Determination of Similarity Threshold in Clustering Problems for Large Data Sets

Guillermo Sánchez-Díaz[1] and José F. Martínez-Trinidad[2]

[1] Center of Technologies Research on Information and Systems,
The Autonomous University of the Hidalgo State
Carr. Pachuca – Tulancingo Km. 4.5; C.U
42084 Pachuca, Hgo., Mexico
sanchezg@uaeh.reduaeh.mx
[2] National Institute of Astrophysics, Optics and Electronics,
Luis Enrique Erro No. 1, Sta. María Tonantzintla,
72840 Puebla, Mexico
fmartine@inaoep.mx

Abstract. A new automatic method based on an intra-cluster criterion, to obtain a similarity threshold that generates a well-defined clustering (or near to it) for large data sets, is proposed. This method uses the connected component criterion, and it neither calculates nor stores the similarity matrix of the objects in main memory. The proposed method is focussed on unsupervised Logical Combinatorial Pattern Recognition approach. In addition, some experimentations of the new method with large data sets are presented.

1 Introduction

In unsupervised Pattern Recognition area many algorithms have been proposed [1]. Some of them are based on graph theory. In this paper we will consider the approach based on graph proposed in the Logical Combinatorial Pattern Recognition [2, 3].

In this context, it is assumed that the structure of one universe is not known. To find such structure an initial sample is given, then the problem is precisely to find the classes, the groupings.

The main idea consists in consider the data as vertexes in a graph and the similarity among the objects as edges. In this way the problem of unsupervised classification can be seen as finding subgraphs (clusters) in the initial graph (initial sample).

Note that there exists a natural correspondence among data, their similarity and a graph whose vertexes are objects and the weight of their edges is the similarity between adjacent vertexes.

In this context a parameter β_0 can be introduced for controlling how many similar a pair of objects must be in order to be considered similar. As result then a new graph containing only edges with weight greater or equal than β_0 (the parameter) is obtained.

Therefore depending on the desired closeness in similarity, an appropriate value for this parameter can be chosen by the user and then different graphs are obtained.

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 611–618, 2003.

Now the problem is reduced to find in the resultant graph certain subgraphs. For example, we can find connected components in the graph fixing a certain β_0, where each vertex is an object of the sample and all the edges have a weight greater than β_0. Note that when the value of β_0 is modified the graph may change and then the connected components also can change obtaining a different clustering for each value of β_0. Here rises a natural question, what value of β_0 must be chosen?.

There are many criteria to find subgraphs (clustering criteria), in [4] are presented some of them as β_0-connected components, β_0-compact sets, β_0-strictly compact sets, and β_0-complete maximal sets.

The problem of choosing an adequate value for β_0, without neither to calculate nor to store a similarity matrix for the objects, is studied in this paper. A new automatic method to obtain a similarity threshold β_0 to generate a well-defined clustering (or near to it) for large data set is proposed. This method is based on the maximum and minimum values of similarity among objects and it calculates an intra-cluster similarity for each cluster. Then the method uses this intra-cluster similarity to obtain a global value, which indicates what clustering has the best average intra-cluster similarity. The new method uses the GLC algorithm [5], which generates a clustering based in connected components criterion for large data sets.

2 Related Works

In [6] a method to determine the parameter β_0 for a hierarchical algorithm is proposed. This method uses as similarity between two clusters C_i and C_j the expression $\beta(C_i, C_j) = \max_{\substack{O \in C_i \\ O' \in C_j}} \{\beta(O, O')\}$. Then using a traditional hierarchical algorithm, i.e, grouping the two clusters more similar in each level, a dendograme is built. Using the dendograme the user can choose the parameter β_0 that prefers according to the number of clusters generated by this β_0 value.

In this case, the user determines the value of β_0 in function of the number of cluster that he want to get analyzing the dendograme, the method automatically not determine the value β_0.

Another related work is [7], where some concepts and theorems to demonstrate the number of forms that a sample of objects can be partitioned in β_0-connected components, are introduced. The values β_0 that characterize the different forms of partitioning the sample and the cardinality for each component are also studied.

In this paper, an algorithm such that given a number $0<k<m$, (where m is the number of objects in the sample), the k β_0-connected components are generated, is proposed.

Note that this method is very useful if the number of β_0-connected components to form is known, otherwise when the number of clusters to form is an incognita in the problem we can not use the method.

Although these techniques were not developed in order to process large data sets.

The problem of determining the value of β_0 that generates natural clusters (well-defined clusters) is very important in the context of Logical Combinatory Pattern Recognition approach. Therefore in the next sections a new method to estimate β_0 for large data sets is introduced.

3 Basic Concepts

In this section, the context of an unsupervised classification problem in the Logical Combinatorial Pattern Recognition is explained and also some basic concepts that our method uses to determine β_0 *a*re introduced.

Let $\Omega=\{O_1,O_2,...,O_m\}$be a set of objects and $R=\{x_1,x_2,\ldots,x_n\}$ a set of features. A description $I(O)$ is defined for every $O \in \Omega$ and this is represented by an n-tuple, e.g. $I(O)=(x_1(O),...,x_n(O))\in D_1\times\ldots\times D_n$ (initial representation space), where $x_i(O)\in D_i$; $i=1,...,n$; and D_i is the domain of admissible values for the feature x_i. Di can be a set of nominal, ordinal and/or numerical values.

Hereafter we will use O instead of $I(O)$ to simplify the notation.

Definition 1. A *comparison criterion* [4] is a function $\varphi_i:D_i\times D_i\longrightarrow L_i$ which is associated to each feature x_i ($i=1,...,n$), where:

$\varphi_i(x_i(O),x_i(O))=\min\{y\}$, $y\in L_i$, if φ_i is a *dissimilarity* comparison criterion between values of variable x_i, or

$\varphi_i(x_i(O),x_i(O))=\max\{y\}$, $y\in L_i$, if φ_i is a *similarity* comparison criterion between values of variable x_i, for $i=1,\ldots,n$. φ_i is an evaluation of the similarity or dissimilarity degree between any two values of the variable x_i. L_i $i=1,\ldots,n$ is a total ordered set, usually it is considered as $L_i=[0,1]$.

Definition 2. Let a function β: $(D_1\times...\times D_n)^2\rightarrow L$, this function is named similarity function [8], where L is a total ordered set.

The similarity function is defined using a comparison criterion for each attribute.

Definition 3. In a clustering process (also in supervised classification) will understood by Data set (DS) such collection of object descriptions that the size of the set of descriptions together with the size of the result of the comparison of all object descriptions between objects (*similarity matrix*) does not exceed the available memory size. A Large Data Set (LDS) will be called in the case when only the size of the set of descriptions does not exceed the available memory size. And a Very Large Data Set (VLDS) will be called when both sizes exceed the available memory size [9].

Definition 4. Let β be a similarity function and $\beta_0\in L$ a similarity threshold. Then two objects O_i, $O_j\in\Omega$ are β_0*-similars* if $\beta(O_i,O_j)\geq\beta_0$. If for all $O_j\in\Omega$ $\beta(O_i,O_j)<\beta_0$, then O_i is a β_0*-isolated* object.

Definition 5 (Intra_i criterion). Let $C_1,\ldots,C_k$ be k clusters obtained after apply a clustering criteria with a certain β_0. The intra-cluster similarity criterion (*Intra_i*) is defined as follows:

$$Intra\_i(C_i)=\begin{cases}\max_s & if\ \max_{C_i}=\min_{C_i}=\max_s\\ \max_{C_i}-\min_{C_i} & if\ \max_{C_i}\neq\min_{C_i}\\ \min_s & if\ \max_{C_i}=\min_{C_i}\neq\max_s\end{cases}$$

where $\max_s$ and $\min_s$ are the maximum and minimum similarity values that β, the similarity function, can takes (i.e. $\max_s$=1.0 and $\min_s$=0.0, if L=[0, 1] for example). Besides, $\max_{Ci}$ and $\min_{Ci}$ are the maximum and minimum similarity values among objects belonging to the cluster C_i.

According to Han and Kamber [10] a good clustering method will produce high quality clusters (well-defined clusters), with high intra-cluster similarity and low inter-cluster similarity.

The proposed *Intra_i* criterion was inspired in two facts. First, The *Intra_i* criterion gives a low weight to those clusters where the difference between the maximum and the minimum similarity between objects is low (or null). Also, this criterion gives a high weight to those clusters formed by only one object. This is because these clusters may be outliers or noise, and its global contribution can generates not adequate results.

Second, in this approach of unsupervised classification based in Graph Theory there are two trivial solutions: when $\beta_0=\min_s$, obtaining one cluster with all objects of the sample, and when $\beta_0=\max_s$, generating m clusters, each one formed by only one object.

Then, while β_0 is increased from 0.0 to 1.0 the *Intra_i* takes several values in function of the difference between the maximum and the minimum similarity between objects in the different clusters for each clustering. Therefore, we propose that a reasonable way to determine an adequate value of β_0 (associated to a well-defined clustering) is considering a clustering (a set of clusters) with minimum average intra-cluster similarity.

4 The Method for Determining β_0

In this section, we describe the proposed method to obtain a threshold β_0 such that a well-defined clustering is associated to this value.

4.1 Description of the Method

The proposed method works in the following way. First, the threshold β_0 is initialized with the minimum value of similarity between objects (β_0=0.0). After this, in order to generate several clustering, the method handles a loop, which increases the threshold value with a small constant (INC) and then, a clustering using the GLC algorithm is generated with this β_0 value. For each cluster in a clustering, the maximum and minimum values of similarity among objects are calculated and the *Intra_i* criterion is computed. Finally, the method calculates the average *Intra_i* for each clustering and takes the minimum value obtaining the threshold β_0 that generates a well-defined clustering in the data set. This process continues until INC reaches the maximum similarity value β_0=1.0.

The increase value (INC) for β_0 can takes several values, depending of the accurate required in the problem (0.1, 0.15, 0.5, 0.01, etc., for example). Two ways to handle this parameter in our method are proposed. The first option simply consists in increase INC until reaches the maximum similarity value. The second option is proposed for similarity functions that depend on comparison functions for features (φ_t t=1,…,n) that have the form

$$\beta(O_i, O_j) = \left| \{ x_t / x_t \in R, x_t(O_i), x_t(O_j) \text{ are similar according to } \varphi_t \} \right| / n \quad \text{or}$$

$$\beta(O_i, O_j) = \sum_{x_t \in R} \varphi(x_t(O_i), x_t(O_j)) / n$$

where n is the number of attributes. In this case, the value of INC is fixed as INC=1.0/n, if $\beta_0 \in$ [0.0, 1.0]. In this way, the values for β_0 among 1.0/h and 1.0/(h+1), h=1,...,n−1 do not generate any change in the clustering.

The method proposed in this paper is the following:

```
Input: Ω (data set), INC (β0 increase value)
Output: β0 (threshold calculated)

β0= mins
   Repeat
      CCk=GLC(β0)
      Cik=Clusters(CCk, i), i=1,...,hk
      maxci-ik=Max(Cik, i), i=1,...,hk
      minci-ik=Min(Cik, i), i=1,..., hk
      valueik=Intra_i(Cik, maxci-ik , minci-ik)
      meanCCk=∑ valueik / |CCk|
      β0=β0 + INC
   until β0= maxs
β0=β0_min_average_Intra_i(meanCCk)
```

The GLC(β_0) function calls to GLC algorithm in order to build a clustering (i.e. CC_k) with a specific similarity threshold β_0, applying the connected components criterion. The function Clusters(CC_k, i) returns the cluster i, from the clustering CC_k. The functions Max(C_{ik}, i) and Min(C_{ik}, i) return the maximum and minimum values of similarity among objects for the cluster i, in the clustering CC_k. The *Intra_i*(C_{ik}, $\max_{ci\text{-}ik}$, $\min_{ci\text{-}ik}$) function calculates and return the intra-cluster criterion for the cluster i. $|CC_k|$ denotes the number of clusters in a clustering. Finally, β_0_min_mean_Intra_i(meanCC$_k$) obtains and return the minimum average value of the *Intra_i* criterion.

The threshold β_0 obtained by the method indicates that the clustering associated to β_0 is a well-defined clustering.

5 Experimental Results

In this section, two examples of applications of the method to data sets and large data sets are presented.

The first data set (DS1) contains 350 numerical objects in 2D, and it is shown in figure 1(a). The clusters shown in this figure have several shapes, sizes and densities, and they are not lineally separable. The method was applied to DS1 and 7 clustering were obtained. The thresholds β_0 obtained in each clustering (CC_i) are as follows:

CC_1: β_0=0.00; No. Clusters = 1; Average *Intra_i*= 0.8631;
CC_2: β_0=0.89; No. Clusters = 2; Average *Intra_i*= 0.4777;
CC_3: β_0=0.90; No. Clusters = 3; Average *Intra_i*= 0.3095;
CC_4: β_0=0.93; No. Clusters = 4; Average *Intra_i*= 0.2772;
CC_5: β_0=0.94; No. Clusters = 5; Average *Intra_i*= 0.2378;
CC_6: β_0=0.98; No. Clusters = 86; Average *Intra_i*= 0.6418;
CC_7: β_0=0.99; No. Clusters = 350; Average *Intra_i*= 1.0000;

The minimum value of these averages determines the value β_0=0.94, which corresponds to a well-defined clustering (i.e. CC_5), formed by the clusters shown in the figure 1(b). The same well-defined clustering was obtained in [5]. For this example, the value INC=0.01 was employed.

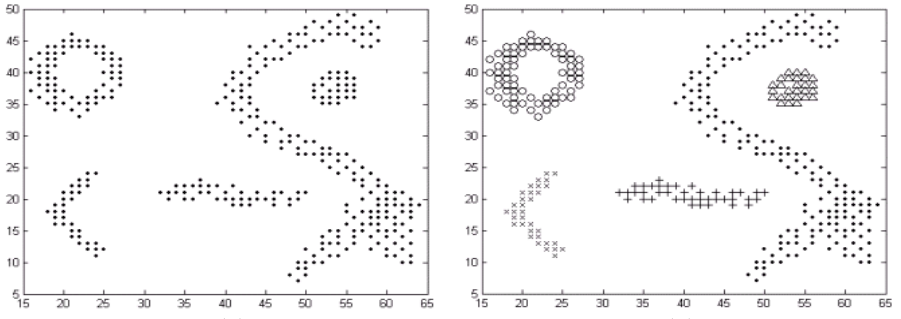

Fig. 1. (a) The objects corresponding to DS1; (b) Clustering obtained with β_0=0.94, for DS1 (well defined clustering discovered)

The second data set used for experimentation was a Mushroom database [11]. The mushroom data set (a LDS, according to our definitions) contains records with information that describes the physical characteristics of a single mushroom (e.g. color, odor, shape, etc.). This data set contains 8124 records. All attributes are categorical, and contain missing values. Each record also contains a poisonous or edible label for the mushroom.

In order to show the behavior of the proposed method, several clustering were obtained with their respective β_0 and them are presented in tables 1 and 2. Again, we show the well-defined clustering generated for DS2, which corresponds with a

β_0=0.9545 value, generating 23 clusters, with an average *Intra_i* value (AIV) of 0.2115. The same well-defined clustering was obtained in [9]. For this experimentation the value INC=0.0454=1.0/22 (number of features = 22) was used.

The cases with β_0=0.00, AIV=0.8182 and β_0=0.99, AIV=1.0 are not shown, because in the first case the clustering obtained has all the objects. And, for the second case each cluster contains only one object.

The notation handled in tables 1 and 2 is as follows: CN denotes the cluster number; NE indicates the number of edible mushrooms; NP denotes the number of poisonous mushrooms, and AIV indicates the average *Intra_i* value.

The experiments were implemented in C language on a personal computer with Pentium processor at 833 Mhz and 128 RAM Megabytes.

Table 1. Clusters obtained for DS2 with β_0=0.6810; β_0=0.7273; and β_0=0.7727;

β_0=0.6810, AIV=0.3788			*β_0=0.7273,* AIV=0.2818			*β_0=0.7727,* AIV=0.3182		
CN	NE	NP	CN	NE	NP	CN	NE	NP
1	4016	3880	1	4016	2576	1	392	808
							0	
2	192	0	2	0	1296	2	0	1296
3	0	36	3	192	0	3	48	1768
			4	0	36	4	48	0
			5	0	8	5	192	0
						6	0	36
						7	0	8

Table 2. Clusters obtained for DS2 with β_0=0.8182; β_0=0.8636; β_0=0.9091; and β_0=0.9545

β_0=0.8182, AIV=0.2238			*β_0=0.8636,* AIV=0.2273			*β_0=0.9091,* AIV=0.2208			*β_0=0.9545, AIV=0.2115*		
CN	NE	NP	CN	NE	NP	CN	NE	NP	CN	NE	NP
1	2848	808	1	896	448	1	0	256	1	0	256
2	768	0	2	768	0	2	704	0	2	512	0
3	0	1296	3	1728	0	3	768	0	3	768	0
4	0	1728	4	0	1296	4	96	0	4	96	0
5	48	0	5	0	288	5	96	0	5	96	0
6	48	0	6	192	0	6	1728	0	6	192	0
7	0	32	7	0	1728	7	0	1296	7	1728	0
8	0	8	8	48	0	8	0	192	8	0	1296
9	192	0	9	32	72	9	0	288	9	0	192
10	288	0	10	48	0	10	192	0	10	0	288
11	0	36	11	0	32	11	0	1728	11	192	0
12	0	8	12	0	8	12	48	0	12	0	1728
13	16	0	13	192	0	13	32	72	13	48	0
			14	288	0	14	48	0	14	0	72
			15	0	36	15	0	32	15	48	0
			16	0	8	16	0	8	16	0	32
			17	16	0	17	192	0	17	0	8
						18	288	0	18	192	0
						19	0	36	19	288	0
						20	0	8	20	32	0
						21	16	0	21	0	36
									22	0	8
									23	16	0

6 Conclusions

The method proposed in this paper allows obtaining a well-defined clustering, based in an intra-cluster similarity criterion.

The method gives a threshold value β_0 to obtain a well-defined clustering for large data sets.

The method does not establish any assumptions about shape, size or cluster density characteristics of the resultant clusters in each generated clustering. However, the proposed method is still susceptible to noise.

Our method uses the β_0-connected component criterion for clustering. As future work we will work in the generalization of the proposed *Intra_i* criterion in order to handle other clustering criteria as those exposed in [4].

Acknowledgement. This work was financially supported by CONACyT (Mexico) through project J38707-A.

References

1. Duda R., Hart P., and Stork D.: Pattern Classification (2nd ed). Wiley, New York, NY (2000)
2. Martínez-Trinidad J. F. and Guzmán-Arenas A.: The logical combinatorial approach to pattern recognition an overview through selected works. Pattern Recognition 34(4) (2001) 741–751
3. Ruiz-Shulcloper J. and Mongi A.: A. Logical Combinatorial Pattern Recognition: A Review. In Recent Research Developments in Pattern Recognition, Ed. Pandalai, Pub. Transword Research Networks, USA. (To appear)
4. Martinez Trinidad J. F., Ruiz Shulcloper J., and Lazo Cortes M.: Structuraliation of universes. Fuzzy Sets and Systems. Vol. 112 No. 3 (2000) 485–500
5. Sanchez-Diaz G. and Ruiz-Shulcloper J.: MID mining: a logical combinatorial pattern recognition approach to clustering large data sets. Proc. 5th Iberoamerican Symposium on Pattern Recognition, Lisbon, Portugal (2000) 475–483
6. Pico Peña R.: Determining the similarity threshold for clustering algorithms in the Logical Combinatorial Pattern Recognition through a dendograme. Proc. 4th Iberoamerican Simposium of Pattern Recognition. Havana Cuba (1999) 259–265
7. Reyes Gonzales R. and Ruiz-Shulcloper J.: An algorithm for restricted structuralization of spaces. Proc. 4th Iberoamerican Simposium of Pattern Recognition. Havana Cuba (1999) 267–278
8. Ruiz-Shulcloper J. and Montellano-Ballesteros J.: A new model of fuzzy clustering algorithms. Proc. of the 3rd EUFIT, Aachen, Germany (1995) 1484–1488
9. Ruiz-Shulcloper J., Sanchez-Diaz G. and Abidi M.: Clustering Mixed Incomplete Data. Heuristics & Optimization for Knowledge Discovery. Idea Group Publishing, USA (2002) 88–106
10. Han J. and Kamber M.: Data mining: concepts and techniques. The Morgan Kaufmann Series in Data Management Systems, Jim Gray Series Editor (2000)
11. Blake C.L. and Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)