Reaching consensus when experts use different linguistic terms

Aïda Valls
Dept. d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
Carretera de Salou, s/n
E-43006 Tarragona, Spain
avalls@etse.urv.es

Vicenç Torra
Institut d'Investigació en Intel.ligència Artificial
IIIA-CSIC
Campus Universitat Autònoma Barcelona
E-08193 Bellaterra, Barcelona, Spain
vtorra@iiia.csic.es

Abstract

In this work, we study synthesis methods when the experts use different linguistic labels to describe the same objects. This difference can be either syntactic or semantic. The consensus method that has been used is classification. We introduce a method to select and adapt one expert's vocabulary to express the consensued values for each object.

Keywords: Linguistic labels, Consensus, Cluster analysis.

1. INTRODUCTION

The problem of reaching the agreement among a group of opinions has been studied in recent years. In this framework, the knowledge is usually expressed with functions of the form f: object*attribute $\rightarrow$ value [8]. In this case, the possible values that an object can take, for a given attribute, are restricted by the type of this attribute and by its corresponding domain. For example, the domain of an attribute can be a set of integers, an interval of reals, or a specified set of linguistic terms [4]. When different experts provide information about the same objects, their opinions must be consensued before this information can be used in a knowledge based system.

In this work we focus on the consensus of values that are linguistic (i.e. labels). We can define our problem as a set of objects, a set of attributes and an evaluation of each object for each attribute. These values can vary with the expert, so for every attribute a domain is set, which determines the possible labels that the expert can use. Our goal is to find a consensus label for each object and to define, with these labels, a new attribute domain.

Previous studies in this matter are [1,2,3]. These methods assume all the labels to be equally informative; however, this is not always true. We use the definition of the negation function in [9] to make explicit the semantics of the labels. In addition, some of these methods force all the experts to use the same labels; on the contrary, our approach allows the experts to use their own vocabulary, and the consensus solution is also in terms of this vocabulary.

In the next sections we explain our methodology. As each expert is, in fact, making a partition of the objects in as many classes as values he uses, we find a consensued partition by using a classifier (section 2). Then, these classes are ordered according with the values of their members (section 3). Finally, a label is attached to each class. We do not invent a new domain, we adapt the terms used by the expert whose valuations are more similar to the consensus (section 4). In section 5 there are some examples of the process. Conclusions and future work are shown in section 6.

2. CLASSIFICATION USING *SEDÀS*

In [11] we introduced classification as a method to aggregate data in Decision Making. As it has been said, many consensus methods do not consider the case in which experts use different labels to describe the same objects; however, this is a common situation in real problems. On the other hand, in [2] objects described with different term sets are synthesized. Nevertheless, each object is consensued independently, as if others did not exist. If we consider that an expert gives the same value to objects that are similar under his opinion then, when we compute the agreed value of an object, we must consider the other objects that have the same values, in order to keep the similarity relations among them. This is natural in the classification process.

In our methodology, the first step in the synthesis of an attribute is to build a data matrix formed by the objects as rows, and the description made by each expert in columns. Then, the classification process joins, at each step, the objects that are closer using a similarity measure. In the end, we obtain a tree that shows the relations of the objects considering all the experts. The partition obtained cutting this tree has the most similar objects in the same class. After that, a label describing each class has to be found.

In this study we use a classification tool called *Sedàs* [10]. It is a parameterized hierarchical classifier. Its parameters allow the user to choose from a list of similarity functions and a list of classification techniques. The system produces n-ary trees, because more than two objects can merge in a single step. This is good to accelerate the classification process but it becomes a drawback when a partition on a fixed number of classes is to be obtained. In fact, experts can differ in the number of labels they use, so we choose the cut that produces a number of classes nearest to the mean of different labels in each column. In section 4, we will see how to adapt the vocabulary of one expert to the number of classes obtained.

3. THE ORDERING STAGE

The first problem in the consensus process is how to order the classes obtained in the classification. If we want to build a new attribute that synthesizes others, it must keep the properties of the original ones, that is, it must have an order over the labels, and a negation function over them.

To order the classes we make a Principal Components Analysis [6]. This well-known statistical method takes a set of variables that describe objects, and finds combinations of these variables to produce indices that are uncorrelated, called principal components. In addition, indices are also ordered, and the first one displays the largest amount of information in the original data. When the original variables are highly correlated the first component can summarize the 95-99% of the variation in the original values, so the projection of the data into the first principal component axis shows the same relations that were in the original space. In our framework, there should be a high correlation between the attributes because they are descriptions of the same property and they come from expert people. That is why we have chosen this method, taking as variables the opinions of the experts. The projection of the classes into the first principal component gives us the order among them. An interesting observation is that an index is a vector whose components express the contribution of each variable to the formation of this axis. Then we can detect when an expert differs from social opinions, because his contribution is smaller than that of the others.
Moreover, we can calculate a measure of the goodness of the projection into the principal component, in terms of the proximity of the class to the axis. Then, the closer to 1 is this value, the nearer is the class and the better is its projection. This allows us to discover objects that can not be synthesized because the experts do not agree in their descriptions, that is, the underlying semantics of their terms are in contradiction. In this situation, as the consensus is not possible, this object or group of objects[1] are getting out from the study, giving them the "unknown" label. This "unknown" label, in case it exists, is taken from the set of terms that experts use; on the contrary, a predefined linguistic label is used.

4. A MAPPING WITH AN EXPERT

The process of giving an appropriate linguistic label to each class is not easy. Generating arbitrary names for the labels we reduce the expressiveness of the values, because their meaning is not natural for people. Therefore, it is more suitable to adapt the domain of the experts. As, each attribute domain generates a partition on the set of objects according to the values that the objects take for the attribute, we find out which of these partitions is closer to the one obtained in the classification. To measure this similarity between pairs of partitions we use the distance defined in [5], which works with the average information of the partitions, that measures the randomness of the distribution of objects over the classes of the partition.

$$d(P_A, P_B) = \frac{I(P_B / P_A) + I(P_A / P_B)}{I(P_A \cap P_B)} \qquad (1)$$

where $I(P_A/P_B)$ is the conditional information of partition B given partition A, and $I(P_A \cap P_B)$ is the mutual average information of the intersection of the two partitions.

If the number of labels in the closer expert domain is the same than the number of classes in the consensus partition we map each label with its corresponding class, after ordering them. In the other case, when there are too few or too many labels, we have to adapt the expert's vocabulary by adding or deleting some values. By now, we have studied how to reduce a set of labels.

4.1 ELIMINATING LINGUISTIC LABELS

We proceed by eliminating a single label at each step, and repeat the process until the number of labels in the expert domain is equal to the number of classes in the consensus partition.
Let P_c be the consensus partition, and P_i be the partition of the expert after eliminating the i-th label, L_i. We choose for suppression the label, L_i, that minimizes $d(P_i,P_c)$, that is, the one that produces the most similar partition to the consensus one.

[1] Usually these groups of objects are small.

We have to take into account that when P_i is built, the objects originally valued with label L_i have to be valued with another label. Taking into consideration the order among the labels, the most suitable candidates are the previous and next labels, that is, L_{i-1} and L_{i+1}. Let us denote one of these neighbour labels as L_j. As we know the classes in the consensus that the objects in L_i belong to, we count the number of objects of L_i and L_j that are in the same class in P_c. This number is divided by the total number of objects in L_i and L_j, to obtain a ratio. Then we choose the L_j that maximizes this proportion.

5. THE NEGATION FUNCTION ON THE CONSENSUS DOMAIN

To build a new consensus attribute with the same properties as the original ones, the negation of the labels must be calculated.

Firstly, a numerical interval in [0,1] is attached to each label, $I(L_i)$. Then the negation of each label can be computed as:

$$Neg(L_i) = \{ L_j \mid I(L_j) \cap 1\text{-}I(L_i) \neq \emptyset \} \qquad (2)$$

where $1\text{-}I(L_i)$ is the interval between $1\text{-}\max(I(L_i))$ and $1\text{-}\min(I(L_i))$.

Disjoint intervals are built with the projections of the classes into the first principal component. We suppose that the labels have a triangular membership function [13], so the projection is taken as the point of the label with membership equal to 1. The middle point between two consecutive projections is the one that has membership equal to 0.5. We take these points as the limits of intervals.

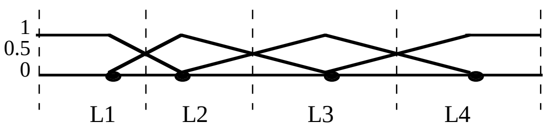

Figure 1: Fuzzy partition used to establish the semantics

6. SOME EXAMPLES

We have chosen the well-known Iris domain [7] to perform the initial studies. It is a set of 120 flowers described by means of 4 numerical attributes: petal and sepal length, and petal and sepal width. We will present two examples which are made considering only 90 flowers (to reduce the execution time) and synthesizing the attribute Petal Length, so the others are removed from the matrix. In order to generate different experts, we modify this attribute by adding to its values some levels of error. This error is generated with a normal distribution $N(0,s)$, with s being a proportion of the real variation fixed by the user. After that, the new values are discretized in a fixed number of labels [12], to generate a qualitative attribute.

Example 1: We build 4 matrices corresponding to experts E1, E2, E3 and E4. The error introduced in each one is 0%, 5%, 10% and 15% respectively. Each expert has 7 labels.

After classifying the objects, we can cut the tree in 7 classes. The distances of this partition to each expert are shown in table 1. The most similar expert is the second one. Therefore, later we will adapt his vocabulary.

Table 1: Distances between the consensus and the experts

E1	E2	E3	E4
0.47	0.37	0.47	0.51

Then, we compute the prototype of each consensus class, denoted C_i, for each expert, and a Principal Components Analysis is carried out.

Table 2: Results of the PCA in the first example

	E1	E2	E3	E4	Project	Good
C1	0.13	0.19	0.09	0.07	-0.24	0.87
C2	0.05	0.04	0.05	0.07	-0.10	0.96
C3	0.90	0.92	0.92	0.92	-1.83	0.99
C4	0.54	0.45	0.54	0.47	-1.0	0.99
C5	0.35	0.42	0.35	0.35	-0.73	0.99
C6	0.66	0.68	0.66	0.67	-1.34	0.99
C7	0.77	0.81	0.77	0.92	-1.63	0.99

Looking at the results shown in table 2, we can see the values of the prototypes of the classes in the first four columns, whereas the fifth is the projection value to the first component, and the last column shows the goodness of the projection (remember, the closer to 1, the better). In this case, the results are really good because the first axis has the 99% of the total variation. Furthermore, the contributions of the experts to the axis are: E1= -0.51, E2= -0.499, E3= -0.501 and E4= -0.499, which are almost identical, so they play the same role in the consensus.

The final order of the classes, according to the projections in the axis, is: C2 < C1 < C5 < C4 < C6 < C7 < C3.

The final step consists in giving to these classes their corresponding label. Let us suppose that terms used by E2 are the ones in the first row of table 3, then the class that takes each label is the one under it.

Table 3: Correspondence between the classes and the terms

tiny	little	small	medium	long	large	huge
C2	C1	C5	C4	C6	C7	C3

In this case we have exactly the number of terms we need to describe the consensus. The next example shows the case when there are too many labels in expert's domain.

Example 2: Let us take 4 matrices corresponding to experts E5, E6, E7 and E8. All of them have a 10% of error in data values, and are discretized in 7 labels.

The tree obtained in the classification cannot be cut in 7 labels. The nearest number of disjoint classes is 5. The distance measure between partitions shows that E8 is the most similar to the consensus achieved. From the Principal Component Analysis we obtain the order among the five classes, which is: C5 < C4 < C2 < C1 < C3. After this, we have to find the two labels to eliminate from the vocabulary of E8: {huge, large, long, medium, small, little, tiny}.
Then we study which label we should delete. The ratio for each pair of labels, as defined in section 4.1, is:

tiny-little:	1	medium-long:	0
little-small:	1	long-large:	12/17=0.7
small-medium:	0	large-huge:	17/20=0.85

Choosing for each label, L_i, the neighbour, L_j, that maximizes this ratio, we obtain the pairs shown in table 4. After studying all the possibilities, the best one is merging *tiny* and *little* into a unique word, and we keep the most used, which is *tiny*. Repeating the process we merge *huge* and *large*, keeping the label *large*.

Table 4: Pairs of labels we consider to merge

L_i	tiny	little	small	medium	long	large	huge
L_j	little	tiny	little	small	large	huge	large

Let's see the number of flowers in the intersections between the domain of the most similar expert, E8, and the consensus partition. We can see that the labels merged really correspond to the same consensus class.

Table 5: Relation between the terms in E8 and the consensus (before eliminating terms)

huge	0	0	0	0	7
large	0	0	0	3	10
long	0	0	0	21	0
medium	0	0	14	0	0
small	0	5	0	0	0
little	10	0	0	0	0
tiny	20	0	0	0	0
	C5	C4	C2	C1	C3

7. CONCLUSIONS AND FUTURE

We have introduced a method to adapt an expert's linguistic vocabulary to express the consensued values for a set of objects. In particular, we have studied the ordering and mapping stages.
The tests on the ordering process show that if the valuations of the experts are correlated the Principal Components Analysis captures high percentages of the information in the classes (about 98%). It is also observed that then the projection of the classes is good, and the order found agrees with the majority of the experts. Another important property is that an expert that disagrees with the group does not modify significantly the final consensus. Finally, the aforementioned capacity of discovering contradictions is also interesting to detect conflicting situations.
In the mapping process, the elimination method usually substitutes terms that are not discriminant in the consensus classes, which is an interesting behaviour.
Nevertheless, there are situations in which the number of classes in the consensus partition is greater than those used by the experts. To solve this problem, we are working in a method to generate new label names by adding qualifiers to the terms of the selected vocabulary.
The next step will be to make experiments on real data.

Acknowledgements

We acknowledge the support of the CYCIT project SMASH: TIC-96-1038-C04-04.

References

[1] L. Godo, V. Torra (1998) On qualitative weighted means, *Proceedings of the VIII Congreso Español soble Tecnologías y Lógica Fuzzy (ESTYLF)*, pp.185-192.

[2] F. Herrera, E. Herrera-Viedma, L. Martínez (1999) A fusion approach for managing multi-granularity linguistic terms sets in decision making, *Fuzzy Sets and Systems*, in press.

[3] F. Herrera, E. Herrera-Viedma, J.L. Verdegay (1997) A rational consensus model in group decision making using linguistic assessments, *Fuzzy Sets and Systems*, 88, 31-49.

[4] A.K. Jain, R.C. Dubes, (1988) *Algorithms for clustering data*, Prentice-Hall, Englewood Cliffs, NJ.

[5] R. López de Mántaras (1991) A distance-based attribute selection measure for decision tree induction, *Machine Learning*, 6, pp.81-92

[6] B.F.J. Manly (1994*) Multivariate Statistical Methods. A primer.* Ed. Chapman&Hall.

[7] P. Murphy, M. Aha, D. W., (1994) UCI Repository machine learning databases, University of California, http://www.ics.uci.edu/~mlearn/MLRepository.html

[8] P.H.A. Sneath, R.R. Sokal (1973) *Numerical taxonomy. The principles and practice of numerical classification,* W.H. Freeman and company.

[9] V. Torra (1996) Negation functions based semantics for ordered linguistic labels,*I.J.Intelligent Systems*,11,975-988.

[10] A. Valls, D. Riaño, V. Torra, (1997) Sedàs: A Semantic Based General Classifier System, *Mathware and Soft Computing*, 4-3, pp.267-279.

[11] A. Valls, V. Torra (1999) Using classification as an aggregation tool in MCDM, *Fuzzy Sets and Systems,* in press

[12] A. Valls, V. Torra (1999) On the semantics of qualitative attributes in knowledge elicitation, *Int. J. Intelligent Systems,* 14-2, pp.195-210.

[13] Y. Yuan, M.J. Shaw, (1995) Induction of fuzzy decision trees, *Fuzzy Sets and Systems*, 69, pp.125-139.