy quantifiers (operational means). Fuzzy quantifiers are just a fuzzy generalization of crisp or classic quantifiers:

Definition 1: Fuzzy Quantifier. [6, pag. 66] An n -ary fuzzy quantifier $\tilde{Q}$ on a base set $E \neq \emptyset$ is a mapping models fulfilling a very strict axiomatic framework [6], that assures an excellent theoretical behavior. Standard DFSs induce standard fuzzy operators ($\min$ tnorm and $\max$ tconorm).

ping $\tilde{Q} : \tilde{P}(E)^n \rightarrow \mathbf{I} = [0, 1]$. Here $\tilde{P}(E)$ denotes the fuzzy powerset of E .

A fuzzy quantifier assigns a gradual result to each choice of $X_1, \dots, X_n \in \tilde{P}(E)$. An example of a fuzzy quantifier could be **all** : $\tilde{P}(E)^2 \rightarrow \mathbf{I}$ and a reasonable definition for it: **all**(X_1, X_2) = $\inf \{ \max(1 - \mu_{X_1}(e), \mu_{X_2}(e)) : e \in E \}$.

Although a certain consensus may be achieved to accept this previous expression as a suitable definition for **all** this is not the unique one. The problem of establishing consistent fuzzy definitions for quantifiers (e.g., “at least eighty percent”) is faced in [6] by introducing the concept of semi-fuzzy quantifiers. A semi-fuzzy quantifier represents a medium point between classic quantifiers and fuzzy quantifiers, and it is close but is far more general than the idea of Zadeh’s linguistic quantifiers [10].

Definition 2: Semi-fuzzy Quantifier. [6, pag. 71] An n -ary semi-fuzzy quantifier Q on a base set $E \neq \emptyset$ is a mapping $Q : P(E)^n \rightarrow \mathbf{I}$.

Q assigns a gradual result to each pair of crisp sets $(Y_1, \dots, Y_n)$.

An example of a semi-fuzzy quantifiers with the meaning of “at least about 80%” is:

$$ab \geq 80\%(\mathbf{Y}_1, \mathbf{Y}_2) = \begin{cases} S_{0.5,0.8}\left(\frac{|Y_1 \cap Y_2|}{|Y_1|}\right) & Y_1 \neq \emptyset \\ 1 & Y_1 = \emptyset \end{cases}$$

where $S_{0.5,0.8}(x)$ is a Zadeh’s S function.

In order to evaluate fuzzy quantified sentences mechanisms that are capable to transform semi-fuzzy quantifiers into fuzzy quantifiers are needed, i.e., mappings with domain in the universe of semi-fuzzy quantifiers and range in the universe of fuzzy quantifiers. Glöckner names these mechanisms *quantifier fuzzification mechanisms*.

Definition 3: [6, pag. 74] A quantifier fuzzification mechanism (QFM) F assigns to each semi-fuzzy quantifier $Q : P(E)^n \rightarrow \mathbf{I}$ a corresponding fuzzy quantifier $F(Q) : \tilde{P}(E)^n \rightarrow \mathbf{I}$ of the same arity $n \in \mathbb{N}$ and on the same base set.

In [2, 4] the finite QFM F^A is proposed. This model is based on a probabilistic interpretation of fuzzy sets, although it can also be interpreted in a fuzzy way [2, chapter three].

Under the interpretation of fuzzy sets used in the F^A model, membership grades are interpreted as probabilities, and independence between probabilities of different elements is assumed.

Definition 4: F^A . Let $Q : P(E)^n \rightarrow \mathbf{I}$ be a semi-fuzzy quantifier, E finite. The QFM F^A is defined

as

$$F^A(Q)(X_1, \dots, X_n) = \sum_{Y_1 \in P(E)} \dots \sum_{Y_n \in P(E)} m_{X_1}(Y_1) \dots m_{X_n}(Y_n) Q(Y_1, \dots, Y_n) \quad (1)$$

for all $X_1, \dots, X_n \in \tilde{P}(E)$, where $m_X(Y) = \prod_{e \in Y} \mu_X(e) \prod_{e \in E \setminus Y} (1 - \mu_X(e))$.

This model is a finite DFS [2], and to our knowledge the unique non-standard known DFS. Moreover, the theoretical behaviour of the model is excellent [2, chapter three], fulfilling some properties that makes it very adequate for applications. In [4, 2] we have tested the behaviour of the F^A and other quantification models in the information retrieval basic task, providing that the F^A model can work in realistic scenarios.

3 Ruspini quantified partitions and probabilistic QFMs

Let us suppose we use a set of semi-fuzzy quantifiers (“nearly none”, “a few”, “several”, “many” and “nearly all”) to split the quantification universe. If semi-fuzzy quantifiers can be interpreted in a probabilistic way; that is, for a certain proportion x , $\sum_{l_i \in L} l_i(x) = 1$ where the l_i s are the fuzzy numbers used in the definition of the quantifiers, then the F^A model allows us to interpret fuzzy quantifiers in a probabilistic way.

Definition 5: A set of semi-fuzzy quantifiers $Q_1, \dots, Q_r : P^n(E) \rightarrow \mathbf{I}$ forms a probabilistic Ruspini partition of the quantification universe if for all $Y_1, \dots, Y_n \in P(E)$ it holds that $Q_1(Y_1, \dots, Y_n) + \dots + Q_r(Y_1, \dots, Y_n) = 1$.

In figure 1 we show a Ruspini quantified partition of the quantification universe.

The application of the F^A model on a Ruspini partition of quantifiers can be interpreted as a probability, due to the following property, which is fulfilled by this and other probabilistic models [2, chapter three].

Definition 6: [2] A QFM F^A fulfills the property of probabilistic interpretation of quantifiers if for all probabilistic Ruspini partitions of the quantification universe $Q_1, \dots, Q_r : P(E)^n \rightarrow \mathbf{I}$ it holds that²

$$F(Q_1)(X_1, \dots, X_n) + \dots + F(Q_r)(X_1, \dots, X_n) = 1$$

²This property allows us to use the entropy to measure the dispersion of the data for the fuzzy quantifiers. If the entropy is low then most of the probability is concentrated in a small set of quantifiers that describes very well the data (e.g., “many temperatures are low in January”). If the entropy is high then the probability is distributed between several quantifiers.

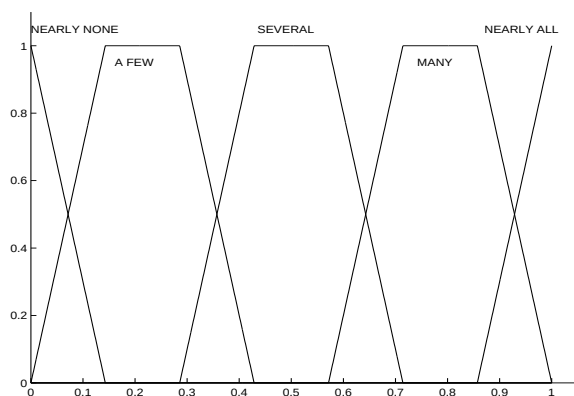


Figure 1: Fuzzy quantified proportional Ruspini partition.

Moreover, for an easy definition of summarization algorithms, we will use Ruspini quantified partitions fulfilling the next symmetry relation: $l_i(x) = l_{w-i-1}(1-x)$. In this way, the Ruspini quantified partition is closed for the antonym property; e.g., “many” is the antonym of “a few” and “nearly all” is the antonym of “nearly none”. Thinking in crisp data, if we know “many data are A” then we also know “few data are not A”, that is also a quantifier of the Ruspini partition. In the same way, if we know “many data are related to label l”, we also know the rest of data are associated to quantifiers “nearly none” or “a few”.

Probabilistic interpretation of quantifiers also allows us to merge quantifiers in a very intuitive way. In figure 2 an hypothetical example of the process to merge quantifiers is presented. We are using a very simple result: the evaluation of the combination of two quantifiers is the sum of the evaluation of the individual quantifiers.

4 Using quantifiers in data summarization

Different kinds of fuzzy quantified summaries can be found in the literature [7, 8, 9], generally linking the building of summaries to specific applications. We can classify them into two different summarization problems:

Explaining the data with fuzzy quantifiers. The problem we deal with in this paper. For a given set of data, we look for procedures that allow us to obtain a group of quantifiers that describes the set of data conveniently. Summaries are associated to these quantifiers.

Quantified pattern detection. In this case, we have some knowledge about the quantification pat-

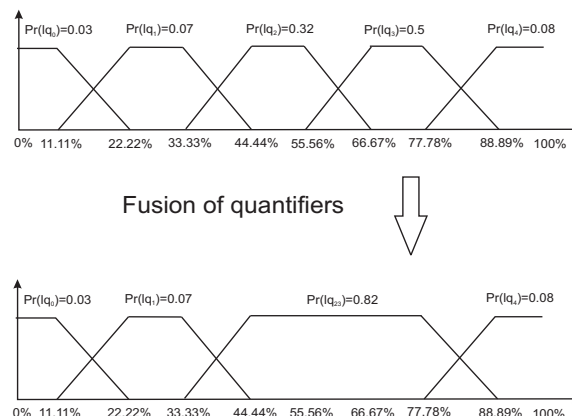


Figure 2: Example of the process of merging of quantifiers. In the bottom of the figure the semi-fuzzy quantifier related to the label lq_{23} has associated the 82% of the probability.

terns that are of interest and can be found in the data. Data are analyzed in search of these patterns and results are ranked using evaluation results of quantified patterns and other criteria as the amount of data covered by the patterns.

Summaries in the first problem are related with the issue of learning the best set of quantifiers to describe the data. By means of quantifiers, we adapt the data to a granularity level that is reasonable for human consumption. That is, we explain the distribution of the data by means of fuzzy quantifiers. In the second problem, a rough idea on the kind of summaries we are searching for in the data is the starting point. Fuzzy quantification is needed for the evaluation of the quantification patterns. Other criteria could be needed to select or rank the interesting summaries, as fuzzy quantification models are only used to analyze the validity of the patterns.

Generating a summary as “many temperatures are hot but a few are warm” for a given daily temperature data set is an example of the first problem, as quantified expressions are used to explain the amount of data associated to each linguistic label (i.e., cold, warm and hot).

The second problem is generally application dependent. For example, it could be interesting in a given application to evaluate the relationship between the number of people injured in traffic accidents and the part of the day they occur, thus generating summaries as “most of the summer days, number of people injured in the morning was higher than in the afternoon”. Here, most is a fuzzy quantifier and higher a fuzzy comparator. And possibly to compare previous pattern with another ones “several summer days,

number of people injured in the morning was less than in the afternoon”

5 Some principles for building quantified summaries

Let us consider a base set E , and let $L = \{l_0, \dots, l_{v-1}\}$ be a linguistic variable defined on a referential universe associated to some particular property of the elements of E (i.e., E could be formed by a set of days and L a linguistic variable defined on the universe of temperatures). Let $F_Q = \{f_{Q_0}, \dots, f_{Q_{w-1}}\}$ be a fuzzy quantified partition of the proportional universe $[0, \dots, 100\%]$ (i.e., “*nearly none*”, “*a few*”, “*several*”, “*many*”, “*nearly all*”). Then the objective is to find a set of quantified expression of the form $Q_i E_s$ are l_j that conveniently summarizes the data.

Generated summaries should be adequate for human consumption. Following data mining principles and conversational principles³ [5, p. 204] we establish a set of criteria to guide the summarization process:

Quality of the summaries. Degree of fulfillment of the summaries should be high. Moreover, well behaved fuzzy quantification models are needed in order to guarantee the quality of the results.

Compatibility of the summaries. A summary composed of the following expressions: “*nearly all the days are hot*” and “*nearly all the days are warm*” is not adequate to human consumption. A good summary should be composed of “compatible quantifiers” covering approximately 100 percent of the data set.

Unambiguity of the summaries. Ambiguous summaries as “*between 25% and 75% of the days are hot*” and “*between 50% and 90% of the days are warm*” are not adequate. It should be noted these two summaries are compatible, but ambiguous.

Minimization of the number of summaries. Summaries consisting of a small number of expressions should be preferred. “Fusion of quantifiers” and other techniques could be employed to increase the quality of the summaries. Moreover, following data mining techniques, summaries explaining a large set of data are more adequate.

6 Building summaries with the F^A model

In this section we propose a greedy algorithm for building summaries using fuzzy quantifiers, following the principles in the previous section.

³We have taken into account *Conversational Maxims of Grice’s theory*.

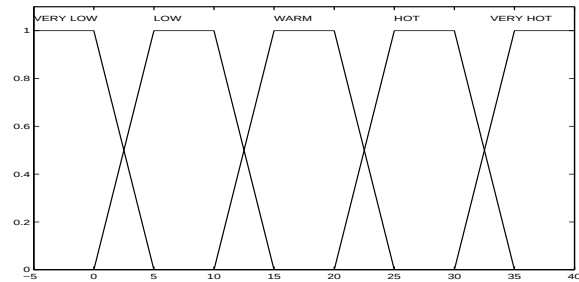


Figure 3: “Fuzzy Temperatures” linguistic variable, in $^{\circ}\text{C}$.

Months	April				
Labels\Quantifiers	nn	f	s	m	na
very low	1	0	0	0	0
low	0	0.72	0.28	0	0
warm	0	0	0.28	0.72	0
hot	1	0	0	0	0
very hot	1	0	0	0	0

Table 1: Evaluation of the quantifiers for the different temperature labels in April.

Let us consider a linguistic variable associated to a certain variable of the data (e.g., “*fuzzy temperature*” in figure 3) and a Ruspini unary proportional quantified partition of the quantification universe (e.g., the one presented in figure 1). Then, a way of summarizing the data is to evaluate the combinations between linguistic labels and linguistic quantifiers. Pairs of label/quantifiers with a high degree of fulfillment are adequate candidates to generate summaries. We will present this idea with an example.

Let us consider a temperature data set (in our experimentation, monthly temperatures in Madrid in January, April and June 2006) (figure 4).

When applying the F^A model to all the possible combinations of temperature labels and quantifiers the results presented in table 1 are obtained. We have evaluated $F^A(Q_i)(lab_j(\text{temperatures}))$ for each possible (i, j) , “*nn=nearly none*, *f=a few*, *s=several*, *m=many* and *na=nearly all*”.

The results shown in table 1 are intuitive. E.g., we find in April that “*many temperatures are warm*” and “*few temperatures are low*” are respectively the most adequate summaries using quantifiers *few* and *many*. Previous matrix is a good summary of the fuzzy set, that adapts the data to the granularities of the fuzzy linguistic variable *temperature* and *fuzzy quantified partition*.

Now we sketch the main ideas of an algorithm to transform previous matrix in a fuzzy quantified summary. For simplicity, a greedy strategy is described.

We firstly introduce an index to measure the adequacy of a given series of quantifiers $Qs = \{Q_i, \dots, Q_h\}$ to summarize a set of data. Definition is based on a sequence or series of quantifiers because consecutive quantifiers of high probability will be merged. We need two previous definitions:

Definition: Specificity of a quantifier. Let $Q : P(E) \rightarrow \mathbf{I}$ be a unary proportional semi-fuzzy quantifier defined by means of a proportional fuzzy number $fn : [0, 1] \rightarrow \mathbf{I}$. We define the specificity of Q as: $Spec(Q) = \int_0^1 fn(x) dx$. Specificity of a quantifier allows us to compare different quantifiers of a Ruspini quantified partition. In general, given two quantifiers of similar probabilities, the most specific one is preferred to summarize a set of data⁴.

Definition: Threshold to merge quantifiers: When we found a consecutive series of quantifiers $Qs = \{Q_i, \dots, Q_h\}$ with a high probability it could be convenient to merge the quantifiers in order to generate a single quantified expression instead of several ones. We will use a threshold to make the decision of merging a series of quantifiers depending on its probability: $u\_fusion = Spec(q_i)^h$. In this manner, less specific (or wider) quantifiers requires a greater probability to be merged with other quantifiers. The $h \leq 1$ parameter allows us to adjust the probability required to merge the quantifiers.

Definition: Adequacy K index of a $Qs = \{Q_i, \dots, Q_h\}$ series of quantifiers. Let $Qs = \{Q_i, \dots, Q_h\}$ be a given series of quantifiers, then we will use the next index K to measure the adequacy of the series of quantifiers to be an adequate summary of a data set:

$$K(Qs) = \sum_{i \leq j \leq h} \Pr(Q_j) \cdot (j - 1) \cdot \frac{1}{(Spec(Q_j))}$$

In previous expression:

- $\Pr(Q_j)$ is the probability of the quantifier. In this way, we look for sequences with a high probability of fulfillment.
- $j - 1$ is used to promote quantifiers covering a significant amount of data, following data mining principles. Quantifier “*nearly all*”, that covers

⁴Here, specificity of a quantifier is not defined in the same way that specificity of a fuzzy set. The idea of the specificity is to measure the amount of possible proportions covered by a proportional quantifier defined by means of a fuzzy number in $[0, 1]$.

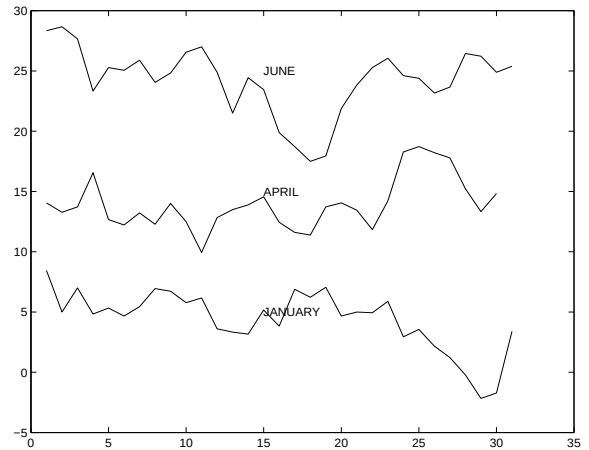


Figure 4: Average daily temperatures in January, April and June of 2006 in Madrid.

a significant amount of data, is preferred to the quantifier “*a few*” that does not cover too much data.

- $\frac{1}{(Spec(Q_j))}$ is used to promote more specific quantifiers.

In this way, the K index would be high for Qs covering a large set of data, with a high probability and a low specificity.

The idea underlying the algorithm is quite simplistic. First we compute a matrix indicating which quantifiers could possibly be merged. Then we look for the best quantifier (or series of quantifiers) to summarize the data (the one that maximizes the K index) and generate the corresponding expression. Then, we repeat the process. In order to generate more understandable expressions, previously used linguistic labels will not be repeated (for human consumption, quantified expressions “*nearly all are hot*” and “*a few are hot*” are incompatible). We also discard generating expressions covering more than 100% of the data. If we interpret the quantifiers as fuzzy numbers, we do not want the sum of the quantifiers associated to the summaries to be greater than 100% (i.e, a summary of data consisting of the expressions “*nearly all are hot*” and “*nearly all are warm*” are clearly not adequate for human consumption). To prevent this, we stop the generation of summaries when the generation of a new summary implies covering more than 100% of the data.

Based on these considerations, we propose the algorithm shown in table 2 to build a quantified summary of a set of data.

Some of the results we have obtained with the previous algorithm for the mentioned temperature data set are:

Computing a quantified linguistic summary
$FL = \{l_0, \dots, l_{v-1}\}$ a linguistic variable $QS = \{Q_0, \dots, Q_{w-1}\}$ a Ruspini quantified partition $R[i, j] = F^A(l_i, QS_j)$, $i = 0, \dots, v-1, j = 0, \dots, w-1$ $Fusion[0 : v-1, 0 : w-1] =$ Possible fusions of quantifiers while we do not cover enough data look for the best series of fusionable quantifiers $Qs = \{Q_i, \dots, Q_h\}$ without repeating linguistic labels without the summaries surpass 100%. Generate a quantified expression for Qs end while

Table 2: Algorithm for generating a quantified summary of a set of data

January: “Many temperatures are low”, “a few temperatures are very low”

April: “Many temperatures are warm”, “a few temperatures are low”

June: “Several or many temperatures are hot”, “a few temperatures are warm”.

For example, in the case of January previous algorithm generates a summary indicating that “many temperatures were low”. Daily temperatures for which the “low” description is not adequate were only “a few” and are associated to the label “very low”. We want to emphasize the simplicity and intuitiveness of generated summaries, very adequate for human consumption.

6.1 Conclusions and future work

In this work we have presented an initial proposal to build linguistic quantified summaries with probabilistic quantification models. Given a particular data set, we have studied how to build a set of fuzzy quantified expressions for summarizing the data, convenient for human consumption.

Several relevant open questions still remain in relation to the proposal. Other criteria and algorithms to guide the building of summaries should be studied. Moreover, our approach deals with unary proportional quantified expressions. Using other quantifiers have not been studied, although a similar approach could be used to generate summaries with binary proportional quantifiers. In “Q of the $\underline{l}$ man are $\underline{r}$ paid”, by fixing the label associated to the first variable “man” (e.g. $\underline{l} = \text{young}$) a similar procedure could be used to compute a quantified set of expressions explaining how

many man are “ $\underline{r}$ paid” given they are young.

Another important question is to make a deep study of the second problem of summarizing with fuzzy quantifiers (detect the most interesting set of quantified patterns to summarize particular events).

References

- [1] M. Delgado, D. Sánchez, and M. A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *International Journal of Approximate Reasoning*, 23(1):23–66, 2000.
- [2] F. Díaz-Hermida. *Modelos de cuantificación borrosa basados en una interpretación probabilística y su aplicación en recuperación de información*. PhD thesis, Universidad de Santiago de Compostela, 2006.
- [3] F. Díaz-Hermida, A. Bugarín, P. Cariñena, and S. Barro. Voting model based evaluation of fuzzy quantified sentences: a general framework. *Fuzzy Sets and Systems*, 146:97–120, 2004.
- [4] F. Díaz-Hermida, David. E. Losada, A. Bugarín, and S. Barro. A probabilistic quantifier fuzzification mechanism: The model and its evaluation for information retrieval. *IEEE Transactions on Fuzzy Systems*, 13(1):688–700, 2005.
- [5] L.F.T. Gamut. *Logic, Language and Meaning*, volume II of *Logic, Language, and Meaning*. The University of Chicago Press, 1984.
- [6] I. Glöckner. *Fuzzy Quantifiers: A Computational Theory*. Springer, 2006.
- [7] I. Glöckner. Optimal selection of proportional bounding quantifiers in linguistic data summarization. In *Proceedings of SMPS2006 - Special Session on Applications and Modelling of Imprecise Operators, Published in Soft Methods for Integrated Uncertainty Modelling, LNCS, Springer*, pages 173–181, 2006.
- [8] A. Wilbik J. Kacprzyk and S. Zadrozny. Linguistic summarization of time series using a fuzzy quantifier driven aggregation. *Fuzzy Sets and Systems*, 159:1485–1499, 2008.
- [9] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–191, 1988.
- [10] L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Comp. and Machs. with Appls.*, 8:149–184, 1983.

CONSTRUCCION DE CONJUNTOS DE REGLAS DIFUSAS PARA LA CLASIFICACION DE OBJETOS

C.J. Solana-Ciprés¹ J. Albusac¹ J.J. Castro-Schez¹ J. Moreno-García² L. Rodríguez-Benítez¹

¹ Escuela Superior de Informática, Paseo de la Universidad, 4, 13071, Ciudad Real, cayetanoj.solana@uclm.es

² E.U.I.T. Industrial, Avda. Carlos III s/n, 45071, Toledo, juan.moreno@uclm.es

Resumen

En este artículo se presenta un método para la obtención de un conjunto de reglas que puedan ser usadas en la clasificación de objetos en entornos monitorizados. La propuesta está basada en un algoritmo de aprendizaje automático supervisado y toma como información de entrada los resultados obtenidos en un algoritmo de segmentación previamente desarrollado. El método propuesto genera un conjunto de reglas interpretables que pueden ser usadas durante la monitorización de entornos vídeo-vigilados.

Palabras Clave: Vídeo-vigilancia, algoritmo de aprendizaje automático, clasificación de objetos.

1. INTRODUCCION

Hoy en día, existe una preocupación generalizada por la seguridad de las personas y los edificios frente a actos vandálicos o terroristas. En este contexto, se origina un notable aumento de interés en la investigación en el área de la vídeo-vigilancia inteligente. Generalmente, un sistema de vídeo-vigilancia avanzado se divide en los módulos mostrados en la Figura 1.

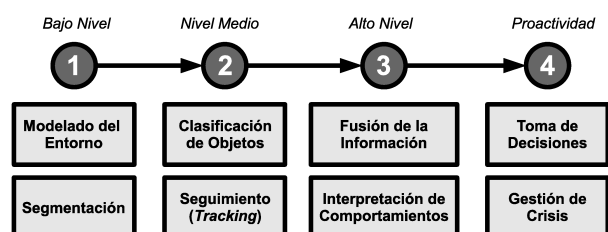


Figura 1: Arquitectura de un sistema de vigilancia.

Dentro de esta división, existe un amplio abanico de

líneas de investigación abiertas. En este artículo se propone un método para poder construir el módulo de clasificación de objetos (*nivel medio*). Este tipo de sistemas suele ejecutar la clasificación después de la segmentación, por lo que la calidad de los resultados de la etapa de segmentación será fundamental para obtener resultados óptimos en el proceso de clasificación.

Habitualmente, los métodos de etiquetado de objetos extraen las características relevantes a partir de la información de los frames capturados. Dichas características suelen basarse en la apariencia, forma, color, movimiento y textura de los objetos detectados. Hu et al. [5] dividen los métodos de clasificación en dos categorías principales: basada en la forma y basada en el movimiento. Sin embargo, Brown [3] distingue entre métodos dependientes o independientes de la perspectiva de la cámara. Por último, Valera y Velastin [9] dividen las propuestas en función del conocimiento geométrico de los objetos a seguir: 2D o 3D.

En trabajos recientes, se ha propuesto una alternativa para resolver la etapa de segmentación a través de un algoritmo de tiempo real sobre el dominio comprimido H.264/AVC [8] y un marco conceptual basado en el análisis de la normalidad para detectar comportamientos anómalos en entornos monitorizados [2]. Dicho esquema se compone de un conjunto de componentes de normalidad que definen el comportamiento de cada objeto desde el punto de vista de la vigilancia. Entre otros, los aspectos a considerar son la trayectoria que sigue un objeto, la velocidad, la normalidad en un paso de peatones, etc. Un objeto se comporta de forma apropiada en un entorno monitorizado si satisface la normalidad definida para cada uno de los componentes de normalidad, los cuales [2] necesitan conocer la clase a la que pertenece cada objeto para poder realizar la interpretación apropiada del comportamiento. Por los motivos expuestos, se puede considerar la clasificación o etiquetado de objetos como una etapa relevante en cualquier sistema de vigilancia avanzado porque permite detectar situaciones anormales.

En el presente trabajo se propone un método para construir un conjunto de reglas útiles para la clasificación de objetos en escenas de vídeo que utiliza un algoritmo de aprendizaje automático supervisado [1]. Dicho algoritmo comienza el proceso de aprendizaje a partir de un conjunto de entrenamiento en el que cada clase de objeto es descrita por un conjunto de ejemplos definidos por medio de un conjunto de características. Es decir, cada ejemplo del conjunto de entrenamiento estará asociado a la clase a la que pertenece. Esta asociación se puede hacer de dos formas, la primera es manual y requiere que sea realizada por un experto en el dominio. La segunda requiere de un proceso automático que realice la asociación. El hacerlo de manera manual tiene como ventaja principal la eliminación de ruido en el conjunto de entrenamiento, lo cual puede tener como efecto la obtención de mejores resultados por el algoritmo de aprendizaje automático; como desventaja hay que destacar que es una tarea lenta y tediosa. En este trabajo se opta por la segunda aproximación para reducir la actividad humana en el proceso, uno de los objetivos del presente trabajo.

A continuación, en la Sección 2 se analiza detalladamente la solución propuesta. En la Sección 3 se muestran los resultados experimentales obtenidos y finalmente en la Sección 4 se resumen las conclusiones y las líneas de investigación abiertas.

2. METODO PROPUESTO

En este trabajo se presenta un método que permite aprender un conjunto de reglas que puedan ser usadas en el diseño y desarrollo de un sistema para la clasificación de objetos móviles en entornos monitorizados. El método propuesto está basado en un algoritmo de aprendizaje automático supervisado. Dicho algoritmo tiene como objetivo aprender un conjunto de reglas lingüísticas que modelen la función S definida como $S : V \rightarrow O$. Donde O es el conjunto de clases de objetos que se mueven por la escena vigilada y V es información obtenida en la etapa de segmentación: posición vertical (v_1) y horizontal (v_2) del objeto, tamaño del objeto en número de macrobloques (v_3) y en número de vectores de movimiento (v_4), relación entre anchura y altura del objeto (v_5) y una medida de dispersión del campo de vectores de movimiento pertenecientes al objeto (v_6), es decir, $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$. Por su parte, el algoritmo de aprendizaje toma como entrada un conjunto de entrenamiento $E = \{e_1, e_2, \dots, e_n\}$ donde cada ejemplo es $e_i = (v_{i1}, v_{i2}, \dots, v_{i6} : o_i)$, siendo v_{ij} el valor del ejemplo i para la variable v_j y o_i el tipo del objeto detectado en la escena ($o_i \in O$).

Las características que se miden son representativas de los diferentes tipos de objetos porque definen su

posición (v_1 y v_2), tamaño (v_3 y v_4) y forma (v_5). La posición y el tamaño son características estrechamente relacionadas porque un objeto será más grande o más pequeño en función de su posición respecto a la cámara. Por su parte, la forma supone un factor de distinción clave entre personas y vehículos, como se observa en la Tabla 1, donde se muestra el porcentaje de cada tipo de objeto en función de su ratio anchura/altura. Se ha llevado a cabo un estudio para identificar la relación entre la altura y la anchura de los objetos en una secuencia de vídeo específica; se concluye que las personas y las motocicletas son más altas que anchas, mientras que los coches y camiones son más anchos que altos. Finalmente, el atributo de la dispersión (v_6) permite conocer la rigidez de un objeto dado que una persona o un grupo de personas serán codificados con diferentes vectores de movimiento debido a sus articulaciones, mientras que un vehículo será codificado con vectores de movimiento similares en dirección y tamaño.

Cuadro 1: Ratio Anchura (W) / Altura (H).

	$W > H$	$W = H$	$W < H$
Persona	10.53 %	1.28 %	88.19 %
Grupo	32.44 %	4.82 %	62.74 %
Motocicleta	0 %	0 %	100 %
Vehículo	96.32 %	0.85 %	2.83 %
Camión	93.94 %	0.4 %	5.66 %

En la Figura 2 se muestra un esquema general del método propuesto para la construcción de reglas que puedan emplearse en la construcción del sistema de clasificación de objetos de una escena. La información de entrada la constituye el flujo de vídeo comprimido en formato H.264/AVC [6] obtenido de cámaras de vigilancia; como salida se obtiene el conjunto de reglas difusas que permite clasificar los diferentes tipos de objetos de forma automática. En la primera parte de la propuesta se realiza la *adquisición de información* a través de cámaras de vigilancia, se aplica un algoritmo de segmentación previamente diseñado [8] y se adapta la información resultante para su uso en las siguientes fases. En la segunda fase (*etapa de entrenamiento*) se aplica el algoritmo de clustering k-means y se construyen de manera automática los conjuntos de entrenamiento necesarios para la tercera etapa. En la *etapa de aprendizaje*, el algoritmo de aprendizaje automático aprende el conjunto de reglas difusas que caracteriza los diferentes tipos de objetos móviles a partir de los conjuntos de entrenamiento generados en la etapa anterior. De este modo, a partir de las reglas generales se puede deducir a qué tipo pertenecerá un objeto móvil en una secuencia de vídeo. Cada una de las etapas será analizada brevemente a continuación.

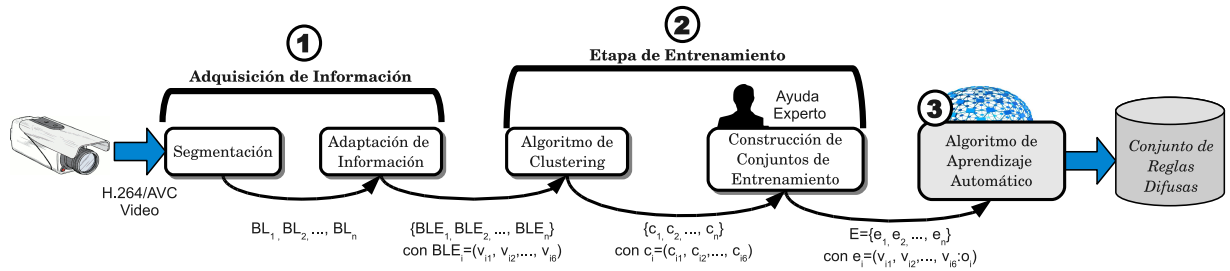


Figura 2: Esquema general del sistema propuesto.

2.1. ETAPA DE ADQUISICIÓN DE INFORMACIÓN

2.1.1. Algoritmo de segmentación

En esta etapa, se utiliza razonamiento aproximado para la segmentación de los objetos móviles en escenas de vídeo [8]. La propuesta de segmentación ha sido implementada utilizando el software H.264/AVC JM 15.1 [6]. El resultado del algoritmo es un conjunto de blobs lingüísticos o regiones [8]. Un **blob lingüístico** se define como una 7-tupla compuesta por uno o más vectores de movimiento conceptualmente similares que representa una región de manera lingüística:

$$BL = \langle NF, Tam, MB, I_{PH}, I_{PV}, I_{VH}, I_{VV} \rangle \quad (1)$$

donde NF es el número de frame en el que se localiza el blob, Tam es el número de vectores de movimiento lingüísticos [7] agrupados en el blob, MB es la lista de los macrobloques pertenecientes al blob y los cuatro últimos elementos son los intervalos lingüísticos que representan la posición y velocidad del blob: *Posicion Horizontal*, *Posicion Vertical*, *Velocidad Horizontal* and *Velocidad Vertical*. Dado que la lista de los macrobloques se almacena para cada blob (MB), los objetos se pueden reconstruir mediante un proceso de defuzzificación para obtener su representación visual, como se observa en la Figura 3.

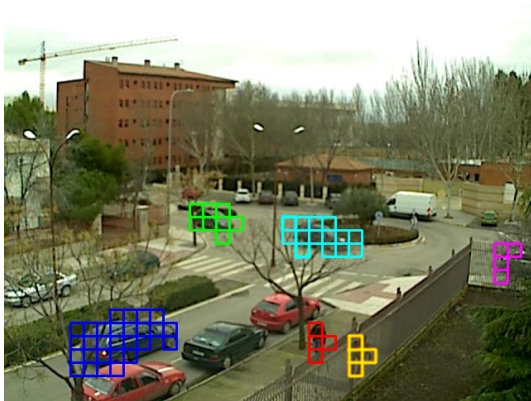


Figura 3: Ejemplo de frame segmentado.

El algoritmo de segmentación basado en razonamiento aproximado permite obtener la descripción de regiones (*blobs*) mediante características como su forma, tamaño, posición y velocidad. Estas representaciones lingüísticas caracterizan a los objetos y facilitan la clasificación de los mismos en diferentes grupos predefinidos como personas, grupos de personas, motocicletas, coches o camiones.

2.1.2. Adaptación de información

El algoritmo k-means es una de las técnicas más sencillas y extendidas para la agrupación de objetos basándose en particiones, donde cada partición representa un grupo o *cluster*. Sin embargo, para ser aplicado, la información de entrada debe representarse mediante un conjunto de características numéricas, por lo que el resultado del algoritmo de segmentación se debe transformar. Dicha adaptación se lleva a cabo mediante la aplicación de una función que transforma cada elemento BL_i en un nuevo elemento BLE_i definido sobre el conjunto de variables utilizadas en el algoritmo de aprendizaje. Así, un **blob lingüístico extendido** BLE es una 6-tupla que define numéricamente una región móvil:

$$BLE = \langle v_1, v_2, v_3, v_4, v_5, v_6 \rangle \quad (2)$$

donde cada elemento se obtiene del siguiente modo:

- $v_1 = \mu_{PV}^1(I_{PV})$, es decir, v_1 se obtiene mediante la defuzzificación de la variable *Posicion Vertical*.
- $v_2 = \mu_{PH}^1(I_{PH})$, es decir, v_2 se obtiene mediante la defuzzificación de la variable *Posicion Horizontal*.
- $v_3 = size(MB)$, v_3 es el número de elementos de la lista MB .
- $v_4 = Tam$, luego v_4 es el segundo elemento del objeto BL .
- $v_5 = width(MB)/height(MB)$, es decir, v_5 se calcula como la relación entre la anchura y la altura en macrobloques del objeto si su límite se definiese dentro de un rectángulo.

- $v_6 = disp(BL)$, donde $disp(BL)$ es una medida de dispersión del campo de vectores de movimiento del objeto.

Para la obtención de $disp(BL)$, es necesario obtener previamente el vector que define el desplazamiento global del blob, denominado **vector de velocidad general**, el cual se calcula mediante la defuzzificación de las variables *Velocidad Horizontal* y *Velocidad Vertical*:

$$VVG(BL) = \langle \mu_{VH}^1(I_{VH}), \mu_{VV}^1(I_{VV}) \rangle \quad (3)$$

Así, a partir de este vector, se obtiene $disp(BL)$ calculando la suma de las diferencias entre los vectores de movimiento lingüísticos pertenecientes al blob y el correspondiente *vector de velocidad general*. Finalmente, el resultado se divide entre el número de vectores de movimiento del blob o región.

2.2. ETAPA DE ENTRENAMIENTO

2.2.1. Algoritmo de clustering

El objetivo de la primera parte de la etapa de entrenamiento consiste en generar los grupos (*clusters*) que se utilizarán para la construcción del conjunto de datos de entrenamiento. Para ello, se sugiere la utilización del algoritmo k-means. Como se ha mencionado previamente, este algoritmo será aplicado para obtener los *clusters* a los que pertenece cada tipo de objeto. Así, en esta etapa se genera un conjunto de *clusters* $C = \{C_1, C_2, \dots, C_n\}$, donde cada *cluster* C_j se identifica con su centroide c_j , donde $c_j = (c_{j1}, c_{j2}, \dots, c_{j6})$ siendo cada c_{ji} el valor que toma el centroide j en la variable v_i , y es asociado a una clase de objeto o_j (tal que $o_j \in O$), es decir $C_j = (c_j : o_j)$.

Dado que el algoritmo k-means trata de agrupar la información de características similares, una de ellas el tamaño de los objetos, aparece el problema de la perspectiva: diferentes tipos de objetos pueden tener el mismo tamaño en función de su posición con respecto a la cámara y, por tanto, ser agrupados en un mismo *cluster*. Para evitar este problema, se propone la división de la información de entrada en función de la distancia a la cámara. La Figura 4 muestra un ejemplo de escena particionada en la que se utilizan sólo divisiones horizontales; sin embargo, la escena se puede dividir de diferentes maneras en función de las recomendaciones de un experto. En este ejemplo, la escena se divide en cuatro regiones: *Fuera*, *Lejos*, *Medio* y *Cerca*. De este modo, se pueden discriminar las áreas de la imagen en las que no debería haber objetos móviles (*Fuera*) y aquellas con menor o mayor distancia a la cámara. Así, el algoritmo k-means se aplica sobre los objetos móviles de cada área y obtiene un conjunto de

clusters diferenciados para cada una de ellas. Después de un número determinado de iteraciones, los objetos son agrupados en torno a los centroides. El algoritmo termina cuando los centroides no se modifican en dos iteraciones consecutivas o cuando se alcanza una iteración prefijada (*poda*).



Figura 4: Ejemplo escenario particionado.

La calidad de los resultados depende de la selección inicial de los centroides. Habitualmente, el conjunto inicial de centroides se selecciona de forma aleatoria porque el algoritmo k-means tiene la habilidad de distribuir los centroides de forma eficiente si el conjunto de entrenamiento es suficientemente amplio. Esta opción es la más rápida, pero no la más eficiente. Es preferible inicializar los centroides utilizando una heurística que facilite la distribución de los grupos. En este trabajo, se utiliza un algoritmo para la selección automática de centroides iniciales basado en la variable con mayor varianza.

Una vez que se obtiene el conjunto inicial de centroides para cada partición de perspectiva, el algoritmo k-means distribuye el conjunto de entrenamiento en k grupos modificando la representación de los grupos iniciales. Como resultado, se obtiene la caracterización de los k grupos. Por ejemplo, el conjunto de grupos para la partición *Cerca* podría ser el siguiente con $k = 5$:

$$\begin{aligned} c_0 &- (26.5, 16.9, 10.9, 45.3, 1.05, 3.7) \\ c_1 &- (24.7, 23.4, 4.1, 8.2, 0.54, 2.8) \\ c_2 &- (26.4, 18.1, 8.3, 26.9, 1.03, 3.2) \\ c_3 &- (25.4, 4.0, 23.6, 154.8, 2.12, 4.9) \\ c_4 &- (27.2, 16.2, 4.6, 13.8, 0.96, 3.0) \end{aligned}$$

En este punto, existe un conjunto de grupos característicos para cada partición del escenario. Sin embargo, se debe etiquetar el tipo de objeto correspondiente a cada grupo en función de las propiedades que lo definen. Este proceso lo lleva a cabo un experto de forma manual. Por ejemplo, los centroides anteriormente citados podrían ser etiquetados en los siguientes grupos: $C = \{c_0 : \text{Grupo}; c_1 : \text{Persona}; c_2 : \text{Grupo}; c_3 : \text{Vehículo}; c_4 : \text{Persona}\}$.

2.2.2. Construcción de conjuntos de entrenamiento

Una vez que los clusters han sido generados, la siguiente fase de la etapa de entrenamiento consiste en la generación del conjunto de entrenamiento E donde cada ejemplo se define como $e_i = (v_{i1}, v_{i2}, \dots, v_{i6} : o_i)$. Este proceso se ejecuta automáticamente mediante la comparación de cada blob lingüístico extendido $BLE_i = (v_{i1}, v_{i2}, \dots, v_{i6})$ con cada centroide $c_j = (c_{j1}, c_{j2}, \dots, c_{j6})$ obtenido en la fase anterior. Así, el ejemplo de entrada e_i será etiquetado con el tipo de objeto o_j del cluster C_j más cercano. La similaridad entre un blob lingüístico extendido BLE_i y un cluster C_j se calcula mediante la distancia Euclídea de las seis variables:

$$distancia(BLE_i, C_j) = \sqrt{\sum_{n=1}^6 (v_{in} - c_{jn})^2} \quad (4)$$

Por ejemplo, siendo $BLE_0 : (26, 8, 20, 144, 2, 3)$, la distancia al grupo C_3 será: $distancia(BLE_0, C_3) = 12,23$. El centroide c_3 asociado al grupo C_3 es el más cercano a BLE_0 , por lo que se puede concluir que el ejemplo e_0 tendrá las seis variables v_i del blob BLE_0 y se le asignará como objeto o_0 el valor *Veh culo* dado que C_3 ha sido etiquetado como grupo *Veh culo*. De este modo, los objetos detectados en el algoritmo de segmentación podrían ser usados en la construcción de los conjuntos de entrenamiento.

2.3. ETAPA DE APRENDIZAJE

Para el aprendizaje de la función S a partir del conjunto de entrenamiento E se va a hacer uso del algoritmo de aprendizaje automático presentado por Castro et al. [4]. Dicho algoritmo trabaja en tres fases. La primera de ellas consiste en la división de los dominios de las variables de entrada (en este caso, son seis). Cada uno de los dominios de las variables de entrada se divide en varias regiones, cada una de las cuales será presentada por medio de una función trapezoidal. La elección de las regiones que conforman tal división se podrá hacer de forma manual en la que un experto hace la división del dominio, determinando cuantos valores puede tomar cada variable y la definición de cada valor, o de manera automática empleando algunas heurísticas para hacer la mejor división del dominio posible.

En la segunda fase, cada ejemplo del conjunto de entrenamiento se convierte en una regla inicial o básica, la cual carece de toda generalidad. De hecho, esta regla inicial únicamente especifica cómo clasificar un ejemplo muy concreto o un conjunto de ejemplos muy próximos, es decir, cómo clasificar el ejemplo o

ejemplos a partir del cual se ha generado la regla. Por tanto, en el peor de los casos, existirán tantas reglas particulares como ejemplos haya en el conjunto de entrenamiento. En cada una de estas reglas, cada valor de cada variable de entrada será presentado por medio de una región de su dominio y la salida del ejemplo será codificada.

Finalmente, la tercera fase tiene como objetivo reducir el conjunto de reglas iniciales para construir reglas más generales. Cada una de las reglas generales abarca varias reglas particulares o iniciales. Para ello se utiliza un proceso de ampliación en el que una regla inicial se extiende, con valores del dominio aún no considerados, cubriendo nuevas situaciones del espacio global de posibles situaciones, para las cuales no existen contraejemplos, es decir no están cubiertas por reglas iniciales que clasifiquen de distinta forma. Para una especificación detallada del algoritmo ver [4] y una modificación del mismo aplicado a la adquisición de conocimiento en entornos monitorizados en [1]. Esta modificación ha sido necesaria para evitar un problema de sobregeneralización que se produce con la propuesta inicial y que en este ámbito tiene efectos no deseados. A continuación se muestran ejemplos de reglas que han sido aprendidas con el método propuesto:

1. SI v_1 es {CentroVertical, Arriba} y v_3 es {Grande} y v_6 es {Pequeño} ENTONCES o_i es *Veh culo*.
2. SI v_1 es {Abajo} y v_5 es {MuyPequeño, Pequeño} y v_6 es {Grande, MuyGrande} ENTONCES o_i es *Persona*.

3. RESULTADOS

El sistema de clasificación ha sido evaluado en secuencias de vídeo comprimido con formato H.264/AVC utilizando el codificador J.M. 15.1. La configuración del codificador ha sido fijada con perfil *baseline* y modo de optimización *RD low complexity* [6]. La resolución de los vídeos es de 640x480 píxeles. En la Figura 5 se muestra una captura de una secuencia de vídeo en la que los blobs aparecen dibujados en el frame y etiquetados en la parte derecha de la imagen¹.

En la Tabla 2 se muestra una comparativa de los resultados obtenidos en dos secuencias de vídeo para la clasificación de objetos a partir de los clusters generados en la etapa de entrenamiento. En este caso, cada objeto se clasifica del tipo perteneciente al clúster con menor distancia (Ecuación 4). Se denomina *probabilidad de detección* a la probabilidad de que un objeto

¹Se ha desarrollado un sistema web para la clasificación automática de objetos en una secuencia de vídeo; se encuentra disponible en: <http://oreto.esi.uclm.es/aplicaciones/labelling/>

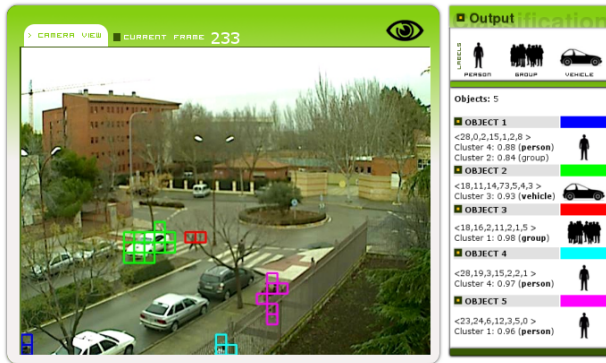


Figura 5: Ejemplo de la aplicación web.

móvil sea detectado y *precision* a la probabilidad de que un objeto detectado esté bien etiquetado. Utilizando este sistema de clasificación, el experto sólo ha de seleccionar a qué tipo de objeto pertenece cada clúster, sin embargo, la interpretabilidad de los resultados es más deficiente que si se usa el conjunto de reglas difusas generado en la fase de aprendizaje.

Cuadro 2: Precisión y probabilidad de detección.

	Vídeo 1	Vídeo 2
Frames	1375	586
Objetos reales	5995	897
Objetos detectados	4958	822
Prob. detección	82.7 %	91.6 %
Precisión	87.3 %	93.1 %

4. CONCLUSION

Se ha presentado un método para la construcción de un conjunto de reglas que puedan ser usadas en la clasificación de objetos, basado en un algoritmo de aprendizaje supervisado. El conjunto de reglas construido determina el tipo al que pertenece cada objeto detectado en una escena. El conjunto de entrenamiento que toma como entrada el algoritmo de aprendizaje se genera de forma automática utilizando técnicas de clustering en la etapa de entrenamiento. Finalmente, hay que destacar que las reglas aprendidas podrían ser usadas en tiempo real.

Como trabajo futuro, se propone dividir la perspectiva del escenario de forma difusa para evitar las imprecisiones que puede acarrear la agrupación de los objetos en tres sub-áreas predefinidas. Por otro lado, se propone el estudio de las ventajas que conlleva la aplicación del algoritmo de aprendizaje supervisado con respecto a la completitud y validez del conjunto de reglas generado.

Agradecimientos

Este trabajo ha sido financiado por la Consejería de Educación y Ciencia de Castilla-La Mancha a través de los proyectos FEDER PII2I09-0052-3440 y PII1C09-0137-6488.

Referencias

- [1] J. Albusac, J.J. Castro-Schez, L. Lopez-Lopez, D. Vallejo, L. Jimenez-Linares. A supervised learning approach to automate the acquisition of knowledge in surveillance systems. *Journal of Signal Processing*, vol. 89, n°. 12, pp. 2400-2414, 2009.
- [2] J. Albusac, D. Vallejo, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. Intelligent surveillance based on normality analysis to detect abnormal behaviours. *Int. J. of Pattern Recognition and AI*, vol. 23, n°. 7, pp. 1223-1244, 2009.
- [3] L.M. Brown. View independent vehicle-person classification. *Proc. ACM 2nd Int. Workshop on Video Surveillance*, pp. 114-123, 2004.
- [4] J.L. Castro, J.J. Castro-Schez, J.M. Zurita. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems*, vol. 101, n°. 3, pp. 331-342, 1999.
- [5] W. Hu, T. Tan, L. Wang, S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, n°. 3, pp. 334-352, 2004.
- [6] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. Reference Software to Committee Draft JVT-F100 JM15.1, 2008.
- [7] L. Rodriguez-Benitez, J. Moreno, J.J. Castro-Schez, J. Albusac, L. Jimenez. Automatic objects behaviour recognition from compressed video domain. *Image and Vision Computing*, vol. 27, n°. 6, pp. 648-657, 2009.
- [8] C.J. Solana-Cipres, G. Fernandez-Escribano, L. Rodriguez-Benitez, J. Moreno, L. Jimenez. Real-time moving object segmentation in H.264 compressed domain based on approximate reasoning. *International Journal of Approximate Reasoning*, vol. 51, n°. 1, pp. 99-114, 2009.
- [9] M. Valera, S.A. Velastin. Intelligent distributed surveillance systems: a review. *IEE Proc. Vision, Image and Signal Processing*, vol. 152, n°. 2, pp. 192-204, 2005.

LINGUISTIC COMPARISON OF TIME SERIES: A FUZZY APPROACH

Rita Castillo-Ortega Nicolás Marín Daniel Sánchez

Department of Computer Science and A.I.,
University of Granada, 18071, Granada, Spain
{rita,nicm,daniel}@decsai.ugr.es

Abstract

In this paper, we provide a method for the linguistic comparison of time series. The method is based on the linguistic summarization of time series obtained as the difference between the series being compared. Several approaches to compute the difference between the series lead to alternative semantics in the summary. The presented method is applied to an example in the Business Intelligence field.

Keywords: Linguistic Summarization, Time series, Business Intelligence, Fuzzy Logic.

1 INTRODUCTION

Nowadays, it is very important to be able to manage a wide range of information referring to business activities. The process of obtaining information from data is getting more and more important as it allows users to make decision analysis and forecasting [2]. In particular, the research in the Information Systems field has been specially focused on searching for Business Intelligence solutions.

Time dimension usually appears in Information Systems due to its importance in any commercial activity. As an example, dealing with *historical*, current and future predictive views are some of the most important areas within Business Intelligence.

Many authors have focused their research activity on time series Data Mining[1]. As important as the information extraction is the process that permits users a better understanding of this information. In this context, linguistic summarization techniques are of spe-

cial interest because they produce sentences close to natural language to describe data.

One of the most extended ways to face the linguistic summarization problem is through the use of Fuzzy Set Theory. An essential part of this research can be found in R. R. Yager's works, where he uses quantified sentences in the sense of L. A. Zadeh first [15],[16], and OWA operators later [17]. Also following Zadeh's footsteps, we can find J. Kacprzyk et al. in [7], [11], [10], [12], and [9], proposing new quality measures and using the protoform concept. G. Raschia et. al created a model named SaintEtiQ [13], working with hierarchies. From a different point of view, P. Bosc and D. Dubois proposed the use of association rules [3].

In this area, we have focused our efforts in obtaining linguistic summaries that *briefly* present the *essential* information regarding the evolution of a given feature over time. Our proposal [4, 5] is based on the use of *fuzzy quantified statements* and tries to get a brief summary taking advantage of a fuzzy hierarchical partition of a time dimension.

Another problem in this area where linguistic descriptions are of special interest is the comparison of time series. In [6] M. Umamo et al. carry out a study about describing time series data using their global trend and local features. In another related work, J. Kacprzyk and A. Wilbik [8] focus on the evaluation of similarity of time series. They proposed a fuzzy quantifier based aggregation approach and apply their method to the analysis of investment fund quotations in order to show the utility of the technique.

In this paper, a method for the linguistic comparison of time series is presented. The method is based on the application of our linguistic summarization approach to a time series that appropriately describes the difference between the series being compared. We analyze several approaches to compute the difference between the series, leading this way to alternative semantics in the summary.

The paper is organized as follows: Section 2 defines the comparison problem; Section 3 presents the proposed approach to obtain the linguistic comparison. Finally, a practical example appears in Section 4 and some conclusions are presented in Section 5.

2 TIME SERIES COMPARISON ACROSS TIME

The problem we address in this work is, given two time series of data defined over the same time domain, to obtain a linguistic summary that describes the difference in value between the two series across time. As we will see, this linguistic summary is achieved by means of a hierarchical fuzzy partition of the time domain and a fuzzy linguistic granulation of the variable domain. An illustrative example of these elements is depicted in Figure 1.

As we can see, the figure represents the behavior of a given variable V along *time* in two different series. The y-coordinate displays the domain of the variable V as a fuzzy partition. The x-coordinate shows the time dimension hierarchically organized as a set of different fuzzy partitioned levels.

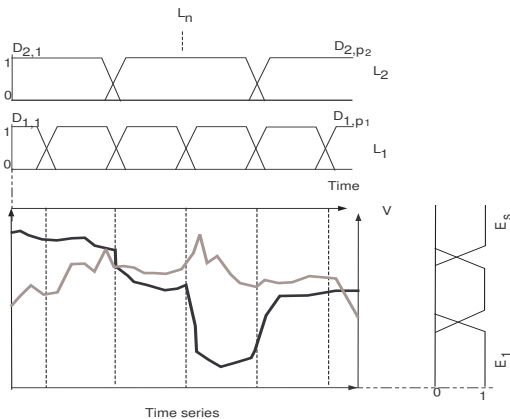


Figure 1: Time series

We assume that the time dimension is described in its finest grained level of granularity by $T = \{t_1, \dots, t_m\}$. Then, we consider a couple of time series defined on this time dimension, namely, TS_1 and TS_2 . $TS_i(t_j)$ represents the value of the variable V under study in t_j as given by TS_i .

As we have previously mentioned:

- The basic domain of variable V is partitioned by a set of linguistic labels $E = \{E_1, \dots, E_s\}$.
- The time dimension is hierarchically organized in n levels, namely, $L = L_1, \dots, L_n$. Each level L_i has

associated a partition $\{D_{i,1}, \dots, D_{i,p_i}\}$ of the basic time domain.

There is no restriction concerning the form of the membership function of a label apart from that it must be normalized. In our approach, we will use trapezoidal functions. When necessary, labels $D_{i,j}$ in time dimension can be the union of a set of trapezoidal functions.

In this work, a set of labels $\{X_1, \dots, X_r\}$ is a partition on X iff:

1. $\forall x \in X, \exists X_i, i \in \{1..r\} | \mu_{X_i}(x) > 0$.
2. $\forall i, j \in \{1..r\}, i \neq j, \text{core}(X_i) \cap \text{core}(X_j) = \emptyset$.

Additionally, concerning the hierarchy of the time dimension, we add the following constraints:

1. $\forall i, j \in \{1..n\}, i < j, p_i > p_j$ (i.e., as we move upward in the hierarchy, the number of labels of the partition decreases).
2. $\forall i \in \{2..n\}, \forall j \in \{1..p_i\}, \forall k \in \{1..p_{i-1}\} | (D_{i,j} \subseteq D_{i-1,k}) \rightarrow (D_{i,j} = D_{i-1,k})$ (i.e., labels cannot generalize another label of an upper level).

3 THE PROPOSED METHOD

In order to face the problem of series comparison, we propose the obtention of a new time series as the difference ΔTS between the two series being compared. Then we can linguistically summarize this new series obtaining information regarding the comparison. These two steps are detailed below.

3.1 Obtaining the ΔTS time series

ΔTS time series can be obtained in several different ways. Let us consider the following definition.

Definition 1 Let TS_1 and TS_2 be two time series defined over the same variable V at a given period of time. Then,

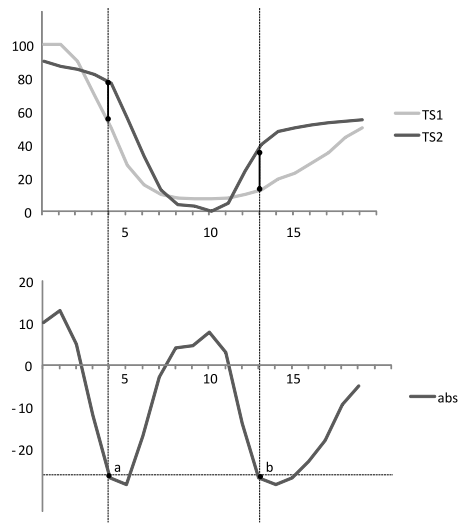
$$\Delta TS_{abs}(t_i) = TS_1(t_i) - TS_2(t_i) \quad (1)$$

$$\Delta TS_{global}(t_i) = \begin{cases} 0, & \text{if } TS_1(t_i) - TS_2(t_i) = 0 \\ \frac{TS_1(t_i) - TS_2(t_i)}{M - m}, & \text{otherwise} \end{cases} \quad (2)$$

$$\Delta TS_{local}(t_i) = \begin{cases} 0, & \text{if } TS_1(t_i) - TS_2(t_i) = 0 \\ \frac{TS_1(t_i) - TS_2(t_i)}{\max(TS_1(t_i), TS_2(t_i)) - m}, & \text{otherwise} \end{cases} \quad (3)$$

where t_i is a specific point in the time domain, M is the global maximum of TS_1 and TS_2 , and m is the global minimum of TS_1 and TS_2 .

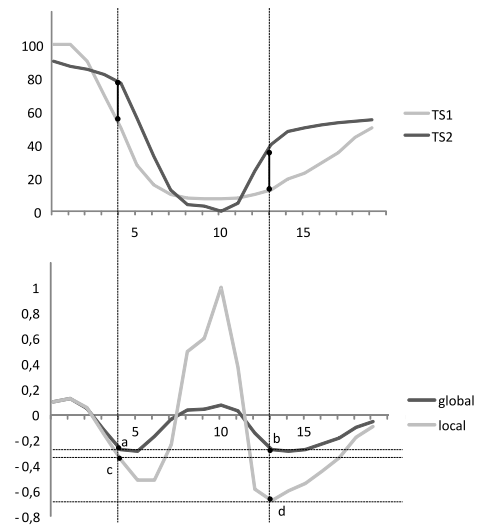
This definition proposes three different ways to face the computation of the new series that describes the difference between the two original ones. At a given time point t_i ,

Figure 2: Time series data TS_1 , TS_2 and ΔTS_{abs}

- $\Delta TS_{abs}(t_i)$ is the difference, in absolute terms, between the two original series at the point t_i (Equation 1).
- $\Delta TS_{global}(t_i)$ is the difference, in relative terms, between the two original series at the point t_i , according to the scale of values of the two original series (i.e., the difference between the global maximum and minimum of the two series) (Equation 2).
- $\Delta TS_{local}(t_i)$ is the difference, also in relative terms, between the two original series at the point t_i , but now according to the scale of values of the given time point in the two original series (i.e., the difference between the maximum value at the given time point and the global minimum) (Equation 3).

The choice of the strategy depends on the necessities of the user or the problem in each particular situation. ΔTS_{abs} is the only option if we are interested in the analysis of the difference between the series in absolute terms. Figure 2 depicts an example of the use of this first alternative. As can be seen, the new series ranges over the same variable domain than the original one.

Nevertheless, if we are interested in the analysis of the difference between the series in relative terms, we have two different alternatives. Figure 3 shows the ΔTS_{global} and ΔTS_{local} obtained for the previous example. The figure illustrates the difference between

Figure 3: ΔTS_{local} and ΔTS_{global}

the two strategies: while in ΔTS_{global} the same difference between the original series produces always the same value in the new one (see points a and b), in ΔTS_{local} the *lower* the original values the greater the *significance* of the difference (see points c and d). In this sense, ΔTS_{abs} behaves like ΔTS_{global} (see Figure 2, points a and b).

Additionally, in contrast to the case of ΔTS_{abs} , the new series (both ΔTS_{global} and ΔTS_{local}) range over $[-1, 1]$.

Once we have opted for a given strategy, we have to provide a linguistic description of the domain of the new series. In our method, we use a linguistic variable with the following features:

- The linguistic variable covers both positive and negatives values.
- In the case of ΔTS_{abs} , it is defined on the range $[-M, M]$.
- In the case of ΔTS_{global} and ΔTS_{local} , it is defined on the range $[-1, 1]$.

As an example, Figure 4 depicts a possible linguistic variable for the interval $[-1, 1]$.

3.2 Obtaining the summary

In the previous subsection, we have analyzed how to obtain a new time series that describes the difference

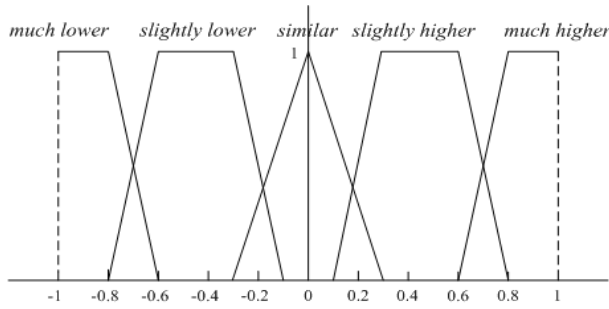


Figure 4: Linguistic labels.

between the two time series under study. Now, we have to obtain an appropriate linguistic summary of this new series.

We consider that the user is interested in linguistic summaries that take the form of a collection of quantified sentences describing the behavior of a time series. We assume that the basic elements of these summaries are the linguistic labels described in Section 2. That is, our approach will deliver a collection of sentences of the form “ Q of $D_{i,j}$ are A ” where:

- $D_{i,j}$ is a label member of a certain level i of the hierarchy associated to the time dimension.
- A is a label or the union of a subset of labels of the partition of the variable V under study (in our case, according to the strategy followed in order to obtain the new series, this partition will be defined on $[-M, +M]$ or $[-1, 1]$).

The user must provide a collection of quantifiers defining the kind of fuzzy quantities and percentages she/he is interested in. This can be defined by choosing among a collection of predefined quantifiers. In this work, we consider that the user provides a *totally ordered* subset $\{Q_1, \dots, Q_{qmax}\}$ of a coherent family of quantifiers Q [14] to be used in the summarization process. In addition, the user will provide a threshold τ for the minimum accomplishment degree he wishes for the quantified sentences comprising the summaries.

Figure 5 represents one of the algorithms we proposed in [5] for obtaining the linguistic summaries. In order to look for brevity of the summary, we start from the time periods in the top level of the hierarchy. Each level has its own quantifier bound ($Qbound_i$) and grouping bound ($Gbound_i$) that, respectively, indicate the less strict quantifier to be considered and the maximum number of labels E_i to be aggregated in a sentence at this level of the time domain. The set $ToSummarize$ is the collection of time periods for which a quantified sentence is missing. If it is possible to obtain an accomplishment degree greater than

-
1. $ToSummarize \leftarrow L_n$;
 2. $Summary \leftarrow \emptyset$; $Summarized \leftarrow \emptyset$;
 3. While $ToSummarize \neq \emptyset$
 - (a) Take $D_{i,j} \in ToSummarize$
 - (b) $ToSummarize \leftarrow ToSummarize \setminus \{D_{i,j}\}$
 - (c) $p \leftarrow qmax$; $covered \leftarrow false$;
 - (d) While $p \geq Qbound_i$ and *not covered*
 - i. $k \leftarrow 1$;
 - ii. While $k \leq Gbound_i$ and *not covered*
 - A. Let $A \leftarrow \argmax_{B \in C_k} GD_{Q_p}(B/D_{i,j})$
 - B. If $GD_{Q_p}(B/D_{i,j}) \geq \tau$ then
$$Summary \leftarrow Summary \cup \{Q_p \text{ of } D_{i,j} \text{ are } A\};$$

$$Summarized \leftarrow Summarized \cup (D_{i,j})$$

$$covered \leftarrow true;$$
 - C. $k \leftarrow k + 1$;
 - iii. $p \leftarrow p - 1$
 - (e) If *not covered* and $i > 1$ then
$$ToSummarize \leftarrow ToSummarize \cup ch(D_{i,j}).$$
 - (f) else if $i = 1$ then
$$Summary \leftarrow Summary \cup \{D_{i,j} \text{ is highly variable}\}$$
-

where $ch(D_{i,j})$ is defined as follows: $ch(D_{i,j}) = \emptyset$ for all j . Otherwise, $ch(D_{i,j}) = \{D_{i-1,k}, k \in \{1..p_{i-1}\} | D_{i-1,k} \cap D_{i,j} \neq \emptyset$ and $\neg \exists D \in ToSummarize \cup Summarized, (D_{i-1,k} \cap D_{i,j}) \subseteq D\}$, and $C_k = \{\cup_{E_h \in F} E_h \mid F \subseteq E, |F| = k\}$.

Figure 5: Algorithm 1 to obtain linguistic summaries.

τ for a certain period using a quantifier Q and a single label, the procedure obtains a summary for that period. If it is not possible, the procedure tries with the union of different subsets of labels: couples, trios, quartets, etc, until we obtain an accomplishment degree greater than τ . The size of the subset is given by k being $Gbound$ its maximum value. When a summary is found in a certain time period we say that the period is *covered*. If all the groups were tried without success, the algorithm repeats the grouping process again, but with a less strict quantifier, until $Qbound_i$ is reached. If no result is found for a given period $D_{i,j}$, we try to obtain such sentences with the *corresponding children* $ch(D_{i,j})$ in the next level. If $ch(D_{i,j}) = \emptyset$, then a sentence indicating the observed variability is added to the summary ($D_{i,j}$ is highly variable¹). The final set of linguistically quantified sentences comprising the summary is $Summary$.

The second algorithm proposed in [5] is similar to that in Figure 5 with the difference that when it is not possible to obtain an accomplishment degree greater than τ for a certain period using a quantifier Q and a single label, the procedure tries with less strict quantifiers before trying with aggregations of labels in E . In

¹Though variability is also analyzed in [9], we refer to this characteristic of the series only when we cannot suitably summarize the values for a given time period.

summary, the first approach tries to obtain summaries with more restrictive quantifiers, while the second one tries to obtain summaries with smaller groups of labels. The obtained results will depend on the family of quantifiers, the threshold τ , the bounds pointed by the user at the different levels of the time hierarchy, and, of course, the data series.

When the summary is obtained, the set of sentences is postprocessed in order to produce an easier to read paragraph. The repetitions can be removed via merging sentences that cover different time periods but produce the same trend. The merging process takes into account sentences with the same quantifier and trend as well as sentences with the same quantifier. For example, if the summary is *In A, the difference was mostly low or very low. In B, the difference was mostly low or very low*, the function produces the sentence *In A and B, the difference was mostly low or very low*, which is shorter and clearer.

4 A PRACTICAL CASE

In this section we will work with a representative example in order to apply our process and clarify concepts or procedures introduced in former sections. Figure 6 contains the time series that shows the patient inflow on two different medical centers during a given year.

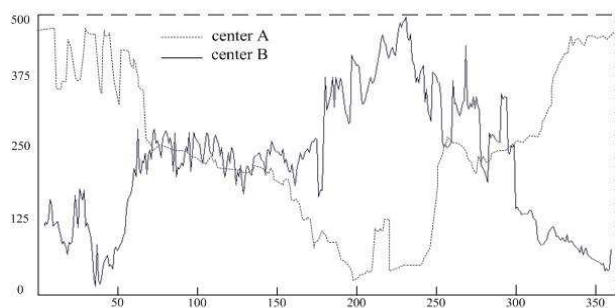


Figure 6: Patient inflow in two medical centers.

Using equations 2 and 3 we obtain the time series data depicted in Figure 7, both relative differences. The y-coordinate represents the differences in terms of Figure 4, and the x-coordinate represents the time dimension. As we can see, the time dimension is hierarchically organized following a meteorological criteria thanks to three fuzzy partition of the time domain, namely: one based on approximate months (in order to avoid a strong dependence of the obtained summaries with respect to the crisp boundaries of conventional months) and two others with different levels of granularity. Fuzziness is specially useful in these two

last partitions because transitions between periods are clearly fuzzy.

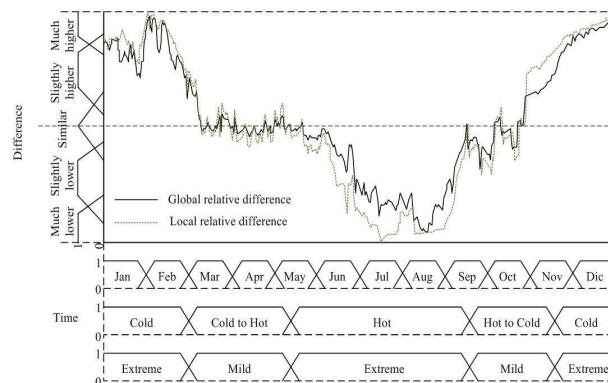


Figure 7: Relative differences.

The example is carried out using a subset of trapezoidal quantifiers $Q = \{Q_1 = (0, 0.4, 0.6, 1), Q_2 = (0, 0.6, 0.8, 1), Q_3 = (0, 0.7, 0.9, 1)\}$ (that we have called *at least half of*, *at least 70% of*, and *most of*, respectively) and a threshold $\tau = 0.7$ and values $Qbound_i = Gbound_i = 2 \forall i$ as parameters.

For instance, we will use algorithm 1 (Figure 5) with the time data taken from the *local* relative difference (ΔTS_{local}) between patient inflow in centers A and B. The set of quantified sentences obtained is:

- At least 70% of the time, the patient inflow with cold weather is much higher in center A than in center B
- At least 70% of the time, the patient inflow with hot weather is slightly lower or much lower in center A than in center B
- At least 70% of the time, the patient inflow in March is slightly higher or similar in center A than in center B
- The patient inflow difference between center A and center B in April presents variability
- The patient inflow difference between center A and center B in May presents variability
- The patient inflow difference between center A and center B in September presents variability
- At least 70% of the time, the patient inflow in October is similar or slightly lower in center A than in center B
- Most of the time, the patient inflow in November is much higher or slightly higher in center A than in center B

After the postprocessing, we will have

Most of the time, the patient inflow in November is much higher or slightly higher in center A than center B. At least 70% of the time, the patient inflow with cold weather is much higher in center A than center B, with hot weather is slightly lower or much lower in center A than center B, and in March is slightly higher or similar in center A than center B. Finally, the patient inflow difference in April,

May and September presents variability.

5 CONCLUSIONS

We have presented an approach to obtain a particular case of linguistic comparison of two time series, as a summary of the difference in each point between both series. This technique is relevant as a Business Intelligence tool for decision making, since it provides an easy-to-understand information to the decision maker. It is worth noticing that understandability and brevity are improved by using user-defined hierarchical fuzzy partition of time dimension and user-defined quantifiers. As future work we plan to perform linguistic comparison on the basis of other features, and to apply new summarization algorithms that are currently being developed.

Acknowledgements

The research reported in this paper was partially supported by the Andalusian Government (Junta de Andalucía) under project P07-TIC03175 and also by the Spanish Government (Science and Innovation Department) under project TIN2009-08296.

References

- [1] I.Z. Batyrshin and L. Sheremetov. Perception-based approach to time series data mining. *Appl. Soft Comput.*, 8(3):1211–1221, 2008.
- [2] I.Z. Batyrshin and T. Sudkamp. Perception based data mining and decision support systems. *International Journal of Approximate Reasoning*, 48(1):1–3, 2008.
- [3] P. Bosc, D. Dubois, O. Pivert, H. Prade, and M. De Calmes. Fuzzy summarization of data using fuzzy cardinalities. In *Int. Conf. Inf. Process. Manag. Uncertainty Knowl. Based Syst.*, pages 1553–1559, 2002.
- [4] R. Castillo-Ortega, N. Marín, and D. Sánchez. Fuzzy quantification-based linguistic summaries in data cubes with hierarchical fuzzy partition of time dimension. In H. Yin and E. Corchado, editors, *IDEAL'09*, volume 5788 of *LNCS*, pages 578–585. Springer, Heidelberg, 2009.
- [5] R. Castillo-Ortega, N. Marín, and D. Sánchez. Linguistic summary-based query answering on data cubes with time dimension. In T. Andreasen et al., editor, *FQAS'09*, volume 5822 of *LNAI*, pages 560–571. Springer, Heidelberg, 2009.
- [6] M. Humano, M. Okamura, and K. Seta. Improved method for linguistic expression of time series with global trend and local features. In *FUZZ-IEEE 2009*, pages 1169–1174, 2009.
- [7] J. Kacprzyk. Fuzzy logic for linguistic summarization of databases. In *IEEE International Fuzzy Systems Conference*, pages 813–818, 1999.
- [8] J. Kacprzyk and A. Wilbik. Using fuzzy linguistic summaries for the comparison of time series: an application to the analysis of investment fund quotations. In U. Kaymak J. P. Carvalho, D. Dubois and J. M. C. Sousa, editors, *IFSA-EUSFLAT 2009*, pages 1321–1326, 2009.
- [9] J. Kacprzyk, A. Wilbik, and S. Zadrozny. Linguistic summarization of time series using a fuzzy quantifier driven aggregation. *Fuzzy Sets and Systems*, 159(12):1485–1499, 2008.
- [10] J. Kacprzyk and R. R. Yager. Linguistic summaries of data using fuzzy logic. In *International Journal of General Systems*, volume 30, pages 133–154, 2001.
- [11] J. Kacprzyk, R. R. Yager, and S. Zadrozny. A fuzzy logic based approach to linguistic summaries in databases. *International Journal of Applied Mathematical Computer Science*, 10:813–834, 2000.
- [12] J. Kacprzyk and S. Zadrozny. Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools. *Inf. Sci. Inf. Comput. Sci.*, 173(4):281–304, 2005.
- [13] G. Raschia and N. Mouaddib. Saintetiq: a fuzzy set-based approach to database summarization. *Fuzzy Sets Syst.*, 129(2):137–162, 2002.
- [14] M. A. Vila, J. C. Cubero, J. M. Medina, and O. Pons. The generalized selection: an alternative way for the quotient operations in fuzzy relational databases. In B. Bouchon-Meunier, R. Yager, and L. Zadeh, editors, *Fuzzy Logic and Soft Computing*. World Scientific Press, 1995.
- [15] R. R. Yager. A new approach to the summarization of data. *Information Sciences*, (28):69–86, 1982.
- [16] R. R. Yager. Toward a language for specifying summarizing statistics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(2):177–187, 2003.
- [17] R. R. Yager. A human directed approach for data summarization. In *IEEE International Conference on Fuzzy Systems*, pages 707–712, 2006.

ON THE SATISFACTION OF THE FUNCTIONAL EQUATION $A(x, N(x)) = c$

Ana Pradera

Dept. de Ciencias de la Computación
Universidad Rey Juan Carlos, Móstoles, Madrid
ana.pradera@urjc.es

Abstract

With the aim of analyzing the behavior of aggregation functions when receiving contradictory information, this paper studies the satisfaction of the functional equation $A(x, N(x)) = c$, where A is an aggregation function on $[0, 1]$, N is a strong negation and c is a constant value belonging to $[0, 1]$. Special attention is paid to the values $c \in \{0, 1, x_N, e, a\}$, where x_N is the fixed point of the negation and e/a denote, respectively, neutral/absorbing elements.

Keywords: Aggregation functions, strong negations, contradictory information.

1 INTRODUCTION

Due to the key role they play in many practical applications, the interest in *aggregation functions*, which perform the combination of several inputs into a single output, is unceasingly growing (see e.g. the recent monographs on the topic [4, 12, 2, 6]). Large families of aggregation functions, as well as different construction methods, are available, so an important -and difficult- issue is how to choose the most appropriate function for a given application. Several criteria may help in making this choice, such as the satisfaction of some specific properties, the fitting to some empirical data or the fulfillment of a given optimization condition.

In some contexts, the behavior of the aggregation function when receiving *contradictory information* could be a useful criterion. For instance, depending on the application, one may be expecting a low, a high or a medium score when aggregating couples of contradictory information, thus seeking an aggregation function

presenting, respectively, an intolerant, a tolerant or a compensative attitude towards contradictory inputs.

This paper proposes a preliminary approach to the above problem, representing contradictory information by means of couples $(x, N(x))$, where N is a negation function, and then studying the satisfaction of the functional equation $A(x, N(x)) = c$ for some distinguished values $c \in [0, 1]$. It is organized as follows. Section 2 briefly recalls the main issues regarding aggregation and negation functions that are needed later on. Section 3 includes the main results of the paper. It first provides some general statements about the fulfillment of the equation $A(x, N(x)) = c$, and then addresses the following distinguished cases: $c = 0$ and $c = 1$, accounting, respectively, for strict intolerant and tolerant behaviors; the case $c = x_N$, where x_N is the negation's fixed point, presenting a compensative behavior; and then the neutral and absorbing behaviors, represented, respectively, by $c = e$ and $c = a$, where e and a stand for neutral and absorbing elements. The paper ends with a summary of the obtained results.

2 PRELIMINARIES

Although *aggregation functions* are defined for inputs of any size, in this paper these terms will be used to refer to *bivariate* aggregation functions, i.e., non-decreasing functions $A : [0, 1]^2 \rightarrow [0, 1]$ verifying the boundary conditions $A(0, 0) = 0$ and $A(1, 1) = 1$. Aggregation functions may be compared pointwise as follows: given two functions A_1 and A_2 , it is said that A_1 is *weaker* than A_2 (or A_2 is *stronger* than A_1), and it is denoted $A_1 \leq A_2$, when it is $A_1(x, y) \leq A_2(x, y)$ for any $x, y \in [0, 1]$. The relation $\leq$, along with the distinguished functions *Min* (minimum) and *Max* (maximum), allows one to classify aggregation functions into the following four categories:

- *Conjunctive* functions, which are those verifying $A \leq \text{Min}$. This class includes the well-known *triangular norms* (*t-norms*) as well as *copulas*.
- *Disjunctive* functions, verifying $\text{Max} \leq A$, such as *triangular conorms* (*t-conorms*) and *dual copulas*.
- *Averaging* functions, which verify $\text{Min} \leq A \leq \text{Max}$, and include, in particular, different functions based on the arithmetic mean, such as *quasi-linear means* or *OWA operators*, as well as integral-based aggregations.
- Finally, the class of *mixed* aggregation functions contains all the operators that do not belong to any of the three previous categories, such as *uni-norms*, *nullnorms*, *compensatory T-S operators* or *symmetric sums*.

Recall also that an aggregation function A has a *neutral element* $e \in [0, 1]$ whenever $A(x, e) = A(e, x) = x$ for any $x \in [0, 1]$, and an *absorbing element* $a \in [0, 1]$ when $A(a, x) = A(x, a) = a$ for any $x \in [0, 1]$.

On the other hand, contradictory information will be represented with the help of *strong negations* ([13]). A strong negation is a non-increasing function $N : [0, 1] \rightarrow [0, 1]$ which is involutive, that is, verifies $N(N(x)) = x$ for any $x \in [0, 1]$. Recall that a function $N : [0, 1] \rightarrow [0, 1]$ is a strong negation if and only if there exists a strictly increasing bijection $\varphi : [0, 1] \rightarrow [0, 1]$ such that $N = N_\varphi$, where $N_\varphi(x) = \varphi^{-1}(1 - \varphi(x))$ for any $x \in [0, 1]$. Due to their definition, strong negations are continuous and strictly decreasing functions, they satisfy the boundary conditions $N(0) = 1$ and $N(1) = 0$, and they have a unique fixed point $x_N = \varphi^{-1}(\frac{1}{2}) \in]0, 1[$, verifying $N(x_N) = x_N$. The set made of all the strong negations defined on $[0, 1]$ will be denoted with $\mathcal{N}_{[0,1]}$.

3 MAIN RESULTS

As already stated in the Introduction, the aim of this paper is to study which aggregation functions satisfy the equation

$$\forall x \in]0, 1[, \quad A(x, N(x)) = c \quad (1)$$

for a given strong negation N and a constant value $c \in [0, 1]$ (the points $(0, 1)$ and $(1, 0)$ are excluded because of their singularity). Note first of all that the weakest and the strongest aggregation functions satisfying (1) are, respectively,

$$A(x, y) = \begin{cases} 0 & \text{if } y < N(x), \\ 1 & \text{if } x = y = 1, \\ c & \text{otherwise.} \end{cases}$$

and

$$A(x, y) = \begin{cases} 0 & \text{if } x = y = 0, \\ c & \text{if } y \leq N(x), \\ 1 & \text{otherwise.} \end{cases}$$

It is also easy to prove that given a strong negation N , the class made of all the aggregation functions satisfying equation (1) is closed under composition by means of any aggregation function having c as idempotent element, i.e., if A_1, A_2 are two aggregation functions satisfying equation (1), then $A(A_1, A_2)$, defined as $A(A_1, A_2)(x, y) = A(A_1(x, y), A_2(x, y))$ for any $x, y \in [0, 1]$, does also satisfy the equation as long as $A(c, c) = c$. When $c \in \{0, 1\}$ any aggregation function may be used as the outer function A ; otherwise, A may be chosen, for example, among averaging functions.

Observe on the other hand that equation (1) refers to one specific strong negation, whereas situations may arise in which one is interested in aggregation functions satisfying the equation for *any* strong negation, i.e., independently of the particular negation that is used to represent contradictory information. This consideration suggests the following stronger version of (1):

$$\forall x \in]0, 1[, \quad \forall N \in \mathcal{N}_{[0,1]}, \quad A(x, N(x)) = c \quad (2)$$

The above equation is evidently much more restrictive than the former one. Averaging functions are to be discarded, and the weakest and the strongest aggregation functions satisfying equation (2) are, respectively

$$A(x, y) = \begin{cases} 0 & \text{if } x = 0 \text{ or } y = 0, \\ 1 & \text{if } x = 1 \text{ and } y = 1, \\ c & \text{otherwise.} \end{cases}$$

and

$$A(x, y) = \begin{cases} 0 & \text{if } x = 0 \text{ and } y = 0, \\ 1 & \text{if } x = 1 \text{ or } y = 1, \\ c & \text{otherwise.} \end{cases}$$

The next subsections are devoted to the following particular cases:

- $c = 0$ and $c = 1$, which reflect, respectively, strict intolerance or large tolerance with respect to contradictory information.
- $c = x_N$, where x_N is the negation's fixed point. Since it is always $\text{Min}(x, N(x)) \leq x_N \leq \text{Max}(x, N(x))$, this case provides a compensative behavior.
- $c = e$ and $c = a$, when dealing, respectively, with aggregation functions with a neutral element $e \in]0, 1[$ or an absorbing element $a \in]0, 1[$.

3.1 THE CASES $c = 0$ and $c = 1$

Equations $A(x, N(x)) = 0$ and $A(x, N(x)) = 1$ were analyzed in [11] while studying the satisfaction of the logical Non-Contradiction and Excluded-Middle principles. The main results regarding the former (the approach to the latter is, thanks to duality, similar) are summarized below:

Proposition 1 [11] *Equation (1) with $c = 0$ is satisfied if and only if for any $x, y \in [0, 1]$ it is:*

$$A(x, y) = \begin{cases} 0, & \text{if } y \leq N(x), \{x, y\} \neq \{0, 1\} \\ B(x, y), & \text{otherwise} \end{cases}$$

where B is any bivariate non-decreasing function verifying $B(1, 1) = 1$.

Clearly the above characterization provides either conjunctive aggregation functions (when B is chosen such that $B \leq \text{Min}$) or mixed ones (otherwise). Among the former, one may find different triangular norms, such as the drastic t-norm, the nilpotent minimum or some t-norms isomorphic to the Łukasiewicz t-norm. Regarding mixed functions satisfying equation (1) with $c = 0$, some instances may be found, for example, within the class of compensatory T-S functions (see [11] for details).

Concerning equation (2) with $c = 0$, the following characterization is available:

Proposition 2 [11] *Equation (2) with $c = 0$ is satisfied if and only if for any $x, y \in [0, 1]$ it is:*

$$A(x, y) = \begin{cases} 0, & \text{if } (x, y) \in [0, 1]^2 \\ B(x, y), & \text{otherwise} \end{cases}$$

where B is any bivariate non-decreasing function verifying $B(1, 1) = 1$.

Aggregation functions characterized in Proposition 2 may only be conjunctive (such as the drastic t-norm, which is the unique t-norm satisfying (2) with $c = 0$) or mixed (e.g. the exponential convex combination of the drastic t-norm and any t-conorm, see [11] for details).

3.2 THE CASE $c = x_N$

The main results associated to the case $c = x_N$, which accounts for a compensative behavior, are the following:

Proposition 3 *Let A be an aggregation function and let N be a strong negation with fixed point x_N .*

1. *If A is either conjunctive or disjunctive, then it does not satisfy equation (1) with $c = x_N$.*

2. *If A has neutral element $e \in]0, 1[$, then A satisfies (1) with $c = x_N$ if and only if it satisfies (1) with $c = e$ (and in such a case it must be $x_N = e$).*
3. *If A has absorbing element $a \in]0, 1[$, then A satisfies (1) with $c = x_N$ if and only if $x_N = a$.*
4. *If A is commutative and N -self-dual (i.e., $A = N \circ A \circ N \times N$) then A satisfies (1) with $c = x_N$.*
5. *A does not satisfy equation (2) with $c = x_N$.*

Proof. The first statement comes from the inequality $\text{Min}(x, N(x)) \leq x_N \leq \text{Max}(x, N(x))$ for all $x \in [0, 1]$. The second one is obtained choosing $x = e$ and taking into account that N is an involution with a unique fixed point. To prove the third one it suffices to choose $x = a$ and to notice that, because of monotonicity, any aggregation function with absorbing element a is such that $A(x, y) = a$ whenever $(x, y) \in [0, a] \times [a, 1] \cup [a, 1] \times [0, a]$. The involutive character of N provides the fourth statement. Finally, to prove the last statement take $(u, v) \in]0, 1[^2$ such that $u \neq v$ and choose two strong negations N_1 and N_2 such that $x_{N_1} \neq x_{N_2}$ and $v = N_1(u) = N_2(u)$; then the fulfillment of equation (2) with $c = x_N$ would entail the contradiction $x_{N_1} = x_{N_2}$. ■

The particular cases where A has a neutral or an absorbing element will be treated in the next subsections. For the general case, the condition given by the fourth item of Proposition 3 is very useful for finding both averaging and mixed aggregation functions satisfying $A(x, N(x)) = x_N$ for any $x \in]0, 1[$, since different families and construction methods for N -self-dual aggregation functions are available (see e.g. [9]). For instance:

Proposition 4 *Given a strong negation $N = N_\varphi$, the quasi-arithmetic mean generated by φ , given by $M_\varphi(x, y) = \varphi^{-1}\left(\frac{\varphi(x) + \varphi(y)}{2}\right)$, is an averaging function satisfying $A(x, N(x)) = x_N$ for any $x \in [0, 1]$.*

Proposition 5 *Given a strong negation $N = N_\varphi$ and an arbitrary commutative aggregation function B , the two following constructions provide aggregation functions verifying $A(x, N(x)) = x_N$ for any $x \in [0, 1]$ (the first one with convention $\frac{0}{0} = \frac{1}{2}$):*

$$A(x, y) = \varphi^{-1} \left(\frac{B(x, y)}{B(x, y) + B(N(x), N(y))} \right)$$

$$A(x, y) = \varphi^{-1} \left(\frac{B(x, y) + 1 - B(N(x), N(y))}{2} \right)$$

3.3 THE CASE $c = e$, $e \neq 0, 1$

As proved in Proposition 3, when dealing with aggregation functions with a neutral element $e \in]0, 1[$ it appears that $A(x, N(x)) = e$ for all $x \in]0, 1[$ is equivalent to $A(x, N(x)) = x_N$ for all $x \in]0, 1[$. Hence, the results given in Proposition 3 also apply here: only averaging and mixed aggregation functions with neutral element $e = x_N$ can be considered, and a wide family of functions in this class satisfying $A(x, N(x)) = e$ for all $x \in]0, 1[$ is given by those which are commutative and N -self-dual.

Remark 6 Note that the satisfaction of $A(x, N(x)) = e$ for any $x \in]0, 1[$ is equivalent to the fact that all the tuples $(x, N(x))$ are neutral tuples (see [1]) for the aggregation function A .

Within the class of averaging functions, the two following functions are, respectively, the weakest and the strongest satisfying $A(x, N(x)) = e$ for any $x \in]0, 1[$:

$$A(x, y) = \begin{cases} \text{Min}(x, y) & \text{if } y < N(x), \\ \text{Max}(x, y) & \text{if } x, y \geq e, \\ e & \text{otherwise.} \end{cases}$$

and

$$A(x, y) = \begin{cases} \text{Min}(x, y) & \text{if } x, y \leq e, \\ \text{Max}(x, y) & \text{if } y > N(x), \\ e & \text{otherwise.} \end{cases}$$

The next proposition provides a means for building commutative N -self-dual aggregation functions with a neutral element $e = x_N$ that, as stated in Proposition 3, satisfy $A(x, N(x)) = e$:

Proposition 7 Given a strong negation $N = N_\varphi$ and a commutative aggregation function B with neutral element x_N , the following construction provides aggregation functions with neutral element x_N satisfying $A(x, N(x)) = x_N$ for any $x \in [0, 1]$:

$$A(x, y) = \varphi^{-1} \left(\frac{\varphi(B(x, y)) + 1 - \varphi(B(N(x), N(y)))}{2} \right)$$

The most popular aggregation functions with a neutral element $e \in]0, 1[$ are those which are associative and commutative, known as *uninorms* ([14]). Are there uninorms satisfying $A(x, N(x)) = e$ for all $x \in]0, 1[$? The next proposition shows that, among the most important classes of uninorms, only one satisfies the above equation.

Proposition 8 Let $U : [0, 1]^2 \rightarrow [0, 1]$ be a uninorm and let N be a strong negation.

1. If U is locally internal on $[0, e] \times [e, 1] \cup [e, 1] \times [0, e]$ (i.e., it is $U(x, y) \in \{x, y\}$ for any (x, y) in that region), then it does not satisfy $U(x, N(x)) = e$ for all $x \in]0, 1[$.
2. If U is continuous on $]0, 1[^2$, then it satisfies $U(x, N(x)) = e$ for all $x \in]0, 1[$ if and only if U is a representable uninorm with generating function u and $N = N_u$, where $N_u = u^{-1} \circ (-u)$.

Proof. Note first that $e = x_N$ is a necessary condition for the satisfaction of $U(x, N(x)) = e$ for all $x \in]0, 1[$ (choosing $x = e$ the equation provides $N(e) = e$, which entails $e = x_N$ due to the uniqueness of the strong negation's fixed point). This means that any couple $(x, N(x))$ belongs to the region $[0, e] \times [e, 1] \cup [e, 1] \times [0, e]$. Then:

1. If U is locally internal, $U(x, N(x)) \in \{x, N(x)\}$, and hence $U(x, N(x)) \neq e$ for any $x \neq e$.
2. The structure of $]0, 1[$ -continuous uninorms given by Hu and Li in [7] proves that, unless U is a representable uninorm, there is always a value $0 < k < e$ such that the output of U on the region $[0, k] \times [k, 1] \cup [k, 1] \times [0, k]$ coincides either with the minimum or with the maximum, and is, hence, different from e . Now, if U is a representable uninorm, then it is well-known that it may be written (see e.g. [5]) as $U(x, y) = u^{-1}(u(x) + u(y))$ for any $(x, y) \in [0, 1]^2 \setminus \{(0, 1), (1, 0)\}$, where $u : [0, 1] \rightarrow [-\infty, +\infty]$ is a strictly increasing bijection such that $u(e) = 0$. Then the equation $U(x, N(x)) = e$ is clearly equivalent to $N = N_u$.

■

Remark 9 Note that the class described in the first item of Proposition 8 includes well-known uninorms, such as idempotent uninorms or uninorms in the classes $\mathcal{U}_{\min}$ and $\mathcal{U}_{\max}$ (those obtained when choosing, respectively, Min and Max in the region $[0, e] \times [e, 1] \cup [e, 1] \times [0, e]$). Observe in addition that the result regarding representable uninorms (Proposition 8, item 2) could have been obtained, alternatively, from the fourth item in Proposition 3, since representable uninorms are self-dual, except at the points $(0, 1)$ and $(1, 0)$, with respect to their associated negation N_u (see e.g. [5]).

3.4 THE CASE $c = a$, $a \neq 0, 1$

When dealing with aggregation functions possessing an absorbing element $a \in]0, 1[$, both conjunctive and disjunctive functions are to be discarded (since they have absorbing elements 0 and 1, respectively) and the following may be stated:

Proposition 10 *Let A be an aggregation function with absorbing element $a \in]0, 1[$ and let N be a strong negation with fixed point x_N .*

1. *If $a = x_N$, A satisfies equation (1) with $c = a$.*
2. *A satisfies equation (2) with $c = a$ if and only if $A(x, y) = a \quad \forall (x, y) \in]0, 1[^2$.*

Proof. The first item is a consequence of item 3 in Proposition 3; the second one is a matter of calculation. ■

Let us focuss on the class of *nullnorms* ([10, 3]), the most important aggregation functions with an absorbing element. Recall that nullnorms are aggregation functions $V : [0, 1]^2 \rightarrow [0, 1]$ which are associative, commutative and possess an element $a \in]0, 1[$ such that for any $x \in [0, 1]$, it is $V(x, 0) = x$ if $x \leq a$ and $V(x, 1) = x$ if $x \geq a$. Nullnorms have absorbing element a , they behave as a t-conorm in $[0, a]^2$ and as a t-norm in $[a, 1]^2$, and verify $V(x, y) = a$ otherwise. They have the following structure (where T is a t-norm and S is a t-conorm):

$$V_{a,T,S}(x, y) = \begin{cases} a \cdot S(\frac{x}{a}, \frac{y}{a}), & \text{if } x, y \leq a \\ a + (1-a) \cdot T(\frac{x-a}{1-a}, \frac{y-a}{1-a}), & \text{if } x, y \geq a \\ a, & \text{otherwise} \end{cases}$$

Proposition 11 *Let $V_{a,T,S}$ be a nullnorm and let N be a strong negation with fixed point x_N .*

1. *$V_{a,T,S}$ satisfies equation (1) with $c = a$ if and only if one of the following conditions holds:*
 - (a) $x_N = a$
 - (b) $x_N < a$ and $S(u, v) = 1 \quad \forall u \geq \frac{N(a)}{a}, \forall v \geq \frac{N(ua)}{a}$.
 - (c) $x_N > a$ and $T(u, v) = 0 \quad \forall u \leq \frac{N(a)-a}{1-a}, \forall v \leq \frac{N(a+(1-a)u)-a}{1-a}$.
2. *$V_{a,T,S}$ satisfies equation (2) with $c = a$ if and only if T and S are, respectively, the drastic t-norm and the drastic t-conorm.*

Remark 12 *Note that cases (b) and (c) in Proposition 11 pose the problem of the N_α -annihilation of t-conorms and t-norms, a generalization of the N -annihilation problem (see e.g. [8]). Indeed, focussing on (c) (the problem for t-conorms is dual), and denoting $\alpha = \frac{N(a)-a}{1-a}$ and $N_\alpha(u) = \frac{N(a+(1-a)u)-a}{1-a}$, condition (c) may be equivalently written as*

$$x_N > a \text{ and } T(u, v) = 0 \quad \forall u \leq \alpha, \forall v \leq N_\alpha(u)$$

where $N_\alpha : [0, \alpha] \rightarrow [0, \alpha]$ is a strong negation on $[0, \alpha]$.

4 CONCLUSIONS

This paper has proposed a simple preliminary approach for studying the behavior of bivariate aggregation functions when receiving contradictory information. The existence of aggregation functions satisfying equations (1) and (2) has been investigated, taking into account the four different categories in which aggregation functions may be classified, as well as the most important distinguished values for the constant c (see tables 1 and 2 for a summary). Whenever possible, concrete examples of aggregation functions satisfying equations (1) and (2) have been provided.

Table 1: Existence of aggregation functions satisfying $A(x, N(x)) = c \quad \forall x \in]0, 1[$

	Conj.	Disj.	Aver.	Mix.
$c = 0$	✓	×	×	✓
$c = 1$	×	✓	×	✓
$c = x_N$	×	×	✓ (any if $x_N = a$)	
$c = e$ $\neq 0, 1$	×	×	✓ (only if $e = x_N$)	
$c = a$ $\neq 0, 1$	×	×	✓ (any if $a = x_N$)	

Table 2: Existence of aggregation functions satisfying $A(x, N(x)) = c \quad \forall x \in]0, 1[, \forall N \in \mathcal{N}_{[0,1]}$

	Conj.	Disj.	Aver.	Mix.
$c = 0$	✓	×	×	✓
$c = 1$	×	✓	×	✓
$c = x_N$	×	×	×	×
$c = e$ $\neq 0, 1$	×	×	×	×
$c = a$ $\neq 0, 1$	×	×	×	✓

References

- [1] G. Beliakov, T. Calvo, and A. Pradera. Handling of neutral information by aggregation operators. *Fuzzy Sets and Systems*, 158(8):861–880, 2007.
- [2] G. Beliakov, A. Pradera, and T. Calvo. *Aggregation Functions: A Guide for Practitioners*. Springer, 2007.
- [3] T. Calvo, B. De Baets, and J. Fodor. The functional equations of Frank and Alsina for uninorms and nullnorms. *Fuzzy Sets and Systems*, 120:385–394, 2001.

- [4] T. Calvo, A. Kolesárová, M. Komorníková, and R. Mesiar. Aggregation operators: properties, classes and construction methods. In T. Calvo, G. Mayor, and R. Mesiar, editors, *Aggregation Operators. New Trends and Applications*, pages 3–104. Physica-Verlag, Heidelberg, New York, 2002.
- [5] J. Fodor, R. Yager, and A. Rybalov. Structure of uninorms. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 5:411–427, 1997.
- [6] M. Grabisch, J.L. Marichal, R. Mesiar, and E. Pap. *Aggregation Functions*. Cambridge University Press, 2009.
- [7] S.k. Hu and Z.f. Li. The structure of continuous uni-norms. *Fuzzy Sets and Systems*, 124(1):43–52, 2001.
- [8] S. Jenei. How to construct left-continuous triangular norms-state of the art. *Fuzzy Sets and Systems*, 143:27–45, 2004.
- [9] K. C. Maes, S. Saminger, and B. De Baets. Representation and construction of self-dual aggregation operators. *Europ. J. Oper. Res.*, 177:472–487, 2007.
- [10] M. Mas, G. Mayor, and J. Torrens. T-operators. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 7(1):31–50, 1999.
- [11] A. Pradera and E. Trillas. Aggregation, Non-Contradiction and Excluded-Middle. *Mathware & Soft Computing*, XIII:189–201, 2006.
- [12] V. Torra and Y. Narukawa. *Modeling Decisions: Information Fusion and Aggregation Operators*. Springer, 2007.
- [13] E. Trillas. Sobre funciones de negación en la teoría de los subconjuntos difusos. *Stochastica*, III:47–59 (in Spanish) Reprinted (English version) in *Advances of Fuzzy Logic*, S. Barro et al. (eds), Universidad de Santiago de Compostela 31–43, 1998., 1979.
- [14] R. Yager and A. Rybalov. Uninorm aggregation operators. *Fuzzy Sets and Systems*, 80:111–120, 1996.

UNA GENERALIZACION DE LA MIGRATIVIDAD

Humberto Bustince¹ Javier Fernandez¹ Bernard De Baets² Javier Montero³ Radko Mesiar⁴

¹ Dep. Automática y Computación, Universidad Pública de Navarra, {bustince, fcojavier.fernandez}@unavarra.es

² Dep. of Applied Mathematics, Biometrics and Process Control, Ghent University, bernard.debaets@ugent.be

³ Facultad de Matemáticas, Universidad Complutense, monty@mat.ucm.es

⁴ Dep. of Mathematics and Descriptive Geometry, Slovak University of Technology, mesiar@math.sk

Resumen

En este trabajo presentamos una generalización de los conceptos de función α -migrativa y función migrativa. En particular, nos centramos en la relación de estas funciones migrativas generalizadas con algunas propiedades habitualmente requeridas a funciones de agregación, como puede ser la existencia de elemento neutro.

Palabras Clave: Migratividad; α -migratividad; Función de agregación.

Posteriormente, en [3] los autores realizan un completo estudio de la llamada migratividad generalizada, es decir, la propiedad migrativa cuando el parámetro puede tomar cualquier valor $\alpha \in [0, 1]$. Por otro lado, también se ha considerado la propiedad migrativa para ejemplos específicos de funciones de agregación ([2]).

En este trabajo vamos un paso más allá y presentamos los conceptos de α - B -migratividad y B -migratividad. Es decir, consideramos funciones invariantes en cierto sentido frente a reducciones en una u otra variable, pero donde la reducción o escalamiento no viene dado exclusivamente por un parámetro α , sino también por una función de agregación B . En particular, esta generalización engloba la α -migratividad y la migratividad generalizada sin más que tomar como función B el producto.

El esquema de este trabajo es el siguiente. En la próxima sección presentamos algunos conceptos básicos para este trabajo. En la Sección 3 presentamos algunos resultados sobre funciones α - B -migrativas, centrándonos en particular en las consecuencias de la existencia de elemento neutro. La Sección 4 está dedicada al estudio de algunos resultados sobre funciones B -migrativas. En la Sección 5 mostramos algunos resultados cuando consideramos casos específicos de funciones de agregación, como las t -normas o las medias cuasiaritméticas. Finalmente, en la Sección 6 mostramos algunas conclusiones y líneas de trabajo futuras.

1 INTRODUCCION

La propiedad migrativa expresa el hecho de que si reducimos en una determinada proporción (dada por un parámetro α) una de las variables de una función de agregación (de al menos dos variables), el resultado es independiente de cuál de las variables es la reducida. Desde el punto de vista aplicado, esta propiedad es muy interesante siempre que es necesario agregar informaciones parciales procedentes con diferentes orígenes temporales y/o espaciales. Este es el caso, por ejemplo en toma de decisiones ([7, 8, 9]). También en procesamiento de imagen resulta útil, pues permite expresar el hecho de que una determinada propiedad (i.e., el solapamiento) es independiente de si se varía una o otra parte de la imagen de una misma forma (por ejemplo, el solapamiento es el mismo si se reduce en una proporción dada la intensidad del fondo y no la del objeto o si es la intensidad del objeto y no la del fondo la que se reduce en esa misma proporción).

Desde un punto de vista teórico, la migratividad fue introducida por Durante y Sarkoci ([5], y ver también [6]) para valores fijos del parámetro α , esto es, lo que en este trabajo vamos a denominar α -migratividad.

2 PRELIMINARES

En esta sección introducimos algunos conceptos preliminares sobre funciones migrativas. Nos vamos a reducir al caso de funciones de agregación ([1]), aunque la migratividad tiene sentido también para funciones más generales. Empezamos recordando las definiciones de α -migratividad y migratividad.

Definición 1. Sea $M : [0, 1] \times [0, 1] \rightarrow [0, 1]$ una función de agregación.

- (i) Sea $\alpha \in [0, 1]$. M es α -migrativa si para cualesquiera $x, y \in [0, 1]$ se tiene la identidad $M(\alpha x, y) = M(x, \alpha y)$.
- (ii) M es migrativa en sentido generalizado (o simplemente migrativa) si para cualesquiera $x, y, \alpha \in [0, 1]$ se tiene que $M(\alpha x, y) = M(x, \alpha y)$.

Las funciones migrativas han sido estudiadas con detalle en [3], donde en particular puede hallarse el siguiente resultado.

Teorema 1. Una función de agregación $M : [0, 1] \times [0, 1] \rightarrow [0, 1]$ es migrativa si y solamente si existe una función $g : [0, 1] \rightarrow [0, 1]$ no decreciente y con $g(0) = 0$ y $g(1) = 1$ de modo que

$$M(x, y) = g(xy) \text{ para todo } x, y \in [0, 1]. \quad (1)$$

Los conceptos de α -migratividad y migratividad pueden generalizarse como sigue.

Definición 2. Sean $\alpha \in [0, 1]$ y $B : [0, 1] \times [0, 1] \rightarrow [0, 1]$ una función de agregación. Una función de agregación $M : [0, 1] \times [0, 1] \rightarrow [0, 1]$ se dice que es α - B -migrativa si para cualesquiera $x, y \in [0, 1]$ se tiene que

$$M(B(x, \alpha), y) = M(x, B(\alpha, y)). \quad (2)$$

Ejemplo. Sea la función de agregación

$$B(x, y) = \frac{xy}{1 + 2xy - x - y}$$

para $(x, y) \in [0, 1]^2 - \{(0, 1), (1, 0)\}$. Definamos también $B(0, 1) = B(1, 0) = 0$. Entonces la función $M(x, y) = \min(x, y)$ es $1/2$ - B -migrativa.

Es importante hacer notar que, en la definición de α - B -migratividad no se impone ninguna clase de simetría sobre B . Por tanto, es importante mantener α como "elemento central" en la α - B -migratividad.

Definición 3. Sea $B : [0, 1] \times [0, 1] \rightarrow [0, 1]$ una función de agregación. Una función de agregación $M : [0, 1] \times [0, 1] \rightarrow [0, 1]$ se dice B -migrativa si es α - B -migrativa para todo $\alpha \in [0, 1]$; es decir, si la identidad

$$M(B(x, \alpha), y) = M(x, B(\alpha, y)) \quad (3)$$

se verifica para cualesquiera $x, y, \alpha \in [0, 1]$.

Ejemplo. Si definimos $M(x, y) = 0$ si $xy = 0$ y $M(x, y) = 1$ en cualquier otro caso, entonces M es una función de agregación B -migrativa tomando B tal

que $B(x, y) = 0$ si $xy = 0$ y $B(x, y) = x$ en cualquier otro caso. De hecho, M es una función de agregación B -migrativa para cualquier función de agregación B tal que $B(x, y) = 0$ si y solo si $xy = 0$.

Claramente, una función de agregación A es asociativa si y solamente si A es A -migrativa. En este sentido, también cabe entender la B -migratividad como una posible extensión del concepto de asociatividad.

Por otro lado, cabe destacar que la α -migratividad y la migratividad pueden recuperarse de la α - B -migratividad y la B -migratividad, respectivamente, sin más que tomar como función de agregación el producto $B(x, y) = xy$.

3 ALGUNOS RESULTADOS SOBRE FUNCIONES α - B -MIGRATIVAS

Comenzamos estudiando algunas propiedades de las funciones α - B -migrativas. A lo largo de toda esta sección consideramos un valor $\alpha \in [0, 1]$ fijado.

Proposición 1. Sean B una función de agregación y M una función de agregación α - B -migrativa. Entonces M es constante sobre $[0, B(0, \alpha)] \times [B(\alpha, 1), 1]$.

Demostración. Es suficiente observar que

$$M(B(0, \alpha), 1) = M(0, B(\alpha, 1))$$

y que, por ser una función de agregación, M es no decreciente.

Consideramos ahora la relación entre α - B -migratividad y la existencia de un elemento neutro. En este sentido, un primer resultado es el siguiente.

Proposición 2. Sea B una función de agregación con elemento neutro e_B . Entonces, cualquier función de agregación M es e_B - B -migrativa.

Demostración. Tenemos que, por ser e_B elemento neutro de B ,

$$M(x, y) = M(B(x, e_B), y) \text{ para todo } x, y \in [0, 1]; \quad (4)$$

y también

$$M(x, y) = M(x, B(e_B, y)) \text{ para todo } x, y \in [0, 1]. \quad (5)$$

El resultado se sigue de (4) y (5).

En cuanto a la existencia de elemento neutro para M también podemos presentar algunos resultados. En primer lugar, tenemos la siguiente proposición.

Proposición 3. Sea M una función de agregación α - B -migrativa. Supongamos que M tiene un elemento neutro e_M . Entonces

$$B(0, \alpha) \leq M(0, 1) \leq B(\alpha, 1) \quad (6)$$

Demostración. Por la Proposición 1 y la existencia de elemento neutro, tenemos que

$$B(\alpha, 1) = M(e_M, B(\alpha, 1)) \geq M(0, B(\alpha, 1)) = M(0, 1)$$

y

$$M(0, 1) = M(B(0, \alpha), 1) \geq B(0, \alpha) ,$$

lo que prueba el resultado.

Esta desigualdad puede precisarse algo más, como sigue.

Proposición 4. Sea M una función de agregación α - B -migrativa. Supongamos que M tiene un elemento neutro e_M . Entonces

- (i) Si $e_M \leq B(0, \alpha)$ entonces $B(\alpha, 1) = M(0, 1) = 1$;
- (ii) Si $e_M \geq B(\alpha, 1)$, entonces $B(0, \alpha) = M(0, 1) = 0$

Demostración. Supongamos que $e_M \leq B(0, \alpha)$. Por la Proposición 1, M es constante sobre $[0, B(0, \alpha)] \times [B(\alpha, 1), 1]$. En particular, es constante sobre $\{e_M\} \times [B(\alpha, 1), 1]$. Por ser e_M elemento neutro de M , esto solo es posible si $B(\alpha, 1) = 1$. Además, de nuevo por la Proposición 1, $M(0, 1) = M(e_M, 1) = 1$, con lo que queda demostrado (i).

(ii) se puede probar de forma análoga.

Terminamos esta sección con dos resultados importantes, especialmente de cara a futuros desarrollos. Antes de presentarlos recordamos la definición de función de agregación dual de una dada.

Definición 4. Sea $M : [0, 1] \times [0, 1] \rightarrow [0, 1]$ una función de agregación. El dual de M es la función de agregación $M^d : [0, 1] \times [0, 1] \rightarrow [0, 1]$ definida por:

$$M^d(x, y) = 1 - M(1 - x, 1 - y) \quad (7)$$

para todos los puntos $x, y \in [0, 1]$. En cierto sentido, la dualidad preserva la α - B -migratividad, como queda establecido en el siguiente teorema.

Teorema 2. Sea M una función de agregación α - B -migrativa. Entonces M^d es una función de agregación $(1 - \alpha)$ - B^d -migrativa.

Demostración. En primer lugar, observamos que

$$M^d(B^d(x, 1 - \alpha), y) = 1 - M(1 - B^d(x, 1 - \alpha), 1 - y) ,$$

lo que por definición es igual a

$$1 - M(B(1 - x, \alpha), 1 - y) . \quad (8)$$

Por otra parte, también se verifica la identidad

$$M^d(x, B^d(1 - \alpha, y)) = 1 - M(1 - x, 1 - B^d(1 - \alpha, y))$$

que una vez más por definición es igual a

$$1 - M(1 - x, B(\alpha, 1 - y)) . \quad (9)$$

Finalmente, de la propiedad α - B -migrativa se sigue que las expresiones (8) y (9) son iguales.

Corolario 1. Una función de agregación M es α - B -migrativa si y solamente si su función de agregación dual M^d es $(1 - \alpha)$ - B -migrativa.

Demostración. Es suficiente observar que, para cualquier función de agregación M , se verifica que

$$(M^d)^d(x, y) = M(x, y)$$

para cualesquiera $x, y \in [0, 1]$.

Consideramos ahora un automorfismo (una biyección monótona creciente o decreciente, ver [4]) $\varphi : [0, 1] \rightarrow [0, 1]$. Dada una función de agregación M , denotemos mediante M_φ la función de agregación definida por

$$M_\varphi(x, y) = \varphi^{-1}(M(\varphi(x), \varphi(y))) , \quad (10)$$

para cualesquiera $x, y \in [0, 1]$.

Esta construcción posibilita el siguiente resultado.

Teorema 3. Sea φ un automorfismo sobre el intervalo unidad $[0, 1]$. Entonces, para cualquier par de funciones de agregación M y B , M es α - B -migrativa si y solamente si M_φ es $\varphi^{-1}(\alpha)$ - B_φ -migrativa.

Demostración. Supongamos en primer lugar que M es α - B -migrativa. Entonces, tenemos que

$$M_\varphi(B_\varphi(x, \varphi^{-1}(\alpha)), y) = M_\varphi(\varphi^{-1}(B(\varphi(x), \alpha)), y)$$

que a su vez puede reescribirse en la forma

$$\varphi^{-1}(M(B(\varphi(x), \alpha), \varphi(y))) .$$

Esta última expresión, debido a la α - B -migratividad, es igual a

$$\varphi^{-1}(M(\varphi(x), B(\alpha, \varphi(y)))) = M_\varphi(x, \varphi^{-1}B(\alpha, \varphi(y)))$$

que a su vez puede reescribirse como

$$M_\varphi(x, B_\varphi(\varphi^{-1}(\alpha), y))$$

de forma que obtenemos la $\varphi^{-1}(\alpha)$ - B_φ -migratividad. Para obtener el recíproco, basta observar que para cualquier función de agregación M , $M = (M_\varphi)_{\varphi^{-1}}$. Si φ es una biyección monótona, también lo es, φ^{-1} , de modo que el resultado queda probado.

4 ALGUNOS RESULTADOS SOBRE FUNCIONES B -MIGRATIVAS

Comenzamos esta sección con un resultado análogo a la Proposición 1.

Proposición 5. Sea M una función de agregación B -migrativa. Entonces, M es constante sobre $[0, B(0, 1)] \times [B(0, 1), 1]$.

Demostración.

Tenemos que

$$M(0, 1) = M(B(0, 0), 1) = M(0, B(0, 1))$$

y también que

$$M(0, 1) = M(0, B(1, 1)) = M(B(0, 1), 1) .$$

Por tanto, M es constante sobre $\{0\} \times [B(0, 1), 1]$ y sobre $[0, B(0, 1)] \times \{1\}$, de donde se sigue el resultado.

Cabe destacar que de hecho es suficiente con requerir 0- B -migratividad y 1- B -migratividad a la función M para obtener el resultado.

La B -migratividad impone severas restricciones sobre las funciones de agregación consideradas. Por ejemplo

Proposición 6. Sea M una función de agregación B -migrativa. Supongamos que M posee un elemento neutro e_M . Entonces:

- (i) Si $e_M \leq B(0, 1)$, se tiene que $M(0, 1) = 1$;
- (ii) if $e_M \geq B(0, 1)$, se tiene que $M(0, 1) = 0$.

En particular, $M(0, 1)$ debe ser necesariamente igual a 0 o 1, y e_M debe ser diferente de $B(0, 1)$.

Demostración. Para demostrar (i) basta observar que

$$M(0, 1) = M(0, B(1, 1)) = M(B(0, 1), 1) \geq M(e_M, 1)$$

y este último término es igual a 1 por ser e_M el elemento neutro de M . El apartado (ii) se demuestra de una manera análoga. La observación final es obvia a partir del resultado.

Además, es posible obtener resultados análogos a los de la Sección 3. Por ejemplo.

Teorema 4. Un función de agregación M es B -migrativa si y solamente si su función de agregación dual M^d es B^d -migrativa.

Teorema 5. Sea φ un automorfismo sobre el intervalo unidad $[0, 1]$. Entonces, para cualquier par de funciones de agregación M y B , M es B -migrativa si y solamente si M_φ es B_φ migrativa.

Ambos resultados se siguen fácilmente de los Teoremas 2 y 3 y el Corolario 1 en la sección anterior.

5 ALGUNOS CASOS PARTICULARES

A continuación presentamos un breve estudio de la relación de los conceptos de α - B -migratividad y B -

migratividad con algunas de las funciones más utilizadas en el marco de la teoría difusa, como pueden ser las t-normas

Teorema 6. Sea T una t-norma B -migrativa. Entonces se tiene que

$$T(x, y) = B(x, y)$$

para cualesquiera $x, y \in [0, 1]$.

Demostración. En primer lugar, tenemos la cadena de identidades

$$x = T(x, 1) = T(1, x) = T(1, B(1, x)) = B(1, x) .$$

De una forma similar, llegamos a que

$$B(1, x) = B(x, 1) = x$$

Por tanto, para cualesquiera $x, y \in [0, 1]$, se tiene que

$$B(x, y) = T(B(x, y), 1) = T(x, B(y, 1)) = T(x, y)$$

que es lo que queríamos demostrar

Por otra parte, también tenemos el siguiente resultado.

Teorema 7. Sea M una función de agregación T -migrativa, donde T es una t-norma. Entonces, se verifica la identidad

$$M(x, y) = M(1, T(x, y))$$

para cualesquiera $x, y \in [0, 1]$.

Demostración. Tenemos la identidad

$$M(x, y) = M(T(1, x), y) = M(1, T(x, y))$$

tal como queríamos probar .

Cabe destacar que todos estos resultados pueden extenderse al caso de t-conormas por la dualidad.

Por último, presentamos una propiedad de las medias cuasiaritméticas que son α - B -migrativas.. Dichas medias son simplemente funciones de agregación M de la forma

$$M(x, y) = h^{-1}(ph(x) + qh(y))$$

donde $h : [0, 1] \rightarrow D \subset [-\infty, \infty]$ es una biyección estrictamente decreciente y $p, q \in [0, 1]$ son tales que se verifica $p + q = 1$.

Nuestro resultado es el siguiente.

Proposición 7. Sea M una media cuasiaritmética α - B -migrativa. Entonces se tiene que α es el elemento neutro de B si y solamente si existe $x_0 \in [0, 1]$ de modo que

$$B(x_0, \alpha) = x_0 .$$

Demostración. Si α es el elemento neutro de B , el resultado es obvio. Para ver el recíproco, sea $x_0 \in [0, 1]$ tal que

$$B(x_0, \alpha) = x_0 .$$

A partir de la α - B -migrativity, y para todo $y \in [0, 1]$, tenemos que

$$M(B(x_0, \alpha), y) = M(x_0, B(\alpha, y)) .$$

Si reescribimos esta expresión en términos de h , vemos que

$$0 = ph(B(x_0, \alpha)) - p(h(x_0)) = qh((B(\alpha, y)) - qh(y)$$

de modo que $y = B(\alpha, y)$ para todo $y \in [0, 1]$ y la demostración queda completa.

6 CONCLUSIONES

En este trabajo hemos presentado los conceptos de α - B -migratividad y B -migratividad, como generalización de las ideas de α -migratividad y migratividad, respectivamente. Hemos relacionado estos conceptos con algunas propiedades que suelen resultar de interés al considerar funciones de agregación, como por ejemplo la existencia de elemento neutro.

Los contenidos de este trabajo constituyen un primer paso en el análisis más exhaustivo de estas ideas. En particular, es importante desarrollar un marco teórico que permita comprender mejor el concepto de migratividad y su relación, por ejemplo, con la asociatividad. También sería interesante obtener una caracterización lo más completa posible de las funciones de agregación α - B -migrativas y B -migrativas. Por último, pretendemos estudiar la posible aplicación de nuestras construcciones en campos como la toma de decisiones o el tratamiento de imágenes.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por los proyectos: TIN2007-65981, TIN2009-07901 and APVV-0012-07.

Referencias

- [1] G. Beliakov, A. Pradera, T. Calvo, Aggregation Functions: A Guide for Practitioners, Springer, Berlin, 2007.
- [2] G. Beliakov, T. Calvo, On migrative means and copulas, *Proc. Fifth International School on Aggregation Operators, Palma de Mallorca, 2009*, Pág 107–110.
- [3] H. Bustince, J. Montero, R. Mesiar, Migrativity of aggregation functions. *Fuzzy Sets and Systems*, 160, Pág 766–777, 2009
- [4] H. Bustince, P. Burillo, F. Soria, Automorphisms, negations and implication operators. *Fuzzy Sets and Systems*, 134, Pág 209–229, 2003.
- [5] F. Durante, P. Sarkoci, A note on the convex combinations of triangular norms. *Fuzzy Sets and Systems*, 159, Pág 77–80, 2008.
- [6] J. Fodor, I. J. Rudas, On continuous triangular norms that are migrative. *Fuzzy Sets and Systems*, 158, Pág 1692–1697, 2007.
- [7] J. Montero, V. López, D. Gómez, The role of fuzziness in decision making. D.Ruan et al. *Fuzzy Logic: A Spectrum of Applied and Theoretical Issues*, Springer, Pág 337–349, 2007.
- [8] J. Montero, D. Gómez, S. Muñoz, Fuzzy information representation for decision aiding. *Proc. of the IPMU Conference, Málaga, Spain, June 22–27*, 2008.
- [9] B. Roy, Decision sciences or decision aid sciences. *European Journal of Operational Research*, 66, Pág 184–203, 1993

USING HEAVY AGGREGATIONS IN A UNIFIED MODEL BETWEEN THE WEIGHTED AVERAGE AND THE OWA OPERATOR

José M. Merigó

Montserrat Casanovas

Department of Business Administration, University of Barcelona,
Av. Diagonal 690, 08034 Barcelona, Spain, {jmerigo, mcasanovas}@ub.edu

Abstract

We introduce a new aggregation operator called the heavy ordered weighted averaging – weighted averaging (HOWAWA) operator. It is a new aggregation operator that uses the weighted average and the ordered weighted average in the same formulation and considering the degree of importance that each concept has in the analysis. Moreover, by using heavy aggregations, we are allowing the weighting vectors to range from the minimum to the total operator. Thus, we can consider a lot of different particular cases such as the heavy weighted average, the heavy OWA, the heavy arithmetic weighted average and the heavy arithmetic OWA operator.

Keywords: Aggregation operators, Weighted average, OWA operator, Heavy aggregations.

1 INTRODUCTION

The aggregation operators (also known as aggregation functions) are becoming more relevant in the fuzzy community. Every year we see a lot of new developments and new types of aggregation operators. Some of the most common ones are the weighted average (WA) and the ordered weighted averaging (OWA) operator [14]. For further reading on different types of aggregation operators, see for example [1-5,10,12,15,18].

Recently, some authors [11-13] have tried to unify both concepts in the same formulation. It is worth noting the work developed by Torra [11] with the introduction of the weighted OWA (WOWA) operator and the work of Xu and Da [13] about the hybrid averaging (HA) operator. Both approaches arrived to a unified model between the OWA and the WA because both concepts were included

in the formulation as particular cases. However, as it has been studied in [5-6], a better approach may be the use of a model that unifies both concepts and at the same time considers the degree of importance that each concept may have in the aggregation. This approach is known as the ordered weighted averaging – weighted averaging (OWAWA) operator [5-6]. Note that a similar approach has been suggested in [5,7] by using probabilities.

Another interesting aggregation approach introduced by Yager [16-17] is the heavy OWA (HOWA) operator. The main advantage of this model is that it allows the weighting vector to range between the usual OWA and the total operator. Thus, this approach is able to include a wide range of situations not considered by the usual OWA aggregation. For further information, see, for example [5,8-9,16-17].

The objective of this paper is to present the heavy OWAWA (HOWAWA) operator. It is a new aggregation operator that provides a unified model that uses OWAs, WAs, and heavy aggregations. Thus, we are able to provide a unified model between the OWA and the WA and at the same time we can allow their weighting vectors to move from the usual average to the total operator, that is, to the sum of all the arguments. One of the main advantages of this approach is that it includes a wide range of particular cases such as the heavy weighted average, the heavy OWA operator, different types of partial total operator, the arithmetic heavy weighted average, the arithmetic HOWA, the OWAWA operator, and a lot of other cases.

We also study the applicability of the new approach and we see that it is very broad because we can apply it in a lot of fields such as fuzzy set theory, statistics, engineering, decision theory, business and economics. We briefly develop a short illustrative example about the use of the HOWAWA operator.

The paper is organized as follows. In Section 2, we briefly review some basic concepts and the heavy weighted average. Section 3 presents the HOWAWA operator and

Section 4 studies some of its main families. In Section 5 we study the applicability of the new approach. Finally, in Section 6 we summarize the main conclusions of the paper.

2 PRELIMINARIES

2.1. THE OWA OPERATOR

The OWA operator [14] provides a parameterized family of aggregation operators between the maximum and the minimum. It can be defined as follows.

Definition 1. An OWA operator of dimension n is a mapping OWA: $R^n \rightarrow R$ that has an associated weighting vector W of dimension n with $\sum_{j=1}^n w_j = 1$ and $w_j \in [0, 1]$, such that:

$$\text{OWA}(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j b_j \quad (1)$$

where b_j is the j th largest of the a_i .

2.2. THE HEAVY OWA OPERATOR

The HOWA operator [16] is an extension of the OWA operator that allows the weighting vector to sum up to n . Thus, we are able to include the total operator in the aggregation. It can be defined as follows:

Definition 4. A HOWA operator is a mapping HOWA: $R^n \rightarrow R$ that has an associated weighting vector W with $w_j \in [0, 1]$ and $1 \leq \sum_{j=1}^n w_j \leq n$, such that:

$$\text{HOWA}(a_1, \dots, a_n) = \sum_{j=1}^n w_j b_j \quad (2)$$

where b_j is the j th largest of the a_i and a_i is the argument variable.

As we can see, if $\sum_{i=1}^n w_i = 1$, then, we get the usual weighted average and if $\sum_{i=1}^n w_i = n$, the total operator.

2.3. THE HEAVY WEIGHTED AVERAGE

The heavy weighted average (HWA) is an extension of the weighted average for situations where we allow the weighting vector to move from the usual weighted average to the total operator. Thus, we get a more complete representation of the aggregation process. Note that the heavy aggregations are very useful for comparing

the results obtained by using some kind of average and the sum of all the available results (or arguments), that is, the total results. For example, in distance measures [5,9] this is very useful because we can unify in the same formulation the normalized (or relative) distance with the absolute (or total) distance. The HWA can be defined as follows:

Definition 3. A HWA operator of dimension n is a mapping HWA: $R^n \rightarrow R$ that has an associated weighting vector W , with $w_i \in [0, 1]$ and $1 \leq \sum_{i=1}^n w_i \leq n$, such that:

$$\text{HWA}(a_1, \dots, a_n) = \sum_{i=1}^n w_i a_i \quad (3)$$

where a_i represents the i th argument variable.

Note that if $\sum_{i=1}^n w_i = 1$, then, we get the usual weighted average and if $\sum_{i=1}^n w_i = n$, then, the total operator.

2.4. THE OWAWA OPERATOR

The ordered weighted averaging – weighted averaging (OWAWA) operator is an aggregation operator that unifies the WA and the OWA operator in the same formulation considering the degree that each concept has in the analysis [5-6]. It can be defined as follows.

Definition 4. An OWAWA operator of dimension n is a mapping OWAWA: $R^n \rightarrow R$ that has an associated weighting vector W of dimension n such that $w_j \in [0, 1]$ and $\sum_{j=1}^n w_j = 1$, according to the following formula:

$$\text{OWAWA}(a_1, \dots, a_n) = \sum_{j=1}^n \hat{v}_j b_j \quad (4)$$

where b_j is the j th largest of the a_i , each argument a_i has an associated weight (WA) v_i with $\sum_{i=1}^n v_i = 1$ and $v_i \in [0, 1]$, $\hat{v}_j = \beta w_j + (1 - \beta)v_j$ with $\beta \in [0, 1]$ and v_j is the weight (WA) v_i ordered according to b_j , that is, according to the j th largest of the a_i .

By choosing a different manifestation in the weighting vector, we are able to obtain a wide range of particular types of OWAWA operators [5-6]. Especially, when $\beta = 0$, we get the WA, and if $\beta = 1$, we get the OWA operator. Other interesting cases are found when $w_j = 1/n$, for all a_i , because then, we get the arithmetic probability (AP). And if $v_i = 1/n$, for all a_i , we get the arithmetic OWA operator. Note that inside the arithmetic OWA we find the arithmetic maximum and minimum, and so on.

3 THE HEAVY OWAWA OPERATOR

The heavy ordered weighted averaging – weighted averaging (HOWAWA) operator is a new model that unifies the OWA operator and the weighted average in the same formulation. Therefore, both concepts can be seen as a particular case of a more general one. Moreover, we allow the aggregation to move between the minimum and the total operator. Thus, we can represent a lot of situations not included in the usual OWA and WA aggregations.

Note that some previous models already considered the possibility of using OWA operators and WAs in the same formulation. The main models are the weighted OWA (WOWA) operator [11-12] and the hybrid averaging (HA) operator [13]. In this case, we will get the heavy WOWA operator and the heavy HA operator. However, if we analyze in detail these models, we will see that these formulations cannot deal with heavy aggregations because then, the results and the process become inconsistent.

In the following, we are going to analyze the HOWAWA operator. It can be defined as follows.

Definition 5. A HOWAWA operator of dimension n is a mapping HOWAWA: $R^n \rightarrow R$ that has an associated weighting vector W of dimension n such that $w_j \in [0, 1]$ and $1 \leq \sum_{j=1}^n w_j \leq n$, according to the following formula:

$$\text{HOWAWA}(a_1, \dots, a_n) = \sum_{j=1}^n \hat{v}_j b_j \quad (5)$$

where b_j is the j th largest of the a_i , each argument a_i has an associated weight (WA) v_i with $1 \leq \sum_{i=1}^n v_i \leq n$ and $v_i \in [0, 1]$, $\hat{v}_j = \beta w_j + (1 - \beta)v_j$ with $\beta \in [0, 1]$ and v_j is the weight (WA) v_i ordered according to b_j , that is, according to the j th largest of the a_i .

Note that it is also possible to formulate the HOWAWA operator separating the part that strictly affects the HOWA operator and the part that affects the HWA. This representation is useful to see both models in the same formulation but it does not seem to be as a unique equation that unifies both models.

Definition 6. An HOWAWA operator is a mapping HOWAWA: $R^n \rightarrow R$ of dimension n , if it has an associated weighting vector W , with $1 \leq \sum_{j=1}^n w_j \leq n$ and $w_j \in [0, 1]$ and a weighting vector V that affects the WA, with $1 \leq \sum_{i=1}^n v_i \leq n$ and $v_i \in [0, 1]$, such that:

$$\text{HOWAWA}(a_1, \dots, a_n) = \beta \sum_{j=1}^n w_j b_j + (1 - \beta) \sum_{i=1}^n v_i a_i \quad (6)$$

where b_j is the j th largest of the a_i and $\beta \in [0, 1]$.

In the following, we are going to give a simple example of how to aggregate with the HOWAWA operator. We consider the aggregation with both definitions.

Example 1. Assume the following arguments in an aggregation process: (60, 80, 30, 20, 40). Assume the following weighting vector $W = (0.6, 0.6, 0.4, 0.2, 0.2)$ and the following probabilistic weighting vector $V = (0.3, 0.3, 0.4, 0.5, 0.5)$. Note that the WA has a degree of importance of 50% while the weighting vector W of the OWA a degree of 50%. If we want to aggregate this information by using the HOWAWA operator, we will get the following. The aggregation can be solved either with (5) or (6). With (5) we calculate the new weighting vector as:

$$\hat{v}_1 = 0.5 \times 0.6 + 0.5 \times 0.3 = 0.45$$

$$\hat{v}_2 = 0.5 \times 0.6 + 0.5 \times 0.3 = 0.45$$

$$\hat{v}_3 = 0.5 \times 0.4 + 0.5 \times 0.5 = 0.45$$

$$\hat{v}_4 = 0.5 \times 0.2 + 0.5 \times 0.4 = 0.3$$

$$\hat{v}_5 = 0.5 \times 0.2 + 0.5 \times 0.5 = 0.35$$

Then, we calculate the aggregation process as follows:

$$\text{HOWAWA} = 0.45 \times 80 + 0.45 \times 60 + 0.45 \times 40 + 0.3 \times 30 + 0.35 \times 20 = 97.$$

With (4), we aggregate as follows:

$$\text{HOWAWA} = 0.5 \times (0.6 \times 80 + 0.6 \times 60 + 0.4 \times 40 + 0.2 \times 30 + 0.2 \times 20) + 0.5 \times (0.3 \times 60 + 0.3 \times 80 + 0.4 \times 30 + 0.5 \times 20 + 0.5 \times 40) = 97.$$

Obviously, we get the same results with both methods.

Note that it is possible to distinguish between descending (DHOWAWA) and ascending (AHOWAWA) orders by using $w_j = w_{n-j+1}^*$, where w_j is the j th weight of the DHOWAWA and w_{n-j+1}^* the j th weight of the AHOWAWA operator.

If B is a vector corresponding to the ordered arguments b_j , we shall call this the ordered argument vector and W^T is the transpose of the weighting vector, then, the HOWAWA operator can be expressed as:

$$\text{HOWAWA}(a_1, \dots, a_n) = W^T B \quad (7)$$

The HOWAWA is monotonic, bounded and idempotent. It is monotonic because if $a_i \geq u_i$, for all a_i , then, $\text{HOWAWA}(a_1, a_2, \dots, a_n) \geq \text{HOWAWA}(u_1, u_2, \dots, u_n)$. It is bounded because the HOWAWA aggregation is delimited by the minimum and the total operator. That is, $\text{Min}\{a_i\} \leq \text{HOWAWA}(a_1, a_2, \dots, a_n) \leq \text{Total operator}$. It is idempotent because if $a_i = a$, for all a_i , then, $\text{HOWAWA}(a_1, a_2, \dots, a_n) = a$. Note that this operator is not commutative because the weighted average is not commutative.

Another interesting issue to analyze are the measures for characterizing the weighting vector W . Following a similar methodology as it has been developed for the OWA operator [5,14,16] we can formulate the attitudinal character, the entropy of dispersion, the divergence of W and the balance operator. Note that these measures affect the weighting vector W but not the WAs because they are given as some kind of objective information that cannot be manipulated according to the decision makers interests. The first measure, the attitudinal character, can be defined as follows:

$$\alpha(W) = \frac{1}{|W|} \sum_{j=1}^n \left(\frac{n-j}{n-1} \right) w_j \quad (8)$$

As it can be seen, $\alpha(W) \in [0, 1]$. Note that the total operator has $\alpha(W) = 0.5$. The second measure, the entropy of dispersion, can be defined as:

$$H(W) = -\frac{1}{|W|} \sum_{j=1}^n w_j \ln \left(\frac{w_j}{|W|} \right) \quad (9)$$

Note that for the total operator, $H(W) = -\ln n$. A third measure that can be used based on [14], is the divergence of W , we will use:

$$\text{Div}(W) = \frac{1}{|W|} \sum_{j=1}^n w_j \left(\frac{n-j}{n-1} - \alpha(W) \right)^2 \quad (10)$$

If $|W| = n$, we get the divergence for the total operator and it is the same divergence than the average. That is, $\text{Div}(W) = (1/12)[(n+1)/(n-1)]$.

For the balance operator, we get:

$$\text{BAL}(W) = \frac{1}{|W|} \sum_{j=1}^n \left(\frac{n+1-2j}{n-1} \right) w_j \quad (11)$$

Note also that these four measures are reduced to the usual definitions [5,8-9,16-17] when $|W| = 1$.

4 FAMILIES OF HOWAWA OPERATORS

Different types of HOWAWA operators are found by using a different manifestation in the weighting vector or in the coefficient β . For example, we can obtain the following cases.

- OWAWA operator: When $\sum_{j=1}^n w_j = 1$ and $\sum_{i=1}^n v_i = 1$.
 - The particular cases of the OWAWA operator are explained in [10].
- Total operator: When $\sum_{j=1}^n w_j = n$ and $\sum_{i=1}^n v_i = n$.
- The total weighted average operator: When $\sum_{j=1}^n w_j = n$ and $\sum_{i=1}^n v_i = 1$.
- The total heavy weighted average operator: When $\sum_{j=1}^n w_j = n$.
- The total OWA operator: When $\sum_{i=1}^n v_i = n$ and $\sum_{j=1}^n w_j = 1$.
- The total heavy OWA operator: When $\sum_{i=1}^n v_i = n$.
- The weighted heavy OWA operator: When $\sum_{i=1}^n v_i = 1$.
- The OWA heavy weighted average: When $\sum_{j=1}^n w_j = 1$.
- The HOWA operator: If $\beta = 1$.
- The heavy weighted average: If $\beta = 0$.
- The heavy average: If $w_j = |W|/n$ and $v_i = |W|/n$, or $w_j = |W|/n$ and $\beta = 1$, or $v_i = |W|/n$ and $\beta = 0$.
- The heavy arithmetic weighted average: If $w_j = |W|/n$.
- The heavy arithmetic OWA operator: If $v_i = |W|/n$.
- The arithmetic heavy weighted average: If $w_j = 1/n$.
- The arithmetic heavy OWA operator: If $v_i = 1/n$.
- The arithmetic mean: If $w_j = 1/n$ and $v_i = 1/n$, for all i .
- The minimum: When $w_n = 1$, $w_j = 0$, for all $j \neq n$ and $\beta = 1$.
- The heavy weighted minimum: When $w_n = 1$, $w_j = 0$, for all $j \neq n$.
- Push up allocation: We use $w_j = (1 \wedge (|W| - (j - 1))) \vee 0$ and $\beta = 1$.
- Heavy weighted push up allocation: We use $w_j = (1 \wedge (|W| - (j - 1))) \vee 0$.
- Weighted push up allocation: In this case, $w_j = (1 \wedge (|W| - (j - 1))) \vee 0$ and $\sum_{i=1}^n v_i = 1$.
- The maximum: When $w_1 = 1$, $w_j = 0$, for all $j \neq 1$ and $\beta = 1$.
- Push down allocation: We use $w_{n-j+1} = (1 \wedge (|W| - (j - 1))) \vee 0$ and $\beta = 1$.
- Heavy weighted push down allocation: In this case, $w_{n-j+1} = (1 \wedge (|W| - (j - 1))) \vee 0$.
- Weighted push down allocation: We use $w_{n-j+1} = (1 \wedge (|W| - (j - 1))) \vee 0$ and $\sum_{i=1}^n v_i = 1$.

- Median type allocation: If n is even, we allocate the weights for $j = 1$ to a as $w_{a+j} = w_{a+1-j} = [1 \wedge ((|W| - 2(j - 1))/2)] \vee 0$. If n is odd, we allocate the weights for $j = 1$ to a as $w_{a+1} = 1$ and $w_{a+1-j} = w_{a+1+j} = [1 \wedge ((|W| - 1) - 2(j - 1))/2)] \vee 0$.
- Heavy weighted median allocation: The same than the median type allocation but $\beta \neq 1$.
- Weighted median allocation: The same than the heavy weighted median allocation and $\sum_{i=1}^n v_i = 1$.
- Step-HOWAWA allocation: Assuming $\beta = 1$ and $b = \min[(K - 1), (n - K)]$, we allocate the weights for j to b as $w_K = 1$ and $w_{K+j} = w_{K-j} = [1 \wedge ((|W| - 1) - 2(j - 1))/2)] \vee 0$.
- Heavy weighted step-HOWAWA allocation: We assume $b = \min[(K - 1), (n - K)]$, and we allocate the weights for j to b as $w_K = 1$ and $w_{K+j} = w_{K-j} = [1 \wedge ((|W| - 1) - 2(j - 1))/2)] \vee 0$.
- Weighted step-HOWAWA allocation: The same than the heavy weighted step-HOWAWA allocation and $\sum_{i=1}^n v_i = 1$.
- Olympic-average allocation: We have to distinguish between two cases. Note that in both cases we have $\beta = 1$.
 - In the first case, where $|W| < n - 2m$, we allocate the weight as $w_j = |W|/(n - 2m)$ for $j = m + 1$ to $n - m$, and $w_j = 0$ for $j = 1$ to m and for $j = n - m + 1$ to n .
 - In the second case, where $|W| \geq n - 2m$, we allocate the weights as $w_j = 1$ for $j = m + 1$ to $n - m$ and $w_{m+1-j} = w_{n-m+j} = [1 \wedge ((|W| - (n - 2m)) - 2(j - 1))/2)] \vee 0$ for $j = 1$ to m .
- Heavy weighted olympic allocation: The same than the olympic-average allocation but $\beta \neq 1$.
- Weighted olympic allocation: The same than the heavy weighted olympic allocation and $\sum_{i=1}^n v_i = 1$.
- Arrow-Hurwicz allocation: Assuming that $\beta = 1$, $|W| = q$ and dimension n , we define the weights in two directions, push up and push down. First, we calculate $\omega_j = (1 \wedge (\lambda q - (j - 1))) \vee 0$ for $j = 1$ to n and $\hat{w}_{n-j+1} = (1 \wedge ((1 - \lambda)q - (j - 1))) \vee 0$ for $j = 1$ to n . Then, we define the weights as $w_i = \omega_i + \hat{w}_i$. Note that $\omega_j = 0$ for all $j \geq \lambda q + 1 \geq \lambda|W| + 1$ and $\hat{w}_j = 0$ for $j \leq n - (1 - \lambda)q \leq n - |W| + \lambda|W|$.
- The heavy weighted Arrow-Hurwicz allocation: The same than the Arrow-Hurwicz allocation but $\beta \neq 1$.
- The weighted Arrow-Hurwicz allocation: The same than the heavy weighted olympic allocation and $\sum_{i=1}^n v_i = 1$.

Note that it is possible to develop other types of HOWAWA operators by using other manifestations of the weighting vector of the WA and the OWA.

5 APPLICABILITY OF THE HOWAWA

The HOWAWA operator can be used in a lot of applications. In general, we can conclude that the HOWAWA operator can be implemented in all the studies that use the weighted average or the OWA operator because both of them are particular cases of this new approach. Therefore, all the studies that use the OWA or the WA can be revised and extended with this new approach. For example, we can apply it in fuzzy set theory, soft computing, statistics, decision theory, operational research, business and economics, politics, engineering and a lot of other sciences such as mathematics, physics, chemistry and biology.

In the following we are going to develop a simple numerical example. Assume we have three experts that are analyzing the benefits of a company for the next year. Expert 1 knows very well the European market of the company, Expert 2 the American market and Expert 3 the Asian market. Each of them has confidential information about the market they are working on. Therefore, when predicting the results, each of them gives a general result that takes into account this confidential information. In this example we assume that the three experts have a degree of 50% of independence in their information. Therefore, if we calculate the average result in order to see the predicted benefits by the experts, we have to allow the weighting vector to sum up to 1.5 so we take into account all the information. In this example we assume that all the experts provide information that it is equally important. Therefore, we use the following weighting vector $V = (0.5, 0.5, 0.5)$. The company is very pessimistic so they want to use an OWA aggregation that reflects this aspect in the following way: $W = (0.2, 0.2, 0.6)$. Note that this weighting vector has to be increased 50% obtaining $W^* = (0.3, 0.3, 0.9)$. Assume that the experts predict the following results:

- Expert 1: 300 millions €.
- Expert 2: 250 millions €.
- Expert 3: 350 millions €.

If we aggregate this information using a heavy aggregation we get the following. First, we have to mix the weights of the OWA and the WA in order to provide a unified aggregation. In this example, we assume that both concepts have a degree of importance of 50%.

$$\hat{v}_1 = 0.5 \times 0.3 + 0.5 \times 0.5 = 0.4$$

$$\hat{v}_2 = 0.5 \times 0.3 + 0.5 \times 0.5 = 0.4$$

$$\hat{v}_3 = 0.5 \times 0.9 + 0.5 \times 0.5 = 0.7$$

$$\text{HOWAWA} = 0.4 \times 350 + 0.4 \times 300 + 0.7 \times 250 = 435.$$

As we can see, assuming the previous conditions, the company would assume that the predicted benefit for the next period is 435 millions €.

5 CONCLUSIONS

We have presented a new aggregation operator that uses the weighted average and the OWA operator in the same formulation and considering the degree of importance that each concept has in the aggregation. Moreover, this new approach allows the weighting vector of the OWA and the WA to range from the usual average to the total operator. Therefore, we have been able to consider a lot of new aggregation operators. We have called this new approach the HOWAWA operator. We have studied some of its main particular cases such as the OWAWA operator, the total operator, the total weighted average operator, the total heavy weighted average operator, the arithmetic heavy weighted average, the arithmetic heavy OWA operator, and a lot of other cases.

We have also studied the applicability of the new approach and we have seen that it is very broad because all the studies that use the OWA or the WA can be revised and extended with this new approach.

In future research, we will develop further extensions of this approach by using other sources of information such as the use of interval numbers, fuzzy numbers and expertons. We will also consider the use of order-inducing variables and distance measures and we will develop some applications in decision making problems.

Acknowledgements

We would like to thank the anonymous referees for valuable comments that have improved the quality of the paper.

References

- [1] G. Beliakov, A. Pradera and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Berlin: Springer-Verlag, 2007.
- [2] V. Cutello and J. Montero, Hierarchical aggregation of OWA operators: basic measures and related computational problems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 3:17-26, 1995.
- [3] V. Cutello and J. Montero, Recursive connective rules. *International Journal of Intelligent Systems*, 14:3-20, 1999.
- [4] D. Gómez and J. Montero, A discussion on aggregation operators. *Kybernetika*, 40:107-120, 2004.
- [5] J.M. Merigó, *New extensions to the OWA operators and its application in decision making* (In Spanish). PhD Thesis, Department of Business Administration, University of Barcelona, 2008.
- [6] J.M. Merigó, On the use of the OWA operator in the weighted average and its application in decision making. In: *Proceedings of the WCE 2009 Conference*, London, United Kingdom, pp. 82-87, 2009.
- [7] J.M. Merigó, Probabilistic decision making with the OWA operator and its application in investment management. In: *Proceedings of the IFSA-EUSFLAT Conference*, Lisbon, Portugal, pp. 1364-1369, 2009.
- [8] J.M. Merigó and M. Casanovas, Using fuzzy numbers in heavy aggregation operators. *International Journal of Information Technology*, 4, 177-182, 2008.
- [9] J.M. Merigó and M. Casanovas, Distance measures in induced and heavy aggregation operators. In: *Proceedings of the EUROFUSE Conference*, Pamplona, Spain, pp. 219-226, 2009.
- [10] J.M. Merigó and A.M. Gil-Lafuente, The induced generalized OWA operator. *Information Sciences*, 179:729-741, 2009.
- [11] V. Torra, The weighted OWA operator. *International Journal of Intelligent Systems*, 12:153-166, 1997.
- [12] V. Torra and Y. Narukawa, *Modeling Decisions: Information Fusion and Aggregation Operators*. Berlin: Springer-Verlag, 2007.
- [13] Z.S. Xu and Q.L. Da, An overview of operators for aggregating information. *International Journal of Intelligent Systems*, 18:953-968, 2003.
- [14] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics B*, 18:183-190, 1988.
- [15] R.R. Yager, Families of OWA operators. *Fuzzy Sets and Systems*, 59:125-148, 1993.
- [16] R.R. Yager, Heavy OWA operators. *Fuzzy Optimization and Decision Making*, 1:379-397, 2002.
- [17] R.R. Yager, Monitored heavy fuzzy measures and their role in decision making under uncertainty. *Fuzzy Sets and Systems*, 139:491-513, 2003.
- [18] R.R. Yager and J. Kacprzyk, *The Ordered Weighted Averaging Operators: Theory and Applications*. Norwell: Kluwer Academic Publishers, 1997.

DECISION MAKING WITH DISTANCE MEASURES, WEIGHTED AVERAGES AND INDUCED OWA OPERATORS

José M. Merigó

Anna M. Gil-Lafuente

Department of Business Administration, University of Barcelona,
Av. Diagonal 690, 08034 Barcelona, Spain, {jmerigo, amgil}@ub.edu

Abstract

We develop a new decision making model by using distance measures, weighted averages and OWA operators. We introduce the induced ordered weighted averaging – weighted averaging distance (IOWAWAD) operator. We study some of its main properties and particular cases such as the weighted Hamming distance, the induced OWA distance (IOWAD), the arithmetic weighted distance and the arithmetic IOWAD operator. We apply the new approach in a decision making problem about product management.

Keywords: Decision making, Distance measures, Induced OWA operator, Weighted average.

1 INTRODUCTION

In the literature, we find a wide range of aggregation operators (or aggregation techniques) that have been useful in the field of decision making [2,5-11,15,17-20]. A very useful technique for doing this is the Hamming distance [3] and more generally all the distance measures [2-5,11]. The main advantage of using distance measures in decision making is that we can compare the alternatives of the problem with some ideal result [2-5,11].

In order to develop the decision process we may use different types of aggregation operators (also known as aggregation functions) for the normalization process of the distance measures. The weighted average (WA) is one of the most common aggregation operators found in the literature. It can be used in a wide range of problems including statistics, economics and engineering. Another interesting aggregation operator is the ordered weighted averaging (OWA) operator [15]. The OWA operator

provides a parameterized family of aggregation operators that range from the maximum to the minimum, see, for example [1,5-20].

Recently, some authors [12-13,14] have tried to unify both concepts in the same formulation such as the work developed by Torra [12-13] with the introduction of the weighted OWA (WOWA) operator and the work of Xu and Da [14] about the hybrid averaging (HA) operator. Both models unified the OWA and the WA because both concepts were included in the formulation as particular cases. However, as it has been studied in [5-7], these models seem to be a partial unification but not a total one because they can unify them but they cannot consider how relevant these concepts are in the specific problem considered. For example, in some problems we may prefer to give more importance to the OWA operator because we believe that it is more relevant and vice versa. Recently, Merigó [5-6] has developed a new approach that unifies the OWA and the WA in the same formulation considering the degree of importance that each concept may have in the problem. He called it the ordered weighted averaging – weighted averaging (OWAWA) operator. Moreover, he has further extended this approach by using order-inducing variables and uncertain information [5,7].

The aim of this paper is to develop a new distance measure that we call the induced OWAWA distance operator and apply it in a decision making problem about product management. The IOWAWAD operator is a distance aggregation operator that provides a parameterized family of aggregation operators between the maximum and the minimum distances. It normalizes the Hamming distance (or other distances) with the IOWAWA operator. Therefore, we are able to include the WA and the IOWA operator at the same time in the Hamming distance. Thus, it includes a wide range of distances such as the arithmetic weighted Hamming distance, the arithmetic IOWAD operator, the OWAWA distance operator, the IOWAD operator, the weighted Hamming distance, the maximum weighted Hamming distance, and a lot of other particular cases.

This paper is structured as follows. In Section 2, we review some basic concepts. Section 3 presents the IOWAWAD operator and Section 4 gives a numerical example. In Section 5 we summarize the conclusions.

2 PRELIMINARIES

2.1. THE HAMMING DISTANCE

The Hamming distance [3] is a very useful technique for calculating the differences between two elements, two sets, etc. In fuzzy set theory, it can be useful, for example, for the calculation of distances between fuzzy sets, interval-valued fuzzy sets and intuitionistic fuzzy sets. For two sets A and B , it can be defined as follows.

Definition 1. A weighted Hamming distance of dimension n is a mapping $d_{WH}: [0, 1]^n \times [0, 1]^n \rightarrow [0, 1]$ that has an associated weighting vector W of dimension n such that the sum of the weights is 1 and $w_j \in [0, 1]$. Then:

$$d_{WH}(A, B) = \left(\sum_{i=1}^n w_i |a_i - b_i| \right) \quad (1)$$

where a_i and b_i are the i th arguments of the sets A and B respectively.

Note that it is possible to generalize this definition to all the real numbers by using $R^n \times R^n \rightarrow R$. Note also that if $w_i = 1/n$, for all i , then, we get the normalized Hamming distance. For the formulation used in fuzzy set theory, see for example [2-5,11].

2.2. THE OWA OPERATOR

The OWA operator [15] provides with a parameterized family of aggregation operators that include the arithmetic mean, the maximum and the minimum. It can be defined as follows.

Definition 2. An OWA operator of dimension n is a mapping $OWA: R^n \rightarrow R$ that has an associated weighting vector W of dimension n such that the sum of the weights is 1 and $w_j \in [0, 1]$, then:

$$OWA(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j b_j \quad (2)$$

where b_j is the j th largest of the a_i .

2.3. THE INDUCED OWA OPERATOR

The IOWA operator [17,19] is an extension of the OWA operator. Its main difference is that the reordering step is not carried out with the values of the arguments a_i . In this case, the reordering step is developed with order inducing variables that reflects a more complex reordering process. It can be defined as follows.

Definition 3. An IOWA operator of dimension n is a mapping $IOWA: R^n \times R^n \rightarrow R$ that has an associated weighting vector W of dimension n such that $\sum_{j=1}^n w_j = 1$ and $w_j \in [0, 1]$, then:

$$IOWA(\langle u_1, a_1 \rangle, \langle u_2, a_2 \rangle, \dots, \langle u_n, a_n \rangle) = \sum_{j=1}^n w_j b_j \quad (3)$$

where b_j is the a_i value of the IOWA pair $\langle u_i, a_i \rangle$ having the j th largest u_i , u_i is the order inducing variable and a_i is the argument variable.

2.4. THE INDUCED OWA DISTANCE OPERATOR

The induced OWAD (IOWAD) operator [5] represents an extension of the traditional normalized distance by using IOWA operators. The difference is that we reorder the arguments of the individual distances with order inducing variables that represents a complex reordering process. It can be defined as follows.

Definition 4. An induced OWAD operator of dimension n is a mapping $IOWAD: R^n \times R^n \times R^n \rightarrow R$ that has an associated weighting vector W of dimension n such that $\sum_{j=1}^n w_j = 1$ and $w_j \in [0, 1]$. Then, the distance between two sets is:

$$IOWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = \sum_{j=1}^n w_j D_j \quad (4)$$

where D_j is the $|x_i - y_i|$ value of the IOWAD triplet $\langle u_i, x_i, y_i \rangle$ having the j th largest u_i , u_i is the order inducing variable, x_i is the i th characteristic of the ideal set, y_i is the i th characteristic of the k th alternative and $k = 1, 2, \dots, m$.

2.5. THE OWAWA OPERATOR

The ordered weighted averaging – weighted averaging (OWAWA) operator is an aggregation operator that unifies the WA and the OWA operator in the same formulation considering the degree of importance that each concept has in the analysis [5]. It is defined as follows.

Definition 5. An OWAWA operator of dimension n is a mapping OWAWA: $R^n \rightarrow R$ that has an associated weighting vector W of dimension n such that $w_j \in [0, 1]$ and $\sum_{j=1}^n w_j = 1$, according to the following formula:

$$\text{OWAWA}(a_1, \dots, a_n) = \sum_{j=1}^n \hat{v}_j b_j \quad (5)$$

where b_j is the j th largest of the a_i , each argument a_i has an associated weight (WA) v_i with $\sum_{i=1}^n v_i = 1$ and $v_i \in [0, 1]$, $\hat{v}_j = \beta w_j + (1 - \beta)v_j$ with $\beta \in [0, 1]$ and v_j is the weight (WA) v_i ordered according to b_j , that is, according to the j th largest of the a_i .

By choosing a different manifestation in the weighting vector, we are able to obtain a wide range of particular types of OWAWA operators [5]. Especially, when $\beta = 0$, we get the WA, and if $\beta = 1$, we get the OWA operator.

2.6. THE INDUCED OWAWA OPERATOR

The induced ordered weighted averaging – weighted averaging (OWAWA) operator is a new model that unifies the OWA operator and the weighted average in the same formulation and considering a complex reordering process based on order inducing variables. It can be defined as follows.

Definition 6. An IOWAWA operator of dimension n is a mapping IOWAWA: $R^n \times R^n \rightarrow R$ that has an associated weighting vector W of dimension n such that $w_j \in [0, 1]$ and $\sum_{j=1}^n w_j = 1$, according to the following formula:

$$\text{IOWAWA}(\langle u_1, a_1 \rangle, \langle u_2, a_2 \rangle, \dots, \langle u_n, a_n \rangle) = \sum_{j=1}^n \hat{v}_j b_j \quad (6)$$

where b_j is the a_i value of the IOWAWA pair $\langle u_i, a_i \rangle$ having the j th largest u_i , u_i is the order inducing variable and a_i is the argument variable, each argument a_i has an associated weight (WA) v_i with $\sum_{i=1}^n v_i = 1$ and $v_i \in [0, 1]$, $\hat{v}_j = \beta w_j + (1 - \beta)v_j$ with $\beta \in [0, 1]$ and v_j is the weight (WA) v_i ordered according to b_j , that is, according to the j th largest of the a_i .

3 THE INDUCED OWAWA DISTANCE

The induced ordered weighted averaging weighted averaging distance (IOWAWAD) operator is a distance measure that uses the WA and the OWA operator in the normalization process of the Hamming distance by using

the IOWAWA operator. Then, the reordering of the individual distances is developed according to order-inducing variables that represent a complex reordering process of the individual distances formed by comparing two sets. The main advantage of this new approach is that it is able to deal with situations where we have some subjective information about the possibility of occurrence of the different results and the attitudinal character of the decision maker that it is assessed with order inducing variables in order to represent his attitude in a complete way considering a wide range of aspects such as the degree of optimism, psychological aspects and time pressure. It can be defined as follows for two sets $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$.

Definition 7. An IOWAWAD operator of dimension n is a mapping IOWAWAD: $R^n \times R^n \times R^n \rightarrow R$ that has an associated weighting vector W such that $w_j \in [0, 1]$ and $\sum_{j=1}^n w_j = 1$, according to the following formula:

$$\text{IOWAWAD}(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = \sum_{j=1}^n \hat{v}_j b_j \quad (7)$$

where b_j is the $|x_i - y_i|$ value of the IOWAWAD triplet $\langle u_i, x_i, y_i \rangle$ having the j th largest u_i , u_i is the order inducing variable, x_i is the i th argument of the set $X = \{x_1, \dots, x_n\}$, y_i is the i th argument of the set $Y = \{y_1, \dots, y_n\}$, each argument or individual distance $|x_i - y_i|$ has an associated weight (WA) v_i with $\sum_{i=1}^n v_i = 1$ and $v_i \in [0, 1]$, $\hat{v}_j = \beta w_j + (1 - \beta)v_j$ with $\beta \in [0, 1]$ and v_j is the weight (WA) v_i ordered according to the j th largest u_i .

Note that it is also possible to formulate the IOWAWAD operator separating the part that strictly affects the IOWAD operator and the part that affects the WAD.

Definition 8. An IOWAWAD operator is a mapping IOWAWAD: $R^n \times R^n \times R^n \rightarrow R$ of dimension n , if it has an associated weighting vector W , with $\sum_{j=1}^n w_j = 1$ and $w_j \in [0, 1]$ and a weighting vector V that affects the WAD, with $\sum_{i=1}^n v_i = 1$ and $v_i \in [0, 1]$, such that:

$$\begin{aligned} \text{IOWAWAD}(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = \\ \beta \sum_{j=1}^n w_j b_j + (1 - \beta) \sum_{i=1}^n v_i |x_i - y_i| \end{aligned} \quad (8)$$

where b_j is the $|x_i - y_i|$ value of the IOWAWAD triplet $\langle u_i, x_i, y_i \rangle$ having the j th largest u_i , u_i is the order inducing variable, x_i is the i th argument of the set $X = \{x_1, \dots, x_n\}$, y_i is the i th argument of the set $Y = \{y_1, \dots, y_n\}$, and $\beta \in [0, 1]$.

Note that if the weighting vector is not normalized, i.e., $\hat{V} = \sum_{j=1}^n \hat{v}_j \neq 1$, then, the IOWAWAD operator can be expressed as:

$$IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = \frac{1}{\hat{V}} \sum_{j=1}^n \hat{v}_j b_j \quad (9)$$

If D is a vector corresponding to the ordered arguments b_j , we shall call this the ordered argument vector, and W^T is the transpose of the weighting vector, then, the IOWAWAD operator can be represented as follows:

$$IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = W^T D \quad (10)$$

Note that it is possible to distinguish between descending (DIOAWAD) and ascending (AIOAWAD) orders. The weights of these operators are related by $w_j = w_{n-j+1}^*$, where w_j is the j th weight of the DIOAWAD and w_{n-j+1}^* the j th weight of the AIOAWAD operator.

Note that $IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = 0$ if and only if $x_i = y_i$ for all $i \in [1, n]$. Note also that $IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle)$.

The IOWAWAD operator is monotonic, bounded and idempotent. It is monotonic because if $|x_i - y_i| \geq |s_i - t_i|$, for all $|x_i - y_i|$, then, $IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) \geq IOWAWAD(\langle u_1, s_1, t_1 \rangle, \dots, \langle u_n, s_n, t_n \rangle)$. It is bounded because the IOWAWAD aggregation is delimited by the minimum and the maximum. That is, $\text{Min}\{|x_i - y_i|\} \leq IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) \leq \text{Max}\{|x_i - y_i|\}$. It is idempotent because if $|x_i - y_i| = |x - y|$, for all $|x_i - y_i|$, then, $IOWAWAD(\langle u_1, x_1, y_1 \rangle, \dots, \langle u_n, x_n, y_n \rangle) = |x - y|$.

Note that we could study different measures for characterizing the weights, especially the OWA part, such as the entropy of dispersion, the divergence of W or the balance operator. The entropy of dispersion is defined as follows:

$$H(W) = - \sum_{j=1}^n w_j \ln(w_j) \quad (11)$$

For the balance operator, we get:

$$BAL(W) = \sum_{j=1}^n \left(\frac{n+1-2j}{n-1} \right) w_j \quad (12)$$

And for the divergence of W :

$$DIV(W) = \sum_{j=1}^n w_j \left(\frac{n-j}{n-1} - \alpha(W) \right)^2 \quad (13)$$

A further interesting issue to consider is the different families of IOWAWAD operators that are found in the weighting vector W and the coefficient β .

- If $\beta = 0$, we get the weighted Hamming distance.
- If $\beta = 1$, we get the IOWAD operator.
- The arithmetic weighted distance (if $w_j = 1/n$, for all j).
- The arithmetic IOWA distance (if $v_i = 1/n$, for all i).
- The normalized Hamming distance (if $v_i = 1/n$, for all i , and $w_j = 1/n$, for all j).
- The maximum weighted distance ($w_p = 1$ and $w_j = 0$, for all $j \neq p$, and $u_p = \text{Max}\{|x_i - y_i|\}$).
- The minimum weighted distance ($w_p = 1$ and $w_j = 0$, for all $j \neq p$, and $u_p = \text{Min}\{|x_i - y_i|\}$).
- The IOWAWA operator (if one of the sets is empty).
 - The OWAWA (the ordered position of the u_i is the same than the ordered position b_j).
 - The IOWA ($\beta = 1$).
 - The OWA ($\beta = 1$ and the ordered position of the u_i is the same than the ordered position b_j).
 - The weighted average ($\beta = 0$).
- The Hurwicz weighted distance criteria ($w_p = \alpha$, with $u_p = \text{Max}\{|x_i - y_i|\}$; $w_q = 1 - \alpha$, $u_q = \text{Min}\{|x_i - y_i|\}$; and $w_j = 0$, for all $j \neq p, q$).
- The step-IOWAWAD ($w_k = 1$ and $w_j = 0$, for all $j \neq k$).
- The olympic-IOWAWAD operator ($w_1 = w_n = 0$, and $w_j = 1/(n-2)$ for all others).
- The general olympic-IOWAWAD operator ($w_j = 0$ for $j = 1, 2, \dots, k, n, n-1, \dots, n-k+1$; and for all others $w_j = 1/(n-2k)$, where $k < n/2$).
- The S-IOWAWAD ($w_1 = (1/n)(1 - (\alpha + \beta) + \alpha)$, $w_n = (1/n)(1 - (\alpha + \beta) + \beta)$, and $w_j = (1/n)(1 - (\alpha + \beta))$ for $j = 2$ to $n-1$ where $\alpha, \beta \in [0, 1]$ and $\alpha + \beta \leq 1$).
- The centered-IOWAWAD (if it is symmetric, strongly decaying from the center to the maximum and the minimum, and inclusive).
- Etc.

Note that other families of IOWAWAD operators may be used following a similar methodology as it has been developed for the OWA operator and its extensions [1,5-20]. Moreover, we could extend this analysis to other types of distances such as the Euclidean (or quadratic) distance, the Minkowski (or generalized) distance and the quasi-arithmetic distance.

It is worth noting that some previous models already considered the possibility of using OWA operators and WAs in the same formulation. The main models are the WOVA operator [12-13] and the HA operator [14]. In this case, following these methodologies, we could

develop the induced WOWA distance (IWOWAD) operator and the induced hybrid averaging distance (IHAD) operator in a similar way as it has been done in the IOWAWAD operator. Another approach that could be analyzed is the concept of immediate probability [18] because we could use it as a weighted average (immediate weighted average). Thus, we could also develop the induced immediate weighted averaging distance (IIWAD) operator. Note that other models could also be considered under this framework where we are developing different extensions and generalizations. However, we believe that the approach that we are following seems to be the most complete. Therefore, we will not analyze in detail the other approaches although sometimes we will make some remarks about them.

4 ILLUSTRATIVE EXAMPLE

The IOWAWAD operator can be implemented in a wide range of applications where it is possible to use weighted averages, distance measures and OWA operators. Thus, we see that the applicability is incredibly broad because all the previous models and theories that use the WA can be revised and extended by using the IOWAWAD.

Summarizing some of the main fields where it is possible to apply the IOWAWAD operator, we can mention statistical theory, mathematics, economics, decision theory, engineering, soft computing and physics.

We focus on an application in decision making about selection of products. We analyze a customer that wants to buy a new product and he has five different brands that give him more or less the product he wants.

- A_1 = Product A.
- A_2 = Product B.
- A_3 = Product C.
- A_4 = Product D.
- A_5 = Product E.

In order to evaluate these products, the customer analyzes the products considering five main characteristics (or attributes):

- C_1 = Prize of the product.
- C_2 = Quality of the product.
- C_3 = Knowledge about the brand of the product.
- C_4 = Intuitive preference of the customer.
- C_5 = Other variables.

The results of the available products, depending on the characteristic C_i and the alternative A_k that the decision maker chooses, are shown in Table 1.

Table 1: Available products

	C_1	C_2	C_3	C_4	C_5
A_1	0.6	0.3	0.4	0.8	0.7
A_2	0.7	0.9	0.2	0.3	0.4
A_3	0.4	0.7	0.3	0.5	0.6
A_4	0.8	0.2	0.1	0.6	0.9
A_5	0.4	0.7	0.6	0.3	0.7

According to the objectives of the decision maker, he establishes the following ideal product. The results are shown in Table 2.

Table 2. Ideal product.

	C_1	C_2	C_3	C_4	C_5
I	0.9	1	0.8	0.8	0.9

Due to the fact that the attitudinal character is very complex because it involves the opinion of different members of the board of directors, the recruiters use order inducing variables to express it. The results are represented in Table 3.

Table 3: Order inducing variables

	C_1	C_2	C_3	C_4	C_5
U	14	26	22	17	19

With this information, it is possible to develop different methods based on the IOWAWAD operator for selecting a product. In this example, we will consider the NHD, the WHD, the IOWAD and the IOWAWAD operator. We will assume that $\beta = 0.5$ and the following weights: $W = (0.1, 0.1, 0.2, 0.2, 0.4)$ and $V = (0.3, 0.3, 0.2, 0.1, 0.1)$. The results are shown in Table 4.

Table 4: Aggregated results

	NHD	WHD	IOWAD	IOWAWAD
A_1	0.32	0.4	0.27	0.335
A_2	0.38	0.31	0.35	0.33
A_3	0.38	0.4	0.22	0.31
A_4	0.36	0.43	0.23	0.33
A_5	0.34	0.35	0.39	0.37

If we establish an ordering of the products, a typical situation if we want to consider more than one alternative, then, we get the results shown in Table 5. Note that the first alternative in each ordering is the optimal choice.

Table 5: Ordering of the products

	Ordering
NHD	$A_1 \succ A_5 \succ A_4 \succ A_2 = A_3$
WHD	$A_2 \succ A_5 \succ A_1 = A_3 \succ A_4$
IOWAD	$A_3 \succ A_4 \succ A_1 \succ A_2 \succ A_5$
IOWAWAD	$A_3 \succ A_2 = A_4 \succ A_1 \succ A_5$

As we can see, depending on the aggregation operator used, the ordering of the products may lead to different decisions.

5 CONCLUSIONS

We have presented the IOWAWAD operator. It is a new aggregation operator that uses a unified model between the WA and the OWA. Moreover, it uses distance measures in the analysis and a reordering process based on order-inducing variables. We have studied some of its main properties and we have found a lot of particular cases such as the normalized Hamming distance, the weighted Hamming distance, the OWAD, the IOWAD, the IOWAWA operator, the maximum weighted distance, the minimum weighted distance, the arithmetic weighted distance and the arithmetic IOWA distance.

We have also analyzed the applicability of this approach because we have seen that we can apply it in all the studies that use distance measures, weighted averages or OWAs. We have focussed on a decision making problem about product management.

In future research, we expect to develop further developments by adding new characteristics in the problem such as the use of other types of distance measures such as the Euclidean distance, the Minkowski distance or the quasi-arithmetic distance. Moreover, we will also analyze the use of probabilistic information and uncertain environments.

References

- [1] G. Beliakov, A. Pradera and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Berlin: Springer-Verlag, 2007.
- [2] J. Gil-Aluja, *The Interactive Management of Human Resources in Uncertainty*. Dordrecht: Kluwer Academic Publishers, 1998.
- [3] R.W. Hamming, Error-detecting and error-correcting codes. *Bell Systems Technical Journal*, 29:147-160, 1950.
- [4] A. Kaufmann, *Introduction to the Theory of Fuzzy Subsets*. Academic Press, New York, 1975.
- [5] J.M. Merigó, *New extensions to the OWA operators and its application in decision making* (In Spanish). PhD Thesis, Department of Business Administration, University of Barcelona, 2008.
- [6] J.M. Merigó, On the use of the OWA operator in the weighted average and its application in decision making. In: *Proceedings of the WCE 2009 Conference*, London, United Kingdom, pp. 82-87, 2009.
- [7] J.M. Merigó, A new aggregation operator and its application in the selection of researchers. In: *Proceedings of the EDULEARN 2009 Conference*, Barcelona, Spain, pp. 540-548, 2009.
- [8] J.M. Merigó, M. Casanovas, Induced aggregation operators in decision making with the Dempster-Shafer belief structure, *International Journal of Intelligent Systems*, 24:934-954, 2009.
- [9] J.M. Merigó, M. Casanovas, The induced generalized hybrid averaging operator and its application in financial decision making. *International Journal of Business, Economics, Finance and Management Sciences*, 1:95-101, 2009.
- [10] J.M. Merigó and A.M. Gil-Lafuente, The induced generalized OWA operator. *Information Sciences*, 179:729-741, 2009.
- [11] J.M. Merigó and A.M. Gil-Lafuente, New decision-making techniques and their application in the selection of financial products. *Information Sciences*, (to be published).
- [12] V. Torra, The weighted OWA operator. *International Journal of Intelligent Systems*, 12:153-166, 1997.
- [13] V. Torra and Y. Narukawa, *Modeling Decisions: Information Fusion and Aggregation Operators*. Berlin: Springer-Verlag, 2007.
- [14] Z.S. Xu and Q.L. Da, An overview of operators for aggregating information. *International Journal of Intelligent Systems*, 18:953-968, 2003.
- [15] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics B*, 18:183-190, 1988.
- [16] R.R. Yager, Families of OWA operators. *Fuzzy Sets and Systems*, 59:125-148, 1993.
- [17] R.R. Yager, Induced aggregation operators. *Fuzzy Sets and Systems*, 137:59-69, 2003.
- [18] R.R. Yager, K.J. Engemann, D.P. Filev, On the concept of immediate probabilities, *International Journal of Intelligent Systems*, 10:373-397, 1995.
- [19] R.R. Yager and D.P. Filev, Induced ordered weighted averaging operators. *IEEE Transactions on Systems, Man and Cybernetics B*, 29:141-150, 1999.
- [20] R.R. Yager and J. Kacprzyk, *The Ordered Weighted Averaging Operators: Theory and Applications*. Norwell: Kluwer Academic Publishers, 1997.

REGULAR MULTIDISTANCES

Javier Martín Gaspar Mayor

Department of Mathematics and Computer Science
University of the Balearic Islands
{javier.martin,gmayor}@uib.es

Abstract

The conventional definition of a distance (metric) can be extended to apply to collections of more than two elements. In this paper we introduce a new class of multidistances that we call regular. Regularity, which can be considered in-between weakness and strongness properties, is studied for some remarkable families of multidistances.

Keywords: distance, multidistance, Fermat multidistance, OWA-based multidistance, Sum-based multidistance.

1 INTRODUCTION

The conventional definition of distance over a space specifies properties that must be obeyed by any measure of "how separated" two points in that space are. However often one wants to measure "how separated" the members of a collection of more than two elements are. The usual way to do it is to combine the distance values for all pairs of elements in the collection, into an aggregate measure. Thus, given an Euclidean triangle (A, B, C) we can combine the distances AB, AC, BC using, for instance, a 3-dimensional OWA operator, say W . Then, we calculate the distance of A, B, C by means of the formula $D(A, B, C) = W(AB, AC, BC)$. It is clear that we have to choose the weighting vector of W such that the multi-argument distance function D satisfies a group of axioms that extend to some degree those for ordinary distance functions. Of course we can consider other procedures to measure how separated the vertices A, B, C are: in Euclidean geometry the Fermat point of a triangle (A, B, C) is the point F for which the sum of the distances from F to the vertices is as small as possible; i.e. it is the point F such that $FA + FB + FC$ is minimized. Then we can define $D(A, B, C) = FA + FB + FC$. For pairwise distances and related distance matrix see for example [1].

A recent paper [3] deals with the problem of aggregating pairwise distance values in order to construct a multidistance function [8, 5].

In addition to their intrinsic mathematical interest, multidistances have many potential applications. They can be directly incorporated into many domains: indistinguishability measures [4], distance-based clustering, pattern recognition, etc., where the extension of ordinary (binary) distances to multidimensional collections can be of interest. This is also the case of the so-called Jensen-Shannon divergence (JSD) which is a distance for probability distributions that have been used to treat different problems such as analysis of symbolic sequences, examination of texts in literature or separation of quantum states [6, 2].

In this paper we introduce a class of multi-argument distance functions that we call regular multidistances. After presenting some basic properties, regularity is studied for three relevant classes of multidistances: OWA-based multidistances, Sum-based multidistances and Fermat's multidistances.

2 PRELIMINARIES

Let us begin with the definition of multidistance, given in [5].

Definition 1 A function $D: \bigcup_{n \geq 1} X^n \rightarrow [0, \infty]$ is a multidistance on a set X if it fulfills the following properties for all n , for all $x_1, \dots, x_n, y \in X$:

- (m1) $D(x_1, \dots, x_n) = 0$ if and only if $x_1 = \dots = x_n$,
- (m2) $D(x_1, \dots, x_n) = D(x_{\pi(1)}, \dots, x_{\pi(n)})$ for any permutation π of $1, \dots, n$,
- (m3) $D(x_1, \dots, x_n) \leq D(x_1, y) + \dots + D(x_n, y)$.

We say that D is a strong multidistance if it fulfills (m1), (m2) and:

(m3') $D(x_1, \dots, x_n) \leq D(\mathbf{x}_1, \mathbf{y}) + \dots + D(\mathbf{x}_k, \mathbf{y})$ for any partition¹ $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ of $\mathbf{x} = (x_1, \dots, x_n)$, for all $\mathbf{y} \in \bigcup_{n \geq 1} X^n$.

Distances or multidistances taking values in the interval $[0, 1]$ are called *normalized*.

Remark 1

- 1) If D is a multidistance on X , then the restriction of D to X^2 , $D|_{X^2}$, is an ordinary distance on X .
- 2) An ordinary distance d on X can be extended in order to obtain a multidistance. For example, we can define the following function $D_M: \bigcup_{n \geq 1} X^n \rightarrow [0, \infty]$:

$$D_M(\mathbf{x}) = \begin{cases} 0 & \text{if } n = 1, \\ \max_{i < j} \{d(x_i, x_j)\} & \text{if } n \geq 2. \end{cases} \quad (1)$$

Conditions (m1), (m2) and (m3) are easy to verify and then D_M is a multidistance such that $D_M|_{X^2} = d$. It will be called maximum multidistance and it will be treated in Section 3.1.

We will denote by $\mathcal{D}_d$ the set of multidistances D such that $D|_{X^2} = d$.

Multidistances allow us to define balls centered at lists [5].

Definition 2 Given a multidistance D and a list $\mathbf{x} = (x_1, \dots, x_n) \in \bigcup_{n \geq 1} X^n$, the closed ball of center $\mathbf{x}$ and radius $r \in \mathbb{R}$ is the set:

$$B(\mathbf{x}, r) = \{y \in X: D(\mathbf{x}, y) \leq D(\mathbf{x}) + r\}. \quad (2)$$

This definition is an extension of the usual one for balls centered at points.

3 REGULAR MULTIDISTANCES

This section deals with the property of regularity for multidistances. First we give the definition and some properties and then, the regularity of several classes of multidistances is studied.

Definition 3 A multidistance D is *regular* if the following property holds, for all $\mathbf{x} \in X^n$ and $y \in X$:

$$D(\mathbf{x}) \leq D(\mathbf{x}, y). \quad (3)$$

That is, the multidistance between the elements of a list can not decrease if we add an element.

¹A partition of $\mathbf{x} = (x_1, \dots, x_n) \in X^n$ is a set of lists $\{\mathbf{x}_i = (x_{i_1}, \dots, x_{i_{n_i}}) \in X^{n_i}; i = 1, \dots, k\}$ chosen in such a way that the set of subindexes $\{\{i_1, \dots, i_{n_i}\}, i = 1, \dots, k\}$ is a partition of $\{1, \dots, n\}$.

Remark 2 Regularity has a precise meaning in terms of the balls centered at lists: a multidistance is regular if and only if the balls with negative radius are empty.

Regularity can be seen as a first step of strongness: if we consider a one-part partition, condition (m3') reduces to:

$$D(\mathbf{x}) \leq D(\mathbf{x}, y), \quad (4)$$

but this is equivalent to (3).

As a consequence, we have the following result, relating these two properties.

Proposition 1 Any strong multidistance is regular.

A particular case of interest for strong multidistances is when the added element is one of the list.

Proposition 2 If D is a strong multidistance on X , then

$$D(\mathbf{x}) = D(\mathbf{x}, x_i) \quad (5)$$

for all $\mathbf{x} \in X^n$, for any element x_i of the list $\mathbf{x}$.

Proof: Apply condition (m3') to the list $(\mathbf{x}, x_i)$, the partition $\{(x_i, x_i), (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)\}$ and the added element x_i . As $D(x_i, x_i, x_i) = 0$ and $D(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, x_i) = D(\mathbf{x})$, we have $D(\mathbf{x}, x_i) \leq D(\mathbf{x})$.

The condition of regularity $D(\mathbf{x}) \leq D(\mathbf{x}, x_i)$ completes the proof. ■

Therefore, repeated elements are superfluous when dealing with strong multidistances.

Basic properties of regularity are the following.

Proposition 3 Let D and D' be regular multidistances on X .

- 1) $D + D'$ is also a regular multidistance on X .
- 2) If $k > 0$, then kD is also a regular multidistance on X .
- 3) $\frac{D}{1+D}$ and $\min\{1, D\}$ are normalized regular multidistances on X .

From now on we consider a metric space (X, d) with cardinality $|X| \geq 2$. We are going to study the regularity of some remarkable classes of multidistances belonging to $\mathcal{D}_d$.

3.1 Regularity of the OWA-based multidistances

Let $\mathcal{W} = \{W_n; n \geq 2\}$ be a family of OWAs [7], where the weights of the $\binom{n}{2}$ -dimensional OWA W_n

$\omega_1^n, \dots, \omega_{\binom{n}{2}}^n$, with $\omega_1^n + \dots + \omega_{\binom{n}{2}}^n = 1$, are applied to the list of the $\binom{n}{2}$ pairwise distances arranged in an increasing order.

We can define a function $D_W: \bigcup_{n \geq 1} X^n \rightarrow [0, \infty]$ in this way:

$$D_W(\mathbf{x}) = \begin{cases} 0 & \text{if } n = 1, \\ W_n(\overbrace{d(x_1, x_2), \dots, d(x_{n-1}, x_n)}^{\binom{n}{2}}) & \text{if } n \geq 2, \end{cases} \quad (6)$$

for all $\mathbf{x} = (x_1, \dots, x_n) \in X^n$.

A special case is when $W_n = \max$, with lists of weights $(1, 0, \dots, 0)$ for all n , obtaining the maximum multidistance D_M given in (1).

Lemma 1 *The multidistance D_M is strong.*

Proof. Let $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ be a partition of $\mathbf{x}$. We can suppose that the maximum distance is reached at x_1, x_2 , that is, $D_M(\mathbf{x}) = d(x_1, x_2)$, with $x_1 \in \mathbf{x}_r$, $x_2 \in \mathbf{x}_s$ and $\mathbf{x}_r \neq \mathbf{x}_s$. So, for any component y of the list $\mathbf{y}$,

$$\begin{aligned} D_M(\mathbf{x}) = d(x_1, x_2) &\leq d(x_1, y) + d(x_2, y) \\ &\leq D_M(\mathbf{x}_r, \mathbf{y}) + D_M(\mathbf{x}_s, \mathbf{y}) \\ &\leq \sum_{i=1}^k D_M(\mathbf{x}_i, \mathbf{y}). \end{aligned}$$

And if x_1, x_2 belong to the same part, say $\mathbf{x}_r$, then

$$\begin{aligned} D_M(\mathbf{x}) = d(x_1, x_2) &\leq D_M(\mathbf{x}_r, \mathbf{y}) \\ &\leq \sum_{i=1}^k D_M(\mathbf{x}_i, \mathbf{y}). \end{aligned}$$

Therefore condition (m3') is fulfilled and hence the multidistance D_M is strong. ■

The following result gives a necessary and sufficient condition for these OWA-based functions to be multidistances.

Proposition 4 *The function D_W is a multidistance if and only if, for all $n \geq 3$:*

$$\omega_1^n + \dots + \omega_{n-1}^n > 0. \quad (7)$$

Proof: Condition (m2): d and W_n are symmetric and then D_W also is.

As D_M is a multidistance (lemma 1),

$$D_W(\mathbf{x}) \leq D_M(\mathbf{x}) \leq \sum_{i=1}^n d(x_i, y)$$

for all $y \in X$, and so (m3) holds.

Finally, condition (m1). Let us suppose that D_W fulfills it. If we take the list $(a, \overbrace{b, \dots, b}^{n-1})$, with $d(a, b) =$

$l > 0$, then:

$$\begin{aligned} 0 < D(a, \overbrace{b, \dots, b}^{n-1}) &= W_n(\overbrace{l, \dots, l}^{n-1}, \overbrace{0, \dots, 0}^{\binom{n-1}{2}}) \\ &= l \cdot (\omega_1^n + \dots + \omega_{n-1}^n), \end{aligned}$$

that is, $\omega_1^n + \dots + \omega_{n-1}^n > 0$.

Reciprocally, let us suppose now (7).

Obviously, if $x_1 = \dots = x_n$ then $D(\mathbf{x}) = 0$. And if there exist i, j such that $x_i \neq x_j$, then $d(x_i, x_j) > 0$. In this case, for all $k \neq i, j$ either $d(x_i, x_k) > 0$ or $d(x_j, x_k) > 0$; that is, there are at least $n-1$ non-zero pairwise distances and so $D(\mathbf{x}) \neq 0$. ■

The next proposition gives a necessary condition for an OWA-based multidistance to be regular.

Proposition 5 *If a function D_W defined by a family of OWAs $\mathcal{W} = \{W_n; n \geq 2\}$ is regular, then*

$$\omega_1^n + \dots + \omega_{n-1}^n = 1, \quad (8)$$

for all $n \geq 3$.

Proof: We will prove it by induction. Let $a, b \in X$ such that $d(a, b) = l > 0$.

The base case is $n = 3$. For the lists (a, b) and (a, b, b) the condition of regularity is $l \leq \omega_1^3 l + \omega_2^3 l$, that is, $\omega_1^3 + \omega_2^3 = 1$.

Now the inductive step. Suppose that (8) holds for n .

Consider the lists $(a, \overbrace{b, \dots, b}^{n-1})$ and $(a, \overbrace{b, \dots, b}^n)$, whose

lists of ordered pairwise distances are $(\overbrace{l, \dots, l}^{n-1}, \overbrace{0, \dots, 0}^{\binom{n-1}{2}})$

and $(\overbrace{l, \dots, l}^n, \overbrace{0, \dots, 0}^{\binom{n}{2}})$ respectively. The condition of regularity in this case is:

$$\omega_1^n + \dots + \omega_{n-1}^n \leq \omega_1^{n+1} + \dots + \omega_n^{n+1},$$

and so $\omega_1^{n+1} + \dots + \omega_n^{n+1} = 1$. ■

Remark 3 *Condition (8) is necessary but it does not suffice. Let us consider, for example, the OWAs*

$$\begin{aligned} W_3 &= (1, 0, 0), \\ W_4 &= (0, 0, 1, 0, 0, 0), \end{aligned}$$

fulfilling it and belonging to a family $\mathcal{W}$ which defines a multidistance D_W .

For the points $(0, 1), (1, 0), (0, 0)$ in the Euclidean plane we have the following values, according to (6):

$$\begin{aligned} D_W((0, 1), (1, 0), (0, 0)) &= W_3(\sqrt{2}, 1, 1) \\ &= \sqrt{2}, \end{aligned}$$

$$D_W((0, 1), (1, 0), (0, 0), (0, 0)) = W_4(\sqrt{2}, 1, 1, 1, 1, 0) = 1,$$

and so D_W is not regular.

In fact, condition (8) is sufficient if and only if the range of the distance d consists exactly of two values.

Let us conclude with an example.

Example 1 As we have seen in lemma 1, D_M , with OWAs $(1, 0, \dots, 0)$, is a strong multidistance. On the opposite side, the OWAs $(0, \dots, 0, 1)$ define the function D_m :

$$D_m(\mathbf{x}) = \begin{cases} 0 & \text{if } n = 1, \\ \min_{i < j} \{d(x_i, x_j)\} & \text{if } n \geq 2, \end{cases} \quad (9)$$

which is not a multidistance.

Between them, the multidistance $\frac{D_M + D_m}{2}$, corresponding to the OWAs $(\frac{1}{2}, 0, \dots, 0, \frac{1}{2})$, $n \geq 2$, which is not regular.

Finally, if we take the OWAs $(\frac{1}{2}, \frac{1}{2}, 0, \dots, 0)$, $n \geq 2$, then the result is a multidistance D_W : the semisum of the two greatest pairwise distances. It is regular but in general it is not strong: if there exist three points $a, b, c \in X$ such that $d(a, b) < d(a, c) < d(b, c)$, then

$$D_W(a, b, c) = \frac{d(a, c) + d(b, c)}{2}, \\ D_W(a, b, c, c) = d(b, c),$$

and D_W is not strong because it does not fulfill (5).

3.2 Regularity of the sum-based multidistances

The following multidistances have been defined in [5]. They are based on the sum of the pairwise distance values for all pairs of elements of the list, multiplied by a factor λ which depends on its length.

Definition 4 The sum-based multidistances are the functions $D_\lambda: \bigcup_{n \geq 1} X^n \rightarrow [0, \infty]$ defined by

$$D_\lambda(\mathbf{x}) = \begin{cases} 0 & \text{if } n = 1, \\ \lambda(n) \sum_{i < j} d(x_i, x_j), & \text{if } n \geq 2, \end{cases} \quad (10)$$

where:

- (i) $\lambda(2) = 1$,
- (ii) $0 < \lambda(n) \leq \frac{1}{n-1}$ for any $n > 2$.

With respect to the regularity of these multidistances, we have the following result.

Proposition 6 The multidistances D_λ are regular if and only if $\lambda(n) = \frac{1}{n-1}$ for all $n \geq 2$.

Proof: First we prove by induction that if the multidistance D_λ is regular, then $\lambda(n) = \frac{1}{n-1}$ for all $n \geq 2$.

The base case holds by definition: $\lambda(2) = 1$.

Suppose now that $\lambda(n-1) = \frac{1}{n-2}$. If we take lists of the form $(a, \dots, a, b)$ we have:

$$D_\lambda(\overbrace{(a, \dots, a)}^{n-2}, b) = \lambda(n-1) \cdot (n-2)l = l$$

and

$$D_\lambda(\overbrace{(a, \dots, a)}^{n-1}, b) = \lambda(n) \cdot (n-1)l.$$

The condition of regularity (3) is

$$l \leq \lambda(n) \cdot (n-1)l,$$

that is, $\lambda(n) \geq \frac{1}{n-1}$ or, taking into account (ii) in definition 4, $\lambda(n) = \frac{1}{n-1}$.

Let us see now that the multidistance:

$$D_\lambda(\mathbf{x}) = \begin{cases} 0 & \text{if } n = 1, \\ \frac{\sum_{i < j} d(x_i, x_j)}{n-1}, & \text{if } n \geq 2, \end{cases} \quad (11)$$

is regular. The condition of regularity (3) is:

$$\frac{\sum_{i < j} d(x_i, x_j)}{n-1} \leq \frac{\sum_{i < j} d(x_i, x_j) + \sum_{i=1}^n d(x_i, y)}{n}$$

for any $y \in X$, that is,

$$\sum_{i < j} d(x_i, x_j) \leq (n-1) \sum_{i=1}^n d(x_i, y).$$

But this is fulfilled:

$$\sum_{i < j} d(x_i, x_j) \leq \sum_{i < j} (d(x_i, y) + d(x_j, y)) \\ = (n-1) \sum_{i=1}^n d(x_i, y).$$

■

Remark 4

i) The arithmetic mean of the pairwise distances corresponds to the multidistance D_λ with $\lambda(n) = \frac{1}{\binom{n}{2}}$, $n \geq 2$. It coincides with the multidistance D_W that has the OWAs $W_n = (\frac{1}{\binom{n}{2}}, \dots, \frac{1}{\binom{n}{2}})$, $n \geq 2$, and it is not regular.

ii) The multidistance D_λ , with $\lambda = \frac{1}{n-1}$, is not strong. Consider for example the lists (a, a, b) and (a, a, b, b) , with $d(a, b) > 0$.

$$D_\lambda(a, a, b) = d(a, b), \\ D_\lambda(a, a, b, b) = \frac{4}{3}d(a, b),$$

but this means, see proposition 2, that D_λ is not strong.

iii) There are not multidistances D_λ fulfilling (5): $D(\mathbf{x}) = D(\mathbf{x}, x_i)$. For example, if $d(a, b) > 0$ then it should be

$$D_\lambda(a, a, b) = D_\lambda(a, a, a, b) = D_\lambda(a, a, b, b),$$

but

$$\begin{aligned} D_\lambda(a, a, a, b) &= \lambda(4) \cdot 3d(a, b), \\ D_\lambda(a, a, b, b) &= \lambda(4) \cdot 4d(a, b), \end{aligned}$$

that is, $\lambda(4) = 0$ and D_λ is not a multidistance.

3.3 Regularity of the multidistance of Fermat

This multidistance was defined in [5] and it is based on the idea of the Fermat point explained in the introduction.

Definition 5 The multidistance of Fermat is the function $D_F: \bigcup_{n \geq 1} X^n \rightarrow [0, \infty]$ defined by:

$$D_F(x_1, \dots, x_n) = \min_{x \in X} \left\{ \sum_{i=1}^n d(x_i, x) \right\}. \quad (12)$$

The following result states the regularity of this multidistance.

Proposition 7 The multidistance of Fermat D_F is regular.

Proof:

$$\begin{aligned} D_F(x_1, \dots, x_n) &= \min_{x \in X} \left\{ \sum_{i=1}^n d(x_i, x) \right\} \\ &\leq \min_{x \in X} \left\{ \sum_{i=1}^n d(x_i, x) + d(y, x) \right\} \\ &= D_F(x_1, \dots, x_n, y). \end{aligned}$$

■

Once the regularity of the multidistances of Fermat has been established, let us deal with the strongness.

Proposition 8 The multidistance of Fermat D_F is not strong.

Proof: Consider the lists (a, a, b) and (a, a, b, b) with $d(a, b) > 0$.

$$\begin{aligned} D_F(a, a, b) &= \min_{x \in X} \{2d(a, x) + d(b, x)\} \\ &= d(a, b), \end{aligned}$$

because the minimum is reached at $x = a$, and

$$\begin{aligned} D_F(a, a, b, b) &= 2 \min_{x \in X} \{d(a, x) + d(b, x)\} \\ &\geq 2d(a, b), \end{aligned}$$

for all $x \in X$, which means, taking into account proposition 2, that D_F is not strong. ■

4 CONCLUSIONS

The property of regularity for multidistances has been introduced. This property, which is weaker than strongness, has been study for different families of multidistances, obtaining in each case complete characterizations.

Acknowledgements

The authors acknowledge the support of the Govern Balear grant PCTIB2005GC1-07 and the Spanish DGI grant MTM2009-10962.

References

- [1] L.M. Blumenthal (1970). Theory and applications of distance geometry, 2nd ed. New York, Chelsea Publishing Co.
- [2] P.W. Lamberti, A.P. Majtey, A. Borras, M. Casas, A. Plastino (2008). Metric character of the quantum Jensen-Shannon divergence. Physical Review 77.
- [3] J. Martín, G. Mayor (2009). Aggregating pairwise distance values. Proceedings of the EUROFUSE 2009, Pamplona (Spain), pp. 147-152.
- [4] J. Martín, G. Mayor (2009). An axiomathical approach to T -multi-indistinguishabilities. Proceedings of the IFSA-EUSFLAT 2009, Lisbon (Portugal), pp. 1723-1728.
- [5] J. Martín, G. Mayor (2009). How separated Palma, Inca and Manacor are? Proceedings of the AGOP 2009, Palma de Mallorca (Spain), pp. 195-200.
- [6] M.A. Ré, D.P. Prato, P.W. Lamberti (2007). Discriminación de distribuciones de probabilidad usando distancias entrópicas. Anales AFA 19, pp. 36-39.
- [7] R.R. Yager (1988). On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man and Cybernetics 18, pp. 183-190.
- [8] D.H. Wolpert (2004). Metrics for more than two points at once. International Conference on Complex Systems ICCS 2004, Y. Bar-Yam (Ed.), Perseus books, in press.

SOBRE FUNCIONES Y REGLAS DE AGREGACIÓN

Javier Montero¹, Daniel Gómez², Victoria López³, J. Tinguaro Rodríguez¹, Begoña Vitoriano¹

¹ Facultad de Matemáticas, Universidad Complutense de Madrid {monty,jtrodrig,bvitoriano}@mat.ucm.es

² Escuela de Estadística, Universidad Complutense de Madrid, dagomez@estad.ucm.es

³ Facultad de Informática, Universidad Complutense de Madrid, ylopez@fdi.ucm.es

Resumen

En este artículo se analiza la definición actualmente al uso de función de agregación, haciendo hincapié en que, a pesar de considerar el problema de la variabilidad de la dimensión de la información a agregar, dicha definición no cierra la posibilidad de disfuncionalidades obvias, subyaciendo en su concepción un potencial abuso del formalismo matemático que, junto con las necesarias consideraciones de implementación práctica pueden llevar a reducir a una simple fórmula lo que es un complejo concepto. En este trabajo propondremos las bases para una definición alternativa que tenga en cuenta los objetivos y las limitaciones que se observan en las aplicaciones de las funciones de agregación dentro del contexto difuso.

Palabras Clave: Funciones de Agregación, Reglas de Agregación, Operadores de Agregación, Recursividad.

1 FUNCIONES DE AGREGACIÓN

La agregación de información es, en sentido amplio, una de las piezas clave de todo sistema de ayuda a la decisión, tanto en cuanto elabora o prepara una visión global de la situación sobre la que se va a tomar una decisión (ver [24]). Con frecuencia esta agregación implica una reducción en la dimensión de la información original, manteniendo sin embargo su formato, aunque esta restricción no es ni mucho menos esencial. De hecho, cabría esperar de manera natural que el grado de complejidad del problema sea superior a la complejidad de cada una de sus descripciones parciales. Así, a veces la mejor descripción de un estado psicológico se logra con

una imagen, y una larga estadística numérica se comprende con una gráfica.

En el contexto de operadores difusos, se suele considerar que cada información parcial viene dada en términos de un número dentro del intervalo unidad, para asignar a cada vector de informaciones parciales otro número en el intervalo unidad, que actúa como valor agregado.

De este modo, si nuestro sistema de recogida de datos sabemos alberga indefectiblemente un número determinado de ítems, nuestra agregación puede limitarse a darnos la solución a este problema concreto. Pero en la mayor parte de los casos no podemos garantizar que el cardinal de la información se va a mantener, pues a veces hay datos que se pierden o que deben ser eliminados por errores en la observación o la transmisión, y a veces nos llega información adicional con la que no contábamos, por error en el diseño o por cambios en la realidad observada.

La limitación funcional a una determinada dimensión es sin duda un grave problema. Así, por ejemplo, los operadores OWA son de difícil implementación práctica si mantenemos la definición propuesta por Yager [25], mientras no se aporte un protocolo de actuación ante la pérdida o aparición inesperada de datos [7]. Una propuesta de solución se puede encontrar en [8] (ver también [16]).

Sin embargo, la solución estándar a este problema ha sido con frecuencia mimetizar de manera abusiva la situación en el contexto nítido, donde los valores agregados y la propia agregación toman valores en $\{0,1\}$. Dado que en el caso nítido sólo hay dos operadores razonables (los que corresponden al mínimo y al máximo son los únicos operadores no decrecientes que hacen relevante cada unidad de información recibida), nada más natural que hacerle escoger al decisor entre estos dos contextos (conjunción o disyunción) para aplicar sucesivamente el mismo operador, que es en ambos casos conmutativo y

asociativo. Por tanto, podría parecer natural imponer que la estructura básica para analizar una información que toma valores en $[0,1]$ debe estar formada por dos operadores binarios (quizá interconectados con una *negación*) verificando las propiedades de monotonía no decreciente, conmutatividad y asociatividad, además de algunas condiciones de contorno como que la agregación de un vector formado por todo 1 sea 1 o que la agregación de un vector formado por todo 0 sea 0 (ver una discusión de algunas propiedades posibles en [14]). Pero esta aproximación deja fuera de alcance muchas de las agregaciones más habituales, como la media o la mediana. El clásico resultado de Fung y Fu [12], por ejemplo, exige romper con la hipótesis de que toda agregación se puede resolver con un único operador asociativo [22] o hacer una interpretación menos estricta de cuál es el nivel de información en los que los operadores actúan (la media se puede calcular como subproducto de la suma, que sí que resuelve todos sus cálculos a partir de las sumas de parejas de números).

Y es que es obvio que fuera del contexto nítido cada problema de agregación no tiene que poder explicarse mediante la aplicación sucesiva de un mismo operador binario.

Así es como llegamos, dado que tenemos que ser capaces de resolver la agregación sin conocer *a priori* la cardinalidad de los datos, a la definición actualmente estándar de función de agregación: una familia de operadores $A(n)$ que sea cuál sea el cardinal n de los datos nos diga cómo obtener el valor agregado de esos n datos (ver, por ejemplo, [6]).

2 CRITICA A LA DEFINICIÓN ACTUAL

La cuestión que queremos suscitar formalmente en este trabajo es hasta qué punto la definición al uso responde a los fines y propiedades que se esperan de una función de agregación.

Pues si bien es cierto que una condición *sine qua non* para ser función de agregación es que contenga las instrucciones para calcular la agregación de la información, sea cual sea su cardinalidad, hay dos cuestiones a las que dicha definición no responde, y que toda definición debe al menos considerar, toda vez que se ha comprobado su consistencia lógica: su coherencia conceptual y la viabilidad de su implementación práctica.

2.1. COHERENCIA DE UNA FUNCIÓN DE AGREGACIÓN

Por coherencia no nos referimos por supuesto a la consistencia lógica, sin la que no hablaríamos siquiera de

definición. Por coherencia entendemos la necesidad de *armonización* entre las distintas componentes de una definición.

Por ejemplo, las ternas de De Morgan (ver por ejemplo [10]) se podrían haber definido como ternas simplemente formadas por una *t-norma*, una *t-conorma* y una *negación* [11], dando así capacidad de resolver en otro contexto las tres operaciones básicas observadas en el caso binario (conjunción, disyunción y negación, y de nuevo hacemos notar que en el caso nítido sólo hay una aplicación que invierta el orden de los dos estados $\{0,1\}$, unicidad que no aparece en $[0,1]$, por ejemplo). Pero las ternas de De Morgan se definen imponiendo una propiedad adicional que armoniza esas tres operaciones, como ocurre en el caso nítido (la conjunción de unos datos se puede obtener como la negación de la disyunción de las negaciones de esos datos).

Y es que si no existe una condición adicional que armonice las componentes de una estructura, en realidad no tenemos una estructura. Si una terna de De Morgan estuviese formada simplemente por una *t-norma*, una *t-conorma* y una *negación*, probablemente hablaríamos de una *t-norma*, una *t-conorma* y una *negación*, sin asignar al conjunto un concepto de rango superior.

Sin armonización, sin conexión, sin estructura, no tenemos un edificio, tenemos sólo ladrillos (ver por ejemplo [23] para trasladar a otro contexto la importancia de no obviar el problema de la estructura subyacente).

Del mismo modo, un Espacio de Probabilidad está formado por un espacio muestral, un álgebra de sucesos y una probabilidad, pero estas tres piezas no son arbitrarias sino que tienen que encajar (la probabilidad se define sobre un álgebra de sucesos que tiene que ser isomorfo a un álgebra de subconjuntos del espacio muestral).

Así que, para que una función de agregación merezca ser una definición, debería incorporar alguna propiedad adicional, una restricción que le de un sentido concreto a esa secuencia de operadores $A(2), A(3), A(4), \dots$, para que se pueda entender que detrás de cada función de agregación hay un concepto unificador.

Parecería cuando menos extraño proponer, por ejemplo, el cálculo del mínimo para dos números, la media aritmética para tres números, la media geométrica para cuatro y la mediana para cinco.

Y tampoco parecería apropiado mantener un enfoque excesivamente formal exigiendo una unidad vía fórmula matemática, pues el ejemplo anterior, por ejemplo, admite una expresión matemática compacta trivial.

2.2. IMPLEMENTACIÓN DE UNA FUNCIÓN DE AGREGACIÓN

El segundo aspecto que toda definición debe tener en cuenta tiene que ver con su implementación. Es decir, la definición debe permitir la existencia de un procedimiento ejecutable que permita obtener el resultado, y en tiempo razonable.

De entrada, es interesante recordar que la función de agregación debe quedar total y unívocamente caracterizada. Y puesto que se trata de una familia infinita de operadores, podría sugerirse de nuevo un enfoque estrictamente formal (determinación de expresiones matemática válidas, para eliminar por ejemplo aquellas expresiones que implican cálculos conceptualmente diferentes en función del cardinal de los datos).

Pero en cualquier caso acabaremos teniendo que demostrar que los cálculos implicados son asumibles, lo cual sugiere el estudio de sus algoritmos de construcción (desde luego no únicos para una función de agregación, ni necesariamente de la misma complejidad).

Si una definición tiene asociado un algoritmo de cálculo con complejidad exponencial, esa definición no es implementable en la práctica. Una definición de función de agregación no es *operativa* mientras no se haya determinado efectivamente un algoritmo de complejidad polinomial que permita su ejecución real.

Esta característica de *operatividad* fue en su momento el punto de partida para definir lo que en su momento llamamos reglas recursivas [3], que permitían el cálculo a través de la aplicación secuencial de operadores binarios, y donde se entendía por recursividad la capacidad de ejecutar el cálculo ante un dato adicional a partir del resultado anteriormente obtenido, y en las dos direcciones de modo que aparecía subyacente una ecuación general de asociatividad [2,21]. Quizá el resultado más importante con esta aproximación basada en reglas recursivas fue comprobar como bajo determinadas condiciones los primeros operadores podían llegar a determinar toda la secuencia [9]. Pero aunque dicha noción de regla recursiva [3,9] contiene importantes funciones de agregación, es importante hacer notar que el cálculo recursivo puede existir en otros formatos diferentes al propuesto en [3] (la recursividad considerada en las reglas recursivas introducidas en [3] implica un caso específico entre lo que se conoce como algoritmo recursivo), y sobre todo, que en cualquier caso deja fuera otras importantes funciones de agregación. La mediana, por ejemplo, exige con cada nuevo dato reordenar el conjunto de todos los datos, y ninguna reducción de esta información permite el cálculo siguiente (el algoritmo de ordenación añade un factor multiplicativo de complejidad menor que cuadrática y, por lo tanto, sería ejecutable).

3 DEFINICIÓN ALGORÍTMICA DE UNA REGLA DE AGREGACIÓN

De todo lo anterior se justifica el intento de una definición de regla de agregación como una función de agregación basada precisamente en la existencia de un algoritmo apropiado que garantice la comprobación de su coherencia y su cálculo efectivo.

Definición.- Una regla de agregación es una función de agregación que tiene asociada un procedimiento de cálculo específico, que a partir de un número finito de primeros casos, y mediante un algoritmo polinomial iterativo, asigna a todo vector finito de números en el intervalo $[0,1]$ un valor agregado en dicho intervalo.

Nótese que no imponemos condiciones adicionales, ni siquiera la de monotonía (ver por ejemplo [5]). Además, por supuesto que la restricción al intervalo unidad no es ni mucho menos esencial, ni siquiera que todas esas valoraciones sean numéricas, ni que todas ellas tomen valores dentro de un mismo espacio.

En cualquier caso, debe existir al menos un algoritmo de cómputo eficiente y efectivo que resuelva el problema de agregación asociado en tiempo aceptable.

Se trata ésta de una definición de inspiración práctica que intenta distinguir las propuestas con dificultades operativas, con aplicaciones muy específicas y un tanto inusuales. Pudiera parecer una definición casi hueca desde un punto de vista práctico, dado que las funciones de agregación con un algoritmo no polinomial no son operativas, y en general sólo se suele implementar lo que es operativo. Lo que se ha podido implementar es casi por definición polinomial (un algoritmo no polinomial permite, sin embargo, “ocultar” a otros usuarios la recuperación de la información inicial, y por tanto no son siempre desdeñables).

De esta manera, la función de agregación queda caracterizada por un algoritmo de cálculo que se le ha asociado en su definición, y cuyas propiedades se transfieren nominalmente a la función de agregación (iteratividad, recursividad, eficiencia, etcétera). Queda de esta manera habilitada la posibilidad de una clasificación de reglas en función de las propiedades del algoritmo disponible (la búsqueda del algoritmo más eficiente será por supuesto clave). En concreto, nuestras reglas de agregación se han caracterizado en la definición anterior como aquellas funciones de agregación a las que se les ha podido asociar un algoritmo polinomial.

Así podremos encontrar, por ejemplo, que muchas reglas de agregación necesitan simplemente definir cómo se

agregan los dos primeros números. El número de casos necesarios iniciales para “lanzar” el algoritmo puede establecer una primera distinción entre tipos de reglas de agregación. Pero sin duda la característica limitativa está en que a partir de un momento predeterminado se aplique un mismo procedimiento una y otra vez, las veces que sea necesario hasta procesar el conjunto de datos disponible, que será finito en cada caso, aunque sea desconocido *a priori* (quedan muy lejos de nuestro enfoque polinomial aquéllos planteamientos que incluyen la agregación de infinitos datos).

Además, la catalogación de una regla de agregación está supeditada al algoritmo disponible (el algoritmo polinomial puede existir y no haberlo encontrado, por ejemplo). Como en complejidad algorítmica (ver [13], por ejemplo), es el análisis del algoritmo concreto, o su asimilación a otro análogo, lo que permite su catalogación como polinomial o no polinomial.

Y tanto la monotonía como la conmutatividad, las condiciones de contorno o las demás propiedades habituales, podrán o no ser convenientes en cada contexto. Como en fiabilidad de sistemas (ver por ejemplo [4]), la mayoría de los sistemas mejoran cuando sus componentes mejoran su comportamiento, pero no siempre (ver de nuevo [14]).

4 COMENTARIOS FINALES

En opinión de los autores, la definición al uso de función de agregación debe ser revisada con la mayor urgencia, dado que considerándola errónea, por estar vacía de contenido real, cuanto más tiempo se tarde más difícil será de desinstalarla. Baste aquí hacer una referencia a lo traumática que ha sido la discusión sobre la utilización de la palabra *intuicionista* en los modelos de Atanassov [1,10] después de contar con miles de referencias (ver también [23]), y que la palabra *difuso* se ha mantenido en castellano para referirse a la lógica *fuzzy*, a pesar de ser lingüísticamente incorrecta, y a pesar incluso de que el propio L.A. Zadeh, fundador de la teoría, ha reconocido en alguna ocasión que cuando se dio cuenta de que la palabra *fuzzy* no había sido una buena elección en lengua inglesa, ya no pudo corregirla porque dicha palabra ya se había instalado totalmente en la comunidad científica.

La propuesta que presentamos está en consonancia con la aproximación *operacional* que se planteaba con la propuesta recursiva de Cutello y Montero [9], generalizando su propuesta y dejando la validez de una regla de agregación en manos de la determinación de un algoritmo polinomial a partir del cual se puedan hacer todos y cada uno de los cálculos que vayan siendo requeridos, y donde se pueden aplicar técnicas de verificación para comprobar su consistencia interna

[19,20], garantizando así la presencia de un concepto, a partir de una casuística básica (cómo se agregan los primeros casos) y la repetición sistemática de un procedimiento único a partir de ellos, hasta agotar los datos disponibles o de interés puntual.

Esta aproximación parece contener las funciones de agregación más usuales, abriendo la posibilidad de distinguir casos computacionalmente complejos, introduciendo algunos controles en principio naturales.

El plan de trabajo subsiguiente, toda vez que la clave de nuestra enfoque está en el algoritmo de ejecución de la función de agregación, será trasladar a este concepto todos los matices conocidos en el diseño, análisis y verificación de algoritmos (ver, por ejemplo, [15, 17, 18, 19]).

Pero hemos de resaltar la relevancia de que sea el algoritmo de ejecución quien en última instancia justifica ciertos conceptos asociados a diferentes tipos de función de agregación. Las reglas de agregación en este trabajo definidas, y que destacamos como el campo estándar de trabajo, son nada más que las funciones de agregación polinomiales (es decir, a las que se les ha podido asociar un algoritmo de complejidad temporal polinomial y por lo tanto, eficiente).

Agradecimientos

Este trabajo se ha realizado dentro del proyecto TIN2006-06190 del Plan Nacional de Investigación, y el Grupo de Investigación 910149 de la Universidad Complutense de Madrid.

Referencias

- [1] K.T. Atanassov. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 20, 87-96, 1986.
- [2] J. Aczél. *Lectures on Functional Analysis and their Applications* (Academic Press, New York, 1966).
- [3] A. Amo, J. Montero, E. Molina. Representation of consistent recursive rules. *European Journal of Operational Research* 130, 29-53, 2001.
- [4] R.E. Barlow, F. Proshan. *Statistical Theory of Reliability and Life Testing* (Holt, Rinehart and Winston, New York, 1975).
- [5] G. Beliakov, A. Pradera, T. Calvo. *Aggregation Functions, a Guide to Practitioners* (Springer-Verlag, Berlin, 2007).
- [6] T. Calvo, A. Kolesarova, M. Komornikova, R. Mesiar.

Aggregation operators, properties, classes and construction methods. In T. Calvo *et al.* (Eds.): *Aggregation Operators New Trends and Applications* (Physica-Verlag, Heidelberg), 3-104, 2002.

[7] V. Cutello, J. Montero. Hierarchical aggregation of OWA operators: basic measures and related computational problems. *Uncertainty, Fuzziness and Knowledge-Based Systems* 3, 17-26, 1995.

[8] V. Cutello, J. Montero. Recursive families of OWA operators. *Proceedings FUZZ-IEEE Conference* (IEEE Press, Piscataway, 1994), 1137-1141.

[9] V. Cutello, J. Montero. Recursive connective rules. *International Journal of Intelligent Systems* 14, 3-20, 1999.

[10] D. Dubois, S. Gottwald, P. Hajek, J. Kacprzyk, H. Prade. Terminological difficulties in fuzzy set theory - the case of intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 156, 485-491, 2005.

[11] J. Fodor, M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support* (Kluwer, Dordrecht, 1994).

[12] L.W. Fung, K.S. Fu. An axiomatic approach to rational decision making in a fuzzy environment. In: L.A. Zadeh *et al.* (Eds.): *Fuzzy Sets and their Applications to Cognitive and Decision Processes* (Academic Press, New York, 1975), 227-256.

[13] M.R. Garey, D.S. Johnson. *Computer and Intractability, a Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1978).

[14] D. Gómez, J. Montero. A discussion of aggregation functions. *Kybernetika* 40, 107-120, 2004.

[15] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications ACM* 12, 89-100, 1969.

[16] A. Kolesárová. Sequential aggregation. In: M. González *et al.* (Eds.): *Proceedings of the Fifth International Summer School on Aggregation Operators, AGOP* (Universitat de les Illes Balears, Palma de Mallorca, 2009), 183-187.

[17] D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, Vol. 1 (Addison-Wesley, 1968).

[18] D. E. Knuth. Big Omicron and Big Omega and Big Theta. *SIGACT News, ACM* 8, 18-24, 1976.

[19] V. López, L. Garmendia, J. Montero, G. Resconi.

Specification and computing states in fuzzy algorithms. *Uncertainty, Fuzziness and Knowledge-Based Systems* 16, 301-336, 2008.

[20] V. López, J. Montero. Software engineering specification under fuzziness. *Multiple-Valued Logic and Soft Computing* 15, 209-228, 2009.

[21] K.T. Mak. Coherent continuous systems and the generalized functional equation of associativity. *Mathematics of Operations Research* 12, 597-625 (1987).

[22] J. Montero. A note on Fung-Fu's theorem. *Fuzzy Sets and Systems* 13, 259-269, 1985.

[23] J. Montero, D. Gómez, H. Bustince. On the relevance of some families of fuzzy sets. *Fuzzy Sets and Systems* 158, 2429-2442, 2007.

[24] J. Montero, V. López, D. Gómez. The role of fuzziness in decision making. *Studies in Fuzziness and Soft Computing* 215, 337-349, 2007.

[25] R.R. Yager. On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics* 18, 183-190, 1988.

