LC-CONCEPTUAL ALGORITHM: Characterization using typical testors by class

José Fco. Martínez-Trinidad[1]
José Ruiz-Shulcloper[1,2]
[1]Centro de Investigación en Computación, IPN
Juan de Dios Batiz s/n esq. Othón de Mendizabal, UPALM; C.P. 07738, México D.F.
Tel.: (+)729-6000, ext. 56573; Fax: (+)729-6000, ext. 56607
e. mail: {fmartine, jruiz}@pollux.cic.ipn.mx
[2]Instituto de Cibernética, Matemática y Física. CITMA, Cuba
e. mail: recpat@cidet.icmf.inf.cu

ABSTRACT: In this paper, we present a new way for construct concepts with the LC-conceptual algorithm using typical testors by class. The typical testors by class are irreducible feature sets such that does not confuse objects in a particular cluster or class with objects in the other clusters generated by the LC conceptual algorithm. As result, we have a conceptual algorithm, which can work with mixed data (quantitative and qualitative features) and missing data in a more appropriated way than other conceptual algorithms. Besides, for this algorithm you do not need predefine the number of clusters to form.

KEYWORDS: conceptual clustering, unsupervised pattern recognition, typical testor.

INTRODUCTION

The widespread use of computers and information technology has made extensive data collection in businesses, manufacturing, geological, medical organizations be a routine task. But the primary challenge that drives the relatively new field of database mining or knowledge discovery from data bases is the extraction of potentially useful information by careful processing and analysis of this data in a computationally efficient manner.

In the real world, a majority of the useful data is described by a combination of numeric and nominal valued features also may occur missing data. For example, if we look at geological data, features such as age, porosity, and permeability are numeric valued, but other features such as rock types, crystalline structure are nominal valued. The majority of the classical conceptual algorithms EPAM, UNIMEM, COBWEB, CLASSIT (see Gennari J. H. et al. 1990), handle exclusively qualitative or quantitative information in the description of the objects. More recent algorithms for example COBWEB/3 of McKusick K. and Thompson K. 1990, LINNEO$^+$ of Béjar Javier and Cortés Ulises. 1992 and K-means conceptual of Ralambondrainy H. 1995, handle mixed features but handle inappropriate the mixed information. In general we can observe that for handle mixed qualitative and quantitative data, the proposed conceptual algorithms attempt the following:

1.-Code qualitative feature values as quantitative values, and apply distance measures used for quantitative situations. The change from qualitative information in quantitative information does not make sense, and the similarity values are not interpreted.

2.-Discretize numeric attributes and apply algorithm that handle only qualitative information. The discretization process often causes loss of important information especially the relative (or absolute) difference between values for the numeric features. Also, the original problem must be modified or changed of representation space.

3.-Generalize comparison functions designed for quantitative features to handle quantitative and qualitative feature values. The functions used for quantitative features are based in distances, which can not be extended for handle qualitative too, because both are in different spaces. Several attempts violate this fact since evaluate the total distance as the addition of the distance between qualitative attributes and the distance between quantitative attributes. And consider that the result is in the original n-dimensional space, where centroides can be calculated.

The authors of this paper proposed the algorithm LC-conceptual (see Martínez-Trinidad J.F. and Ruiz-Shulcloper J. 1997 and Martínez-Trinidad J.F. and Ruiz-Shulcloper J. 1999). This algorithm constitute a new alternative for solve conceptual clustering problems. Eliminate the restrictions of the conceptual algorithms before mentioned. The principal

characteristics of the LC-conceptual algorithm are the following: works in cases where appear features of different nature (mixed data), incomplete object descriptions (missing data), the number of clusters to form is not known a priori.
In this paper a new more appropriated way for construct properties or concepts associated to the clusters formed by the LC conceptual algorithm is proposed.

LC-CONCEPTUAL ALGORITHM

Let M be a set of objects. A description I(O) is defined for every object O∈M, represented by a finite sequence $x_1(O),\ldots,x_m(O)$ of values associated with m features of the set $\Re=\{x_1,\ldots,x_m\}$ where $x_i(O)\in M_i$, and M_i is the set of admissible values for feature x_i. Additionally, we will assume that in M_i (i=1,…,m) there exists a symbol $*$ which denotes *absence of information* (*missing data*). In other words, an object description could be incomplete, i.e., there is at least one variable for which we do not know its value. We will consider that $I(O)\in M_1 x \ldots x M_m$=IRS (Initial Representation Space). The natures of these features are not necessarily the same. For example, some of them could be qualitative (i.e. Boolean, many-valued, fuzzy, linguistic, etc.) and others, quantitative (i.e. integer, real), so we do not restrict IRS to have any algebraic or topologic structure. We do not restrict M_i either to have any *a priori* defined algebraic or logic operations nor any distance (metric).

Let $\Gamma: \bigcup_{T\subseteq R}\left(M_{i_1} x \ldots x M_{i_p}\right)^2 \to L$ a function, where L is a totally ordered set; $T=\{x_{i_1},\ldots,x_{i_p}\}\subseteq\Re$, p≥1, which satisfies

I) Let $T_1,\ldots,T_s$ non empty disjoint subsets of $\Re$, ≤ is the order in L and $T=\bigcup_{i=1}^{s}T_i$, then we have:

if for all h=1,…,s $\Gamma(I|_{T_h}(O_i), I|_{T_h}(O_j))\le\Gamma(I|_{T_h}(O_f), I|_{T_h}(O_g))$, then

$$\Gamma(I|_{T}(O_i), I|_{T}(O_j))\le\Gamma(I|_{T}(O_f), I|_{T}(O_g))$$

II) For all sub-description in $\bigcup_{T\subseteq R}\left(M_{i_1} x \ldots x M_{i_p}\right)$ we have

a) $$\max_{O_j\in M}\left\{\Gamma\left(I|_T(O_i), I|_T(O_j)\right)\right\}=\Gamma\left(I|_T(O_i), I|_T(O_i)\right)$$

b) $$\Gamma\left(I|_{T_i}(O), I|_{T_i}(O)\right)=\Gamma\left(I|_{T_j}(O), I|_{T_j}(O)\right)$$

c) $$\Gamma\left(I|_T(O_i), I|_T(O_i)\right)=\Gamma\left(I|_T(O_j), I|_T(O_j)\right)$$

Γ will be denominated similarity function and it is an evaluation of the similarity degree between any two descriptions of objects belonging to MI, being MI the set of object descriptions to structure. Any restriction of Γ to any subset $T\subseteq\Re$ we will be called a partial similarity function. Often we will consider functions over M_i (over IRS) that not satisfies the properties of a metric. Usually, the information about the objects is giving in a matrix $MI=|x_i(O_j)|_{nxm}$ with n rows (descriptions of the objets) and m columns (values of the features in the objects).
The problem to solve consists in determining the covering set $\{K_1,\ldots,K_r\}$ r>1 in M, as well as the set of concepts or properties associated with K_i *i=1,…,c*.
The algorithm consist of two main steps, the first denominated extensional structuralization, where the clusters are constructed. The second step denominated intentional structuralization, where the properties or concepts associated to each cluster are constructed.
In the first step the clustering criterion proposed in the logical combinatorial pattern recognition are used (see Ruiz-Shulcloper, J and Montellano-Ballesteros, J.J. 1995). In the second step, the typical testors (see Lazo-Cortés, M. and Ruiz-Shulcloper, J. 1995) are used for construct the l-complexes (concepts) associated to each cluster.
A testor is a subset of variables $t=\{x_{i_1},\ldots,x_{i_s}\}\subseteq\Re$ such that if we consider only these variables then there not appear equal objects in different classes. A typical testor is a testor for which none of its proper subsets is a testor too.
The algorithm LC- conceptual is as follows:

Algorithm (characterizing using classical typical testors)

Step1.- The comparison criteria for the variables and the similarity function Γ are determined.
Step2.- Select a clustering criterion Π and compute the clusters $K_1,\ldots,K_c$ in MI.
Step3.- A matrix MA is formed considering the clusters obtained in step 2. MA is similar to MI but first appear the objects in K_1, after the objects in K_2 and so on.
Step4.- Typical testors of MA are calculated.

Step5.- The star $G\left(\mathrm{K_i} \,/\, \mathrm{K_1},...,\hat{\mathrm{K}}_\mathrm{i},...,\mathrm{K_c}\right)$ for each cluster is defined in terms of the respective set of typical testors, using the conditioned REFUNION operator.

The quantity of typical testors in some problems may be very large, so each cluster could have associated a great quantity of concepts or properties. For solve this situation we introduce a criterion for select the best or better concepts.

Since the concepts are constructed starting from the typical testors, we associate to each l-complex a weight, this weight is function of the feature's informational weight (Lazo-Cortés, M. and Ruiz-Shulcloper, J. 1995) calculated starting from typical testors.

So we have that any l-complex l_t constructed starting from the typical testor $t=\{x_{i_1},\ldots,x_{i_s}\}$ will have the weight $p(l_t)=\sum_{j=1}^{s} p(x_{i_j})$, where $p(x_{i_j})$ is the weight of the feature $x_{i_j} \in t$. Thus the l-complexes formed using typical testors that contains features with high apparition frequency in the set of all the typical testors, will have a high weight and vice versa. In this work we use as criterion for select the better l-complexes: select those with maximum weight $p(l_t)$. Although in general we can show (if the specialist wish) all the l-complexes ordered in descending way with regard to their weight.

LC-CONCEPTUAL ALGORITHM USING TYPICAL TESTOR BY CLASS

Recently was proposed the concept of testor by class (see Lazo-Cortés, M. et al. 1998). This concept means a discriminate combination of features but relatives to each class. That is to say, a testor by class for the class $\mathrm{K_i}$ is a discriminate combination of features such that it does not appear in $\mathrm{K_i}$ a subdescription same to anyone in the other classes.

Definition (Lazo-Cortés, M. et al. 1998). A testor by class for the class $\mathrm{K_i}$ is a set $R' \subseteq \Re$ of features such that, it does not exists in $\mathrm{K_i}$ a row same to anyone in the other classes.

Definition (Lazo-Cortés, M. et al. 1998). A testor by class R' for the class $\mathrm{K_i}$ is typical if it does not exists $R'' \subset R'$ such that R'' be testor by class for the class $\mathrm{K_i}$.

Proposition 1 (Lazo-Cortés, M. et al. 1998). Let $T \subseteq \Re$ be a classic testor (Zhuravlev's testor) for MA iff, T is a testor by class for each class in MA.

Proposition 2 (Lazo-Cortés, M. et al. 1998). A Classical testor (Zhuravlev's testor) for MA, no necessarily, is a typical testor by class for someone class.

Corollary 1 (Lazo-Cortés, M. et al. 1998). Let T be a typical testor with minimal length and *t* a typical testor by class with minimal length, then we have that $|t| \leq |T|$.

Proposition 3 (Lazo-Cortés, M. et al. 1998). If T is a classic testor for MA and is a typical testor by class for someone class, then is a classic typical testor for MA

Corollary 2 (Lazo-Cortés, M. et al. 1998). A typical testor by class, or is a typical testor or it does not a classic testor.

As an immediate consequence of the before propositions we can propose the following assertions.

Assertion 1. A l-complex generated starting from a classic testor (Zhuravlev's testor), no necessarily, it is a *l*-complex generated from a typical testor by class for someone class.

Assertion 2. The *l*-complexes generated starting from typical testors by class are less or equal than the complexes generated starting from the classic testors (Zhuravlev's testors) in length.

Assertion 3. A l-complex generated starting from a typical testor by class, or it is a l-complex generated starting from a classic typical testor or it does not.

These assertions are immediate consequence of the testor by class concept. And these have great importance because give us the possibility of characterizing each cluster in terms of relatives features to each cluster such that does not confuse objects of a particular cluster with objects in other clusters.

Definition. Let z_i the family of all the typical testor by class for each class Ki, i=1,...,c. The star of a set $\mathrm{K_i}$ with respect to the sets $\mathrm{K_1},...,\hat{\mathrm{K}}_\mathrm{i},...,\mathrm{K_c}$ denoted $G_{z_i}\left(\mathrm{K_i} \,/\, \mathrm{K_1},...,\hat{\mathrm{K}}_\mathrm{i},...,\mathrm{K_c}\right) = \left\{ \mathrm{RUC}(\mathrm{K_i}) = \underset{x_i \in t}{\wedge} [x_i = R_i] \,/\, t \in z_i, i = 1,...,c \right\}$ is the maximal complexes set under inclusion covering any object in $\mathrm{K_i}$ and not covering any object in $\mathrm{K_1},...,\hat{\mathrm{K}}_\mathrm{i},...,\mathrm{K_c}$, where RUC is the conditioned REFUNION operator (see Martínez-Trinidad J.F. and Ruiz-Shulcloper J. 1997).

Algorithm (characterizing using typical testors by class)

Step1.- The comparison criteria for the variables and the similarity function Γ are determined.

Step2.- Select a clustering criterion Π and compute the clusters $\mathrm{K_1},...,\mathrm{K_c}$ in MI.

Step3.- A matrix MA is formed considering the clusters obtained in step 2. MA is similar to MI but first appear the objects in $\mathrm{K_1}$, after the objects in $\mathrm{K_2}$ and so on until $\mathrm{K_c}$.

Step4.- Calculate the families z_i of all the typical testor by class for each class K_i, i=1,...,c.

Step5.- For each cluster calculate the star $G_{z_i}\left(K_i \,/\, K_1,...,\hat{K}_i,...,K_c\right)$ with all the typical testors, using the conditioned REFUNION operator.

As criterion for select the better concepts the same mentioned in the before section is used.

EXAMPLE

Consider a data set of microcomputers (Michalski 1983), each microcomputer is described by the variables that are showed in the following table I.

	Microcomputer	Display	RAM	ROM	MP	Keys		Microcomputer	Display	RAM	ROM	MP	Keys
1	*Apple II*	*Color TV*	*48K*	*10K*	*6502*	*52*	*7*	*HP-85*	*Built-in*	*32K*	*80K*	*HP*	*92*
2	*Atari 800*	*Color TV*	*48K*	*10K*	*6502*	*57-63*	*8*	*Horizon*	*Terminal*	*64K*	*8K*	*Z80*	*57-63*
3	*Commodore VIC20*	*Color TV*	*32K*	*11K*	*6502A*	*64-73*	*9*	*Ohio Sc. Challenger*	*B&WTV*	*32K*	*10K*	*6502*	*53-56*
4	*Exidi Sorcerer*	*B&WTV*	*48K*	*4K*	*Z80*	*57-63*	*10*	*Ohio Sc. II Series*	*B&WTV*	*48K*	*10K*	*6502C*	*53-56*
5	*Zenith H8*	*Built-in*	*64K*	*1K*	*8080A*	*64-73*	*11*	*TRS-80 I*	*B&WTV*	*48K*	*12K*	*Z80*	*53-56*
6	*Zenith H89*	*Built-in*	*64K*	*8K*	*Z80*	*64-73*	*12*	*TRS-80 III*	*Built-in*	*48K*	*14K*	*Z80*	*64-73*

Table I: Descriptions of Microcomputers.

Boolean comparison was used for the variables *Display* and *MP*. According to this criterion, two values are similar if they are in the same set. The equality criterion (matching) was used for the rest of the variables.

As similarity function it was used a real valued function, that takes values in the interval [0,1]. It is defined as follows:

$$\Gamma(O,O') = \frac{\left|\left\{x_i \,/\, x_i \in \Re, C(x_i(O), x_i(O')) = 1\right\}\right|}{|\Re|}.$$

The β_0-connected clustering criterion was used (Ruiz-Shulcloper; Montellano-Ballesteros 1995, Martínez-Trinidad; Ruiz-Shulcloper 1996). This criterion gave us three clusters (see table II).

Cluster 1		Cluster 2	Cluster 3
Atari 800	Ohio Sc. II Series	Zenith #8	HP-85
Commodore VIC20	TRS-80 I	Zenith #89	
Exidi Sorcerer	Apple II	Horizon	
Ohio Sc. Challenger		TRS-80 III	

Table II: Clusters β_0-connected with β_0=0.6.

The table III and IV we can observe the respective stars for each cluster constructed, using the classical typical testors and typical testors by class respectively.

Cluster 1
1.- ROM=[10 4 11-16 12]
2.- RAM=[32.00..48.00] Keys=[52 57-63 53-56 64-73]
3.- Display=[Color_TV B&W_TV] Keys=[52 57-63 53-56 64-73]
4.- Display=[Color_TV B&W_TV] MP=[6502 6502C Z80 6502A]
5.- Display=[Color_TV B&W_TV] RAM=[32.00..48.00]
Cluster 2
1.- ROM=[1 8 14]
2.- RAM=[48.00..64.00] Keys=[64-73 57-63]
3.- Display=[Built_in Terminal] Keys=[64-73 57-63]
4.- Display=[Built_in Terminal] MP=[8080A Z80]
5.- Display=[Built_in Terminal] RAM=[48.00..64.00]
Cluster 3
1.- ROM=[80]
2.- RAM=[32.00..32.00] Keys=[92]
3.- Display=[Built_in] Keys=[92]
4.- Display=[Built_in] MP=[HP]
5.- Display=[Built_in] RAM=[32.00..32.00]

Table III: Stars for the clusters using the classical testors

Cluster 1
1.- ROM=[4 10 12 11-16]
2.- RAM=[32.00..48.00] Keys=[52 57-63 53-56 64-73]
3.- Display=[Color_TV B&W_TV]
Cluster 2
1.- ROM=[1 8 14]
2.- RAM=[48.00..64.00] Keys=[64-73 57-63]
3.- Display=[Built_in Terminal] Keys=[64-73 57-63]
4.- Display=[Built_in Terminal] MP=[8080A Z80]
5.- Display=[Built_in Terminal] RAM=[48.00..64.00]
Cluster 3
1.- Keys=[92]
2.- MP=[HP]
3.- ROM=[80]
4.- Display=[Built_in] RAM=[32.00..32.00]

Table IV Stars for the clusters using the testors by class

The most important concepts appear with bold letter in tables III and IV. We can see in the table IV that in the Cluster 1 are generated 3 concepts being the concept Display=[Color_TV B&W_TV] a concept not constructed using classical typical testors because Display confuse the computers in the cluster 2 and 3. Using typical testors by class, the feature

Display does not confuse the computers in the cluster 1 with the computers in the clusters 2 and 3. For the cluster 2 are constructed the same concepts using typical classical testors and typical testors by class (see tables III and IV). Finally for the cluster 3 appear Keys=[92] and MP=[HP] as concepts which appears because do not confuse the computer in the cluster 3 with the computers in the clusters 1 and 2.

CONCLUSIONS

In this work was presented a new way for construct concepts o properties associated to the clusters generated by the LC-conceptual algorithm. Before is achieved using the typical testors by class instead of classical typical testor. Using the testors by class give us the possibility of generate concepts relatives to each cluster, with out require that the set of features used for construct the concept discriminates any pair of objects in different classes. Removing this condition and considering for a combination of features only the condition of not confuse objects in one class with objects in the other classes (testor by class). Then the concepts generated are less or equal than the concepts generated using classical typical testors in length, this property is good because we are interested in short concepts. Besides the concepts are specifics for each class, so for each cluster, different feature combinations are considered instead of only one for all the clusters.
As result, we have a conceptual algorithm, which can work with mixed data (quantitative and qualitative features), missing data in a more appropriated way than other conceptual algorithms.

This work was partially financed by Dirección de Estudios de Posgrado e Investigación del Instituto Politécnico Nacional and the CONACyT's Projects No.3757P-A9608 and REDI of Mexico.

REFERENCES

Béjar Javier and Cortés Ulises. 1992. LINNEO+: Herramienta para la adquisición de conocimiento y generación de reglas de clasificación en dominios poco estructurados. En las Memorias del 3er Congreso Iberoamericano de Inteligencia Artificial. La Habana Cuba, pp. 471-481.

Gennari J. H.; Langley P. and Fisher D. 1990. Model of incremental Concept formation. In Jaime Carbonell, MIT/Elsevier Machine Learning. Paradigms and Methods, pp11-61.

Lazo-Cortés, M. and Ruiz-Shulcloper, J. 1995. Determining the feature relevance for non-classically described objects and a new algorithm to compute typical fuzzy testors. *Pattern Recognition Letters* 16, pp. 1259-1265.

Lazo-Cortés, M.; Douglas de la Peña Mercedes; Quintana-Gómez Teresita. 1998. Testores por clase: Una aplicación al reconocimiento de caracteres. Memorias del III Taller Iberoamericano de Reconocimiento de Patrones (México, D.F.) pp. 229-236.

McKusick K. and Thompson K. 1990. Cobweb/3: A portable implementation. Technical report FIA-90-6-18-2, NASA Ames Research Center.

Martínez-Trinidad J.F. and Ruiz-Shulcloper J. 1997. Algoritmo LC Conceptual para el agrupamiento de objetos. Memorias del Simposium Internacional de Computación (México, D.F.), pp. 411-418.

Martínez-Trinidad J.F. and Ruiz-Shulcloper J. 1999. Algoritmo LC Conceptual Duro. Memorias del Simposium Iberoamericano de Reconocimiento de Patrones (La Habana, Cuba), to appear.

Michalski, R. 1983. Automated construction of classifications: conceptual clustering versus numerical taxonomy. IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. PAMI-5, No. 4, July.

Ralambondrainy H. 1995. A conceptual version of the K-means algorithm. Pattern Recognition Letters volume 16, pp. 1147-1157.

Ruiz-Shulcloper, J; Montellano-Ballesteros, J.J. 1995. A new model of fuzzy clustering algorithms, Proceedings of the Third European Congress on Fuzzy and Intelligent Technologies and Soft Computing (Aachen, Germany), 1484-1488.