CLASSIFYING QUALITATIVE INFORMATION USING CENTROID BASED METHODS

Vicenç Torra, Lluís Godo

Institut d'Investigació en Intel·ligència Artificial - CSIC

Campus UAB s/n, 08193 Bellaterra (Catalunya, Spain)

{vtorra, godo}@iiia.csic.es

Abstract

In this paper we consider the problem of classification of qualitative data. We present an approach to overcome several difficulties which show SAHN (sequential, agglomerative, hierarchic, nonoverlapping) clustering methods to classify qualitative data when using centroids to compute similarities between pairs of classes. The approach is based on a recently proposed class of qualitative weighted average functions.

Keywords: qualitative aggregation, centroid, classification.

1 INTRODUCTION

Classification is one of the main tasks which is present in many AI systems ranging from knowledge acquisition tools to knowledge based systems verification. A classification process usually consists [Everitt, 1977] of a three step process: (1) domain description, (2) similarity construction and (3) classification construction. There are a lot of available classification methods, even if the data is imprecisely known and modelled as fuzzy sets, e.g. [Bezdek, 1981].

However, almost all the methods only deal with numerical data [Gordon, 1996], and if the available data to be classified is only of qualitative nature, a numerical or fuzzy sets interpretation of the data is required.

Among the algorithms to build classification trees we can distinguish the set of methods known as SAHN [Everitt, 1977; Valls et al., 1997] i.e., sequential, agglomerative, hierarchic, and nonoverlapping clustering methods. Given a set of objects $O = \{o_1, ..., o_m\}$ to classify in a tree of classes C, these methods follow the structure given below:

1. For each object o_i, define a class that consists only of that object
2. Computation of the similarities among all pairs of classes
3. While there exists more than a single class in C
   - 3.1. Selection of the classes K that should integrate the new class
   - 3.2. Creation of a class C_K with the classes in K
   - 3.3. Removal of the classes in K
   - 3.4. Calculation of the similarity between C_K and the ones in C
4. End while

This can be expressed more conscisely as follows:

1. For all $o_i \in O$ loop $C = \cup \{o_i\}$ end loop
2. S = Similarity (C)
3. while |C| >1
   - 3.1. C_K = Selection (S, C), $C_K \subseteq O$
   - 3.2. $C = C \cup \{C_K\}$
   - 3.3. $C = C - \{c \in C \mid c \subset C_K, c \neq C_K\}$
   - 3.4. S = NewSimilarity(S, C, C_K)
4. end while

In this general approach, two aspects are left to the user: the selection of the aggregation criterion (how to select the objects that form the new class), and the classification method (how to calculate new similarities from old ones). Several alternatives exist for both criteria. Among them, we can distinguish, in relation to the former, the selection of all those elements that are related with a large similarity. In relation to the latter criteria, one of the existing methods is the Centroid cluster analysis. In this case, to compute the similarity between two classes, first it is computed a representative of the class (the so-called *centroid*), and then a similarity function is applied to the centroids. See [Aguilar, 1995, and Aguilar et al., 1990] for an interesting and similar approach to classification of qualitative data based on adequation degrees of objects to clusters.

As a matter of notation, in this paper we consider that the set of relevant attributes to be used in the classification is $A = \{A_1, ..., A_n\}$, with domains $Dom(A_i)$ consisting of a finite set of linearly ordered (linguistic) labels. We use $Val(o_i, A_j) \in Dom(A_j)$ to denote the evaluation of the object o_i in relation to the attribute A_j.

In general, using the centroid method to compute the similarity between two classes C_R and C_P built from objects $R = \{r_1, ..., r_r\}$ and $P = \{p_1, ..., p_p\}$ we proceed as follows:

1. Computation of the centroids of R and P

$$\text{Centroid}(R) = (\mathbb{C}_{A1}(\text{Val}(r_i, A_1)_{i \in \{1, ..., r\}}), ..., \mathbb{C}_{An}(\text{Val}(r_i, A_n)_{i \in \{1, ..., r\}}))$$

$$\text{Centroid}(P) = (\mathbb{C}_{A1}(\text{Val}(p_i, A_1)_{i \in \{1, ..., p\}}), ..., \mathbb{C}_{An}(\text{Val}(p_i, A_n)_{i \in \{1, ..., p\}}))$$

where $\mathbb{C}_{Ai}$ is an aggregation operation (usually the arithmetic mean) defined on the domain of the attribute A_i. Note that this aggregation operator has to be defined in such a way that can combine an arbitrary number of linguistic labels. This is due to the fact that not all classes have always the same number of objects.

To simplify the notation in the rest of the work, we will use $\text{Val}(P, A_k)$ and $\text{Val}(R, A_k)$ to denote:

$$\text{Val}(P, A_k) = \mathbb{C}_{Ak}(\text{Val}(p_i, A_k)_{i \in \{1, ..., p\}})$$
$$\text{Val}(R, A_k) = \mathbb{C}_{Ak}(\text{Val}(r_i, A_k)_{i \in \{1, ..., r\}})$$

2. Computation of the multi-dimensional similarity between the two centroids

$$\text{Multi-dimensional-similarity}(R, P) = (S_{A1}(\text{Val}(R, A_1), \text{Val}(P, A_1)), ..., S_{An}(\text{Val}(R, A_n), \text{Val}(P, A_n)))$$

3. Reduction of the multi-dimensional similarity to a one-dimension similarity.

$$S(R, P) = \text{Agg}(\text{Multi-dim-similarity}(R, P))$$

where Agg is an aggregation operator.

In the numerical case, the computation of all these values is straightforward. In that case, $\mathbb{C}_{Ai}$ can be any numerical aggregation operator (e.g. the arithmetic mean), S_{Ai} is any similarity function (e.g. a dual of a normalized Euclidean distance) and the aggregation procedure in step 3 can be again any numerical aggregation operator, since all the values $S_{Ai}(\text{Val}(R, A_i), \text{Val}(S, A_i))$ to aggregate are numerical.

2 QUALITATIVE CLASSIFICATION

In the case attributes take values in an ordinal scale, without any underlying numerical or fuzzy set based representation, difficulties arise at the three stages considered above. Let us consider them in more detail:

1. *Computation of the centroid*. In the qualitative setting, we would need aggregation procedures $\mathbb{C}_{Ai}$ able to combine qualitative terms instead of numerical ones. Besides of that, procedures $\mathbb{C}_{Ai}$ depend on the domain of the attribute A_i. Therefore we would need to define as much operators as attributes we have.

2. *Computation of the multi-dimensional similarity*. As in the previous case, it is required a similarity function for each existing domain.

3. *Reduction to a one-dimension similarity*. The way this reduction is performed depends on how functions S_{Ai} are defined. Note that, in general, given two linguistic terms *a* and *b* in the domain of A_i, the function S_{Ai} will compute its similarity in a certain domain D. However, it is a common assumption that the domain D coincides with the domain of A_i. In this case, the reduction function has to combine n values in n different domains.

Below we briefly describe our approach to face each of these problems.

2.1 CENTROID COMPUTATION

To define the centroid of a class we use, for each attribute, a qualitative aggregation operator based on the qualitative weighted mean defined in [Godo and Torra, 1998, 1999].

This operator performs a weighted aggregation of values u_i belonging to a certain ordinal scale $U=\{0=u_0 < ... < 1=u_n\}$ and weighted by natural numbers. We will denote by $\mathbb{N}$ the set of natural numbers with the usual addition (+) and substraction (-). To overcome the difficulty of having two different domains, the approach consists of building a new unified domain $U_{\mathbb{N}} = \mathbb{N} \times (U-\{1\})$ where suitable addition, product and division like operators needed for the aggregation are defined.

As for the addition operator in $U_{\mathbb{N}}$, we need, first of all, an operation in U accounting for the notion of accumulation. It is well known on fuzzy set theory that t-conorms are the binary operations which are closest to this type of operations, enjoying properties like commutativity and associativity, specially needed in the qualitative setting.

From now on $\oplus: U \times U \to U$ will denote a finite t-conorm on the scale $(U, <)$, that is, a non-decreasing commutative and associative operation fulfilling these boundary conditions: for all $u \in U$, $u \oplus 0 = u$ and $u \oplus 1 = 1$. Moreover, if we denote by n_U the order-reversing involution on U, one can define the t-norm $\otimes: U \times U \to U$ which is the dual operation of $\oplus$ with respect to n_U as:

$$u \otimes v = n_U(n_U(u) \oplus n_U(v)).$$

Once operations $\oplus$ and $\otimes$ are defined, we proceed to embed the algebraic structure $(U, <, \oplus, \otimes)$ into $U_{\mathbb{N}}$. The transformation $T: U \to U_{\mathbb{N}}$ that maps values of U into values of $U_{\mathbb{N}}$ is as follows:

$$T(u) = \begin{cases} (0,u), & \text{if } u<1 \\ (1,0), & \text{if } u=1 \end{cases}$$

The orderings in U and in $\mathbb{N}$ induce the following lexicographic ordering in $U_{\mathbb{N}}$:

$$(n, u) \leq (m, v) \text{ if either}$$

$$n < m \text{ or } (n = m \text{ and } u \le v).$$

Next, a corresponding binary addition-like operation in the extended domain

$$\underline{\oplus}: U_{\mathbb{N}} \times U_{\mathbb{N}} \to U_{\mathbb{N}},$$

extending the $\oplus$ operation in U, is defined in such a way that keeps track, by means of the t-norm, of how much the accumulation exceeds from 1.

Definition 1. $(n,x) \underline{\oplus} (m,y) =$

$$\begin{cases} (n + m,\ x \oplus y), & \text{if } x \oplus y < 1 \\ (n + m + 1,\ x \otimes y), & \text{if } x \oplus y = 1 \end{cases}$$

In [Godo and Torra, 1999] necessary conditions for the t-conorm $\oplus$ are given so that the accumulation operation $\underline{\oplus}$ be associative, and thus, it is enough to have it defined as a binary operation.

The weighting of elements of U by natural numbers is done by means of a product-like operation

$$\bullet : \mathbb{N} \times U_{\mathbb{N}} \to U_{\mathbb{N}},$$

defined as an iterative procedure of the addition operation $\underline{\oplus}$ in the natural way.

Definition 2. For each $n, m \in \mathbb{N}$ and $x \in U-\{1\}$, we define:

$$n\bullet(m, x) = (m, x) \underline{\oplus} \overset{n)}{..}.. \underline{\oplus} (m, x).$$

Notice that $(n_1+n_2)\bullet(m, x) = n_1\bullet(m, x) \underline{\oplus} n_2\bullet(m, x)$ and that $n\bullet[(m_1, x) \underline{\oplus} (m_2, y)] = n\bullet(m_1, x) \underline{\oplus} n\bullet(m_2, y)$.

Finally, to complete the modelling of the averaging of the values $(u_1, ..., u_m)$ with weights $(n_1, ..., n_m)$, a division-like operation $\varnothing$ between the extended value resulting from the addition $[n_1\bullet T(u_1)] \underline{\oplus} \ldots \underline{\oplus} [n_m\bullet T(u_m)]$ and the sum of weights $n_1 + ... + n_m$ is defined.

Definition 3. For each $n, m \in \mathbb{N}$ and $x \in U-\{1\}$, we define $(n, x) \varnothing m = ([n/m], y)$ being $[n / m] = \max\{p \in \mathbb{N} \mid p\cdot m \le n\}$ (integer division) and $y = \max\{z \in U-\{1\} \mid m\bullet(0, z) \le (r, x)\}$, where $r = n - [n / m]\cdot m$.

Finally, since the aggregation should be in U, we use T^{-1} to yield the final result.

Definition 4. Given a t-conorm $\oplus$ in a finite scale U inducing operations ($\underline{\oplus}$, $\bullet$, $\varnothing$) as defined above, the **qualitative weighted mean** of a vector $\mathbf{u} = (u_1, ..., u_m)$ of values of U with respect to a vector $\alpha = (n_1, ..., n_m)$ of weights of $\mathbb{N}$ is defined as

$$QWM_{U,\oplus}(\alpha,\mathbf{u}) = T^{-1}(([n_1\bullet T(u_1)] \underline{\oplus} \ldots \underline{\oplus} [n_m\bullet T(u_m)]) \varnothing \textstyle\sum_{i=1,m} n_i)$$

In our case, to compute the centroid we assume that $n_i=1$ for all i. This means that all objects in the class are equally represented in the centroid.

This definition depends on the domain U and also on the t-conorm $\oplus$. Therefore to use it to compute the centroid we need to settle both of them. Thus we have:

$$\mathbb{C}_{Ai}(\mathbf{u}) = QWM_{DOM(Ai),\oplus}(\mathbf{1}, \mathbf{u})$$

where $\mathbf{1} = (1, ..., 1)$.

2.2 COMPUTATION OF THE MULTI-DIMENSIONAL SIMILARITY

In this case, we have defined a similarity function for each attribute as a many-valued equivalence connective. We have based the similarity on the same t-conorm used to define the aggregation operator to compute the centroid.

$$S_{Ai}(x, y) = \min(\max\{z: (x \otimes z) \le y\}, \max\{z: (y \otimes z) \le x\})$$

In this function, $\otimes$ is the dual t-norm of $\oplus$. Notice that $S_{Ai}(x, x)$ takes the maximum value of the domain. Thus, the function depends on both the domain of the attribute and the t-conorm $\oplus$. Therefore, in general we have a function for each attribute. Note also that $S_{Ai}(x,y)$ is in the domain of A_i.

2.3 REDUCTION TO A ONE-DIMENSIONAL SIMILARITY

To reduce to a one-dimension similarity, we need to introduce a new set of labels D_c as a common domain for all attributes. To this end we take D_c large enough to allow each domain $Dom(A_i)$ to be embedded into D_c by means of an onto order-preserving mapping f_{Ai}: $Dom(A_i) \to I(D_c)$, sending each domain value to an interval of values of D_c, and such that $\cup\{f_{Ai}(a) \mid a \in Dom(A_i)\} = D_c$. Here $I(D_c)$ denotes the set of intervals of D_c.

Let us assume that the multi-dimensional-similarity between two centroids R and P is given by the vector $(a_1, ..., a_n)$, where

$$a_i = S_{Ai}(Val(R, A_i), Val(P, A_i)) \in Dom(A_i)$$

Then, the one-dimensional similarity between R and P is defined as

$$S(R, P) = Agg(a_1, ..., a_n) = Agg'(f_{A1}(a_1), ..., f_{An}(a_n))$$

A particular definition of the Agg' function could be just to let

$$Agg'(f_{A1}(a_1), ..., f_{An}(a_n)) = \text{Min}\{S'(f_{Ai}(a_i), f_{Aj}(a_j)) \mid i \ne j\}$$

where $S'([I^i, I^s], [J^s, J^s]) = \min(S^*(I^i, J^i), S^*(I^s, J^s))$, being S^* a similarity defined (in the same way as in Section 2.2) on D_c.

3 EXAMPLE

We have applied the method developed to a small example. The example consists of only 10 objects (cars), denoted by {c1, c2, .., c10} and extracted from a large file of 1728 objects [Murphy and Aha, 1994], that are evaluated using 6 qualitative attributes: buying price, price of maintenace, number of doors, capacity in terms of persons to carry, size of luggage boot, and estimated safety of the car. The domain of these attributes is given in Table I and the evaluation of the 10 objects is given in Table II.

Attribute	Domain
buying	low med high vhigh
maint	low med high vhigh
doors	two three four 5more
persons	two four more
lug_boot	small med big
safety	low med high

Table I. Attribute domains

	buying	maint	doors	pers.	lug-b.	safety
c1	vhigh	vhigh	two	two	small	low
c2	vhigh	vhigh	three	two	small	low
c3	vhigh	vhigh	three	two	small	med
c4	vhigh	vhigh	four	two	small	med
c5	vhigh	vhigh	5more	two	small	low
c6	high	low	5more	more	small	low
c7	high	low	5more	more	small	med
c8	high	low	four	more	small	med
c9	high	med	four	four	small	high
c10	high	med	three	four	small	high

Table II. Objects to classify

We have defined the aggregation operator in each domain based on the corresponding Lukasiewicz t-conorm ($a_i \oplus a_j = a_{min(i+j,n)}$, where n is the cardinality of the domain).

The common domain D_c where all the attribute domains in Table I have been mapped has been taken as a 12 element scale, in such a way that each element of buying, maint and doors domains are mapped to a three element intervals, while each element of the remaining domains are mapped to a 4 element interval.

The classification we have obtained for these objects is given in Figure 1.

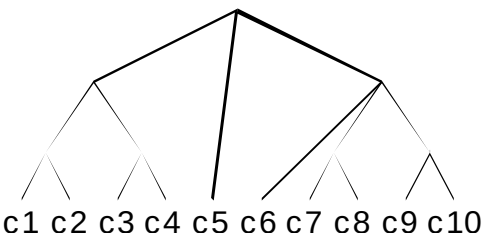

Figure 1. Classification of data in Table I

4 CONCLUSIONS AND FUTURE WORK

We have described how to apply a class of qualitative aggregation procedures in a classifier system based on centroids. The methodology developed permits to operate all the elements while remaining in the qualitative setting without translating them into quantitative one.

As a future work it is needed a deeper study of the process of reduction from a multi-dimensional similarity to a one dimensional similarity. With the approach used here we need to define a unified framework and a transformation of each value in each domain into this unified framework. This assumption needs, however, too much information to be supplied by domain experts. An alternative approach with no so much knowledge would be better for real applications.

Acknowledgements

The authors acknowledge partial support from the CICYT project SMASH, TIC96-1138-C04-01/04.

References

Aguilar-Martin, J., (1995), *Representation inductive des connaissances a partir des donnees*, Report de Recherche, LAAS.

Aguilar-Martin, J., Martín, M., Piera, N., (1990), Conceptual Connectivity Analysis by Means of Fuzzy Partitions, *Proc. of the IPMU 90*, pp. 250-252, Paris, France.

Bezdek J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, NY.

Cignoli R., D'Ottaviano I., Mundici D. (1995), Algebras das logicas de Lukasiewicz, *Coleçao Centro de Logica, Epistemologia e Historia da Ciencia*, Unicam (Brasil) Volumen 12 (1995).

Everitt B., (1977) *Cluster Analysis*, Heinemann Educational Books Ltd.

Godo L., Torra, V., (1998), On qualitative Weighted Means, *Actas del VIII Congreso español de tecnologias y lógica fuzzy*, (ISBN 84-95075-10-5), 185-192, Pamplona.

Godo L., Torra V., (1999) On aggregation operators for ordinal qualitative information, submitted.

Gordon A. D., (1996) Hierarchical Classification. In *Clustering and Classification*, P. Arabie, L.J. Hibert and G. De Soete (eds.), World Scientific Publishing, River Edge NJ, pp. 65-121.

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

Valls, A., Riaño, D., Torra, V., (1997), Sedàs: A semantic based general classifier system, *Mathware and Soft Computing*, 4, 267-279.