Mathware & Soft Computing 4 (1997) 267-279

Sedàs: A Semantic Based General Classifier System

A. Valls D. Riaño V. Torra
Departament d'Enginyeria Informàtica (ETSE)
Universitat Rovira i Virgili
Carretera de Salou, s/n, E-43006 Tarragona (Catalonia, Spain)
e-mail: {*avalls, drianyo, vtorra*}*@etse.urv.es*

Abstract

In this work we present the general classifier system Sedàs. We show how this system implements the description of the domain and how it builds similarity matrices and classification trees. The system uses a new semantics, introduced in [21], to define a distance between qualitative values.

Keywords: Clustering, Knowledge Representation, Knowledge-based Systems.

1 Introduction

In the knowledge acquisition process, the expert, together with the Knowledge engineer, is faced up to the development of a model of his own expertise. The modelling process is, however, a difficult task because the definition of a model oblies the expert to describe an experience that had not been previously expressed in an explicit form. For this reason, nowadays there are several efforts devoted to the definition and construction of tools for automatic knowledge acquisition. These tools that use Artificial Intelligence techniques pretend to help the experts to make their knowledge explicit and to represent it.

Among knowledge acquisition systems we find those based on G. Kelly's Personal Construct Theory [13]. In these systems, the expert defines data matrices that represent functions of the form f: Object*Attribute $\rightarrow$ Value. The definition of these matrices is done [2] in an interactive way. At any time, the system can analize the similarity between attributes and between objects already introduced by the expert so that the expert can reconsider decisions already made or develop non complete aspects. These systems also allow the user to generate a representation of the knowledge by means of rules, fuzzy sets or frame systems relating the concepts and objects already defined. Some of the systems developed under this ideas are ETS [2], extended in AQUINAS [3] and KSSn [7].

However, there are still some open questions in knowledge acquisition tools development. Some of them are the following:

1. Value types other than quantitative ones are needed for object-attribute pairs. In the systems mentioned above -ETS, AQUINAS, KSSn - the mapping f(object, attribute) takes values in the ordinal reduced scale (1 to 5). As not all the knowledge can be expressed in this way there have been several attempts to use other kinds of mappings (into a set of labels or into the boolean set). See for example KAFES [11], LINNEO+ [15], EAR* [18].

2. Dealing with information from several experts is needed. It is not an isolate case that a model should be defined from the knowledge of several experts [14]. In this case a synthesis process of the information provided by each of them is needed. The EGAC [23] tool and also ETS and AQUINAS systems deal with synthesis of information. EGAC is a tool for synthesis of information in knowledge acquisition systems based on personal construct theory. It builds from a set of data matrices defined by a set of experts a new matrix that synthesises the information of original matrices. This new matrix is built translating the synthesis problem into a classification one.

3. Other kind of knowledge representation than mappings for object-attribute pairs is needed. Due to the fact that some knowledge cannot be easily represented by this kind of mappings, sometimes it is required a system that includes the use of rules, predicates, relations between attributes (e.g., causality relations).

4. Friendly interfaces are required. This is, nowadays, one of the most wished characteristic. Since Knowledge Based Systems (KBS) have an increasing complexity, the expert has to deal with more information. Friendly interfaces help the user when manipulating data.

In this work we present a general classification process to be used in knowledge acquisition tools (e.g., EGAC and GAR [19]). This latter tool is to extract rules from classifications. Due to the fact that a classification process is used in Artificial Intelligence not only in the knowledge acquisition setting but also in other frameworks as knowledge based system verification [9], and also due to the fact that there is not a single classification method but, several classification ones, we have built a general classification architecture. The system includes several classification methods and is open to the definition of new ones. The system has been designed to be used as both a stand-alone system or embedded in another tool. The system, Sedàs, is centered to points 1 and 4 described above. On one hand it deals not only with attributes with a quantitative domain but also with qualitative and boolean domains. On the other hand, we have developed a friendly interface for the system.

In this work together with Sedàs, we present the definition of the distance used between values when their domain is a set of ordered qualitative labels. This distance is based on a method, introduced in [21], to extract a semantics for each label in a set from a negation function defined over the set.

The structure of the paper is as follows. First, section 2, we describe the General Classification System. Then, section 3, we introduce the definition of the distance for qualitative values. Section 4 is devoted to some examples. We finish with the conclusions and some future work.

2 The general classification system

Several authors (as [6]) consider the classification process as a three step process: (1) domain description, (2) similarity matrices construction and (3) classification construction. Our system distinguishes these 3 steps to allow the user to choose the appropriate function for each step. Moreover, from a single data matrix several classifications can be analysed changing some of the functions and keeping the others constant. These steps are briefly described in this section.

The system besides of giving the basic functions corresponding to these steps it also offers, in a higher level, a friendly interface. The interfase allows the user:

1. to define data matrices, both from files and from keyboard
2. to classify data (once the similarity function, the classification method and the aggregation criterion are defined - see below).
3. to visualize the stored data structures: data matrices, similarity matrices and classification trees
4. to save data on disk.

2.1 Domain description

In general, the domains where reasoning systems are applied are too wide and with a large amount of characteristics, thus it is usual to consider only a subset of the elements of the domain (data bias), which are enough to represent the whole domain, and only a subset of the properties (description bias).

Definition 1 *Let* $O = \{O_1, O_2, \ldots, O_m\}$ *be the set of objects that define the data bias, let* $A = \{A_1, A_2, \ldots, A_n\}$ *be the set of attributes that define the description bias and let* $DOM(A_k)$ *be the domain (the set of possible values) of the attribute* A_k*.*

In this way the information about the domain (the knowledge that the expert makes explicit) is described by means of an application f: $O*A \rightarrow \cup_{k \in \{1,..,n\}} DOM(A_k)$. In classifier systems the function is represented by means of a two dimensional matrix M. See definition 2.

Definition 2 *Let A be a set of attributes and let O be a set of objects, then a data matrix M is defined as the set of values V(k,j) for each object* $O_k \in O$ *and each attribute* $A_j \in A$*. So, V(k,j) stands for the value corresponding to the attribute j for the object k and it should hold that* $V(k,j) \in DOM(A_j)$*. Notice that the mapping* $f(O_k, A_j)$ *corresponds to V(k,j).*

Distance	Differences, MCD (Mean Character Difference), Manhattan or City Blocks, Camberra Metric, Taxonomic Distance, Minkowski.
Association	Jaccard, Dice, Simple Matching, Rogers & Tanimoto, Yule, Hamann
Correlation	Pearson

Table 1: Similarity functions

Our system considers, concerning the attributes, four types of domains: Boolean, Quantitative, Non ordered Qualitative and Ordered Qualitative. The first type of domain describes whether a certain characteristic is satisfied or not. In the second one, the domain is defined as a subset of real numbers. In non Ordered Qualitative attributes the domain is defined as a set of discrete values without any significative ordering, instead, a total ordering is required when Ordered Qualitative attributes are considered.

2.2 First module: Construction of similarity matrices

Once there is a description of the domain, the next step on the process of constructing a classification consists on the transformation of the data matrix (definition 2) into a similarity relation between objects. This process is achieved by means of functions (they can be found in [6, 20]) defined among pairs of objects that express how similar they are. The functions of our system can be classified into three groups according to [20]:

- Distance based coefficients
- Association coefficients
- Correlation coefficients

The functions implemented are listed in table 1 according to this classification. Their definition is given in [6, 20]. Distance coefficients measure the distance between objects in a numeric space defined in various ways (the most familiar measure of distance is simple Euclidean distance). Association coefficients are based on various algorithms involving qualitative data (two-state attributes or multistate attributes). Correlation coefficients measure proportionality and independence between pairs of object vectors.

When implementing similarity functions, one of the problems that face classifiers is the need of dealing with qualitative values in a numeric way, i.e., to calculate distances and similarities between pairs of linguistic labels. The way this problem was handled is detailed in section 3. We show how to define an interval for each linguistic label in a set L by means of a negation function.

2.3 Second module: The construction of the classification tree

Once the similarity relation is defined for each pair of objects in the data matrix, the system builds a classification tree. The classification tree obtained follows definition 3 below:

Definition 3 *[8] A classification tree over a set of objects O is defined as a set T of subsets of O that satisfy the following conditions:*

1. $O \in T$, $\emptyset \notin T$, $\{O_i\} \in T$ *for all* $O_i \in O$

2. $M \cap N \in \{\emptyset, M, N\}$ *for all* M, $N \in T$

In this way, the classification trees obtained by the system Sedàs are n-trees, i.e., a generalization of the more frequently used structures in AI -decision trees, dendograms. Usually classification trees are forced to be binary, so each node has only two children. The use of binary trees is justified in terms of the facility with which these structures are obtained and treated. However as binary trees are not as much close to the knowledge they represent as n-trees, we have decided to use the latter structure.

The classification process, besides of returning the set of nodes that form the classification tree, assigns to each node a cohesion value of the class it represents. This value corresponds to a measure of similarity of the last union (i.e., when all the subclasses have been gathered to form the class that the node represents).

The algorithms implemented in our system Sedàs to build classification trees belong to the set of methods known as SAHN [6], i.e., sequential, agglomerative, hierarchic, nonoverlapping clustering methods. The structure of the algorithms follows the general classification algorithm:

General classification algorithm

```
0.   Construction of the initial similarity matrix.
1.   Selection of the objects that should integrate
     the new class (aggregation criterion).
2.   Modification of the similarity matrix.
  2.1. Elimination of the objects in the new class.
  2.2. Insertion of the new class.
  2.3. Calculation of the similarity between the new
       class and the rest of objects (classification criterion).
3.   Repeat steps 1-2 until we have a single class.
```

The system allows the user to select the aggregation criterion (how to select the objects that form the new class) and the classification one (how to calculate new similarities from old ones) to be used by the classifier. We outline below some of the aspects related to these criteria.

1. *Aggregation criterion.* It is used to determine which of the elements form a new class. At present, there is only a single criterion implemented that consists on defining the new class with those objects that have a minimum dissimilarity value (so they are the most similar ones). This is the most used method.

   This criterion can be applied gathering at each step only two objects (in this way we obtain binary trees) or gathering in each step all those objects with a minimum dissimilarity (so we obtain n-trees). As it has been said, we have implemented this latter option.

2. *Classification criterion.* It is used to recalculate the similarity matrix when a new class is built. In the system Sedàs we have implemented the following methods: Single linkage, Complete linkage, Arithmetic average, Group average, Centroid cluster analysis, Median cluster analysis, Ward's method. The methods are described in [6, 24].

3 Distance definition for ordered linguistic labels

In section 2.2, when describing the classifier, we have pointed out that a distance function between pairs of linguistic labels is needed when implementing similarity functions. In this section we introduce the definition of a distance between pairs of linguistic labels based on negation functions. See [21] for more details. The section begins with classical negation functions over linguistic labels. Then, we comment some drawbacks of these functions and we introduce a new definition of negation functions that is used latter on to define the semantics. The semantics is used to define the distance.

Negation functions over a set of ordered linguistic labels L = $\{x_0,...,x_n\}$ (such that, $x_0 < ... < x_n$) are usually defined [12, 10] as functions from L to L that satisfy:

N1) if x < x' then Neg(x) > Neg(x') for all x, x' $\in$ L
N2) Neg(Neg(x)) = x for all x $\in$ L
The later condition, N2, can be equivalently rewritten as
N3) if x=Neg(x') then x'=Neg(x)

However, a negation function defined in this way is not always adequate. In fact, when N1 and N2 hold Neg(x) should be defined as Neg(x_i)=x_{n-i}, so, it stands for situations where each label of the pair $<x_i,x_{n-i}>$ is equally informative. It is possible, instead, that some subdomain of the reference set is more informative than the rest. In this case, the density of linguistic labels in this subdomain should be greater than the density in the rest of the domain. In fact, in our case, we are interested on representing situations similar to figure 1 and defining the similarity or the distance between labels accordingly.

Usually when a system considers ordered linguistic labels, it also considers their corresponding semantics defined by means of fuzzy sets (i.e., for each linguistic label there is a fuzzy set over a reference set attached to it). See, for instance [4]

Figure 1: Labels distribution in the domain

in fuzzy control, [27] in KBS and [10] in aggregation of opinions. In this case, the classifier can define the similarity between any pair of labels as a function of their corresponding fuzzy set (e.g., by means of the extension principle [5]). However, in the framework of knowledge acquisition (modelling) the expert is not always able to define a fuzzy set for each linguistic label because this would require an excess of accuracy that he/she can not always supply.

Instead, we want that the expert could express his/her knolwedge about the terms by means of a negation function. Nevertheless, when the set of linguistic labels is intended to be according figure 1, the negation function cannot be of the form $\mathrm{Neg}(x_i)=x_{n-i}$. Notice that in this figure, the negation of *Almost-Nil* (AN), and also the one of *Very-Low* (VL), should be *Very-High* (VH). However, this negation function does not satisfy condition N1. Besides of that, if we consider condition N3, then, it is not even possible to define Neg(*High*).

To overcome these problems, we have defined [21] a negation function as a function Neg from L to parts of L (Neg: L $\rightarrow \wp$(L)) weakening the conditions given above.

Definition 4 *A function Neg: $L \rightarrow \wp(L)$ is a negation function if and only if it satisfies:*

- *C0) Neg is a convex and a non empty function*
- *C1) if $x < x'$ then $Neg(x) \geq Neg(x')$ for all $x,x' \in L$*
  where $A \geq B$ is true, if and only if, $\min \{y \mid y \in A\} \geq \max \{y \mid y \in B\}$
- *C2) if $x \in Neg(x')$ then $x' \in Neg(x)$*

C0 is a technical condition, it requires that the negation of a label is not an empty set (so, every label has a negation) and that the negation of a label is a convex set (i.e., a subset X of L is defined as convex if and only if for all x,y,z $\in$ L such that x $<$ y $<$ z and x,z $\in$ X, then y $\in$ X)

Given a set of linguistic labels L = $\{x_0,...,_x n\}$ (such that, $x_0 < ... < x_n$) and a negation function Neg: L $\rightarrow \wp$(L), in [21] we have defined for each label x_i its corresponding semantics as the interval $I(x_i)$:

$$I(x_i) = [m_{in}, m_{ix}] = [\sum_{x<x_i} |Neg(x_i)| / \sum_{x \in X} |Neg(x_i)|, \sum_{x \leq x_i} |Neg(x_i)| / \sum_{x \in X} |Neg(x_i)|].$$

This semantics has been defined in a way that the negation over L performs similar than the negation over the unit interval. We have considered N(x)=1-x

Figure 2: Labels obtained according to (a) source information, (b) classical intervals and (c) the new semantics.

as the function in this interval. The enunciation of these properties are in [21] together with the proof that the semantics satisfies these properties.

A negation function following definition 4 makes possible to express the relation between labels in figure 1:

Neg(*almost-nil*) = Neg(*very-low*) = {*very-high*}
Neg(*quite-low*)= Neg(*low*) = {*high*}
Neg(*medium*) = {*medium*}
Neg(*high*) = {*quite-low*, *low*}
Neg(*very-high*) = {*almost-nil*, *very-low*}

To see that this semantics modifies the induced interval, we consider again the original intervals in figure 1. We display them in figure 2a together with the limits of the intervals (with its domain normalized in [0,1]). In figure 2b, we display the intervals when the relations among labels are not expressed with a negation function (i.e., classical intervals are considered). In figure 2c, we display the intervals inferred according to the new semantics when the negation function defined above is used. It can be seen that the new semantics induces a set of intervals that are more similar to the original ones than the ones in figure 2b. Notice that as the intervals are built upon the negation function, when the negation function is changed, they change also. Notice also that when $\mathrm{Neg}(\mathrm{x}_i)=\mathrm{x}_{n-i}$, the intervals inferred are the classical ones (i.e., in the example of figure 2a, we would infer 2b).

Once there is a semantics in [0,1] attached to each label, it is possible to define a distance between pairs of them. In our case as there is not more information about the interval, this is achieved calculating a numeric value $q'(l_i)$ corresponding to the center of the interval, i.e., $q'(l_i) = q(l_i)/\sum_{x\in L}|Neg(x)|$ where $q(l_i)$ is defined as:

$$q(l_i) = \frac{\sum_{j=0}^{i-1} |Neg(l_j)| + \sum_{j=0}^{i} |Neg(l_j)|}{2}$$

Recently, we have studied the validity of this semantics in the framework of knowledge acquisition systems. Our first results have been reported in [25]. We have observed that this semantics leads to a classification with more structure in relation to the case that the classical semantics is used. This is, when only one attribute is considered (an ordered qualitative one), the classification depends only on the mapping of each label into the unit interval. In this case, if classical semantics is used, the resulting classification can only be splitted into two α-cuts. One that corresponds to the equivalence relation built upon the linguistic labels. In the other α-cut there is only one class with all the objects. This is not always the case when the semantics used is the one inferred from a negation function of the form of definition 4.

4 Examples

The system has been applied to data on several domains (biology, data on car importation, ...) some of them obtained from a public repository [16]. Here, in order to give an example of the components mentioned above we show the results of a small example taken from [23]: Data matrices about programming languages with 13 objects and 8 attributes. The data matrices were fullfilled by some members of the academic staff at the Computer Science department at the Universitat Politècnica de Catalunya and of the Institut d'Investigació en Intel.ligència Artificial (IIIA, CSIC). The set of objects (O) and the set of attributes (A) are defined in these matrices in the following form:

O = {Lisp, C, Pascal, Scheme, FORTRAN, Prolog, ML, Modula-2, Basic, FP, Assembler, Ada, COBOL}

A = {clarity, compactness, power, comprehension, structures, modularity, facility, type}

One of the matrices has been used to calculate several classifications changing the similarity function, the classification method, or the aggregation criteria. In figure 3, as an example, we show a tree corresponding to the case that the similarity function is a distance based on differences and the classification function is the single linkage. Notice that figure 3 shows that the trees are not forced to be binary. There is a node with three subtrees.

In table 2 we compare different classifications obtained from the same data matrix with several classification methods and similarity functions. The table give the distance between pairs of trees. We have used the distance defined in [17] and [1]:

$$d(T, T') = |T \cup T'| - |T \cap T'|$$

Figure 3: A classification tree calculated by differences and single linkage

It is interesting to see that the similarity between classification trees is highly related to the similarity function used rather than to the classification method. This is reflected in table 2 by means of small distances between classifications obtained with the same similarity function (differences or mean character difference), and greater distances when different similarity functions are considered. We can see, for example, that when we choose a distance function by differences (Dif), the trees obtained by means of the arithmetic average (Dif-a) and the median procedure (Dif-m) have a distance of 8. On the other hand, when arithmetic average is considered with several distance functions we have d(Dif-a, MCD-a)=13. Notice that the distances in the frame are greater than the others in the same column/row.

5 Conclusions and future work

In this paper, we have introduced the general classifier system Sedàs that builds n-ary classification trees from data matrices according to three parameters supplied by the expert: a similarity function, a classification criterion and an aggregation criterion. This arquitecture differs from the usual methods by not limiting the type of the values of the attributes, which can be either quantitative, boolean or qualitative (ordered or not) and also by the new method of calculating the similarity between linguistic terms when their domain is ordered. Sedàs has been defined as an open system so that the user can define new similarity functions, classification criteria and aggregation criteria. These methods can be easily incorporated to the system.

Nowadays, we are using Sedàs to study the synthesis of information in the framework of knowledge acquisition. We have studied the synthesis at two different levels: matrix level (EGAC [23]) and classification level ([22]). Sedàs is going to be used in a methodology [26] designed for the analysis and comparison of these tools on synthesis of information.

	Dif-a	Dif-m	Dif-s	MCD-a	MCD-m	MCD-s	Symbols glossary
Dif-a	0	8	9	13	17	13	Dif: Distance by difference
Dif-m		0	13	17	19	17	MCD: Distance by Mean Character Difference
Dif-s			0	12	16	12	a: Classification by Arithmetic average
MCD-a				0	6	0	m: Classification by Median procedure
MCD-m					0	6	s: Classification by single linkage
MCD-s						0	

Table 2: Distances between classifications

6 Acknowledgments

Aïda Valls acknowledges the support of the Department of Software at the UPC. It is also acknowledged the supports of the European Community, under the contract VIM: CHRX-CT93-0401, and the CICYT (TIC96-1038-C04-04).

References

[1] J. P. Barthélemy and F. R. McMorris. The median procedure for n-trees. *Journal of Classification*, 3:329–334, 1986.

[2] J. H. Boose. *Expertise Transfer for Expert System Design*. Elsevier Science Publishers B.V., 1986.

[3] J. H. Boose and J. M. Bradshaw. Expertise transfer and complex problems: using aquinas as a knowledge-acquisition workbench for knowledge-based systems. *Int. Journal of Man-Machine Studies*, 26:3–28, 1987.

[4] D. Driankov and H. H. M. Reinfrank. *An introduction to fuzzy control*. Springer Verlag, USA, 1993.

[5] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic press, New York, 1980.

[6] B. Everitt. *Cluster analysis*. Heinemann Educational Books Ltd., 1977.

[7] B. R. Gaines and M. L. G. Shaw. Knowledge acquisition tools based on personal construct psychology. *The Knowledge Engineering Review*, 8(1):49–85, 1993.

[8] A. D. Gordon. A review of hierarchical classification. *Journal of the Royal Statistical Society. A*, 150(2):119–137, 1987.

[9] C. Hernández, J. J. Sancho, M. A. Belmonte, C. Sierra, and F. Sanz. Validation of the medical expert system renoir. *Computers and Biomedical Research*, 27:456–471, 1995.

[10] F. Herrera, E. Herrera-Viedma, and J. L. Verdegay. A model of consensus in group decision making under linguistic assessments. *Fuzzy Sets and Systems*, 78:73–87, 1996.

[11] G. J. Hwang. Knowledge acquisition for fuzzy expert systems. *Int. J. of Intelligent Systems*, 10:541–560, 1995.

[12] J.Agustí, F. Esteve, P. García, L. Godó, and C. Sierra. Combining multiple-valued logics in modular expert systems. In *Proc. 7th Conference on Uncertainty in AI*, pages 17–25, Los Angeles, 1991.

[13] G. A. Kelly. *The Psychology of Personal Construct.* W.W. Norton & Co.: New York, 1955.

[14] X. Ling and W. G. Rudd. Combining opinions from several experts. *Applied artificial intelligence*, 3:439–459, 1989.

[15] M. Martín, R. Sangüesa, and U. Cortés. Knowledge acquisition combining analytical and empirical techniques. In Morgan-Kauffman, editor, *Proceedings of the eigth international workshop on Machine learning*, pages 657–661, 1991.

[16] P. Murphy and D. W. Aha. UCI repository machine learning databases, irvine, ca: University of california, department of information and computer science, 1994. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[17] D. A. Newmann and V. T. N. (jr). On lattice consensus methods. *Journal of Classification*, 3:225–255, 1986.

[18] E. Plaza. *EAR**: Sistema d'Ajut a l'adquisició i estructuració de coneixements.* PhD thesis, Universitat Politècnica de Catalunya, 1987.

[19] D. Riaño. Knowledge acquisition from data in classification domains. Master's thesis, Facultat d'Informàtica de Barcelona, Universitat Politècnica de Catalunya, 1994.

[20] P. H. A. Sneath and R. R. Sokal. *Numerical taxonomy. The principles and practice of numerical classification.* W.H. Freeman and Co., 1973.

[21] V. Torra. Negation functions based semantics for ordered linguistic labels. *Int. Journal of Intelligent Systems*, 11:975–988, 1996.

[22] V. Torra and U. Cortés. Consenso y clasificación para matrices de datos. In *proceedings of the V conferencia de la asociación española para la inteligencia artificial*, pages 216–225, Madrid, 1993.

[23] V. Torra and U. Cortés. Towards an automatic consensus generator tool: Egac. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5):888–894, 1995.

[24] A. Valls. Sedàs. mòdul general de classificació de dominis poc estructurats. Master's thesis, ETSE, Universitat Rovira i Virgili, 1995.

[25] A. Valls and V. Torra. Discretización de atributos cuantitativos en sistemas basados en el conocimiento. In *Proceedings of the VI Congreso Español de Tecnologías y Lógica Fuzzy*, pages 247–252, Oviedo, 1996.

[26] A. Valls and V. Torra. Knowledge acquisition from multiple experts. In *Proceedings of the first Student Session of the European Summer School Logic, Language and Information (ESSLLI'96)*, 1996.

[27] Y. Yuan and M. J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69:125–139, 1995.