From Generalization to Clustering in the Relational Database Context [1]

Marie CHAVENT and Véronique STEPHAN
Action SODAS, INRIA Rocquencourt, 78153 Le Chesnay cedex, France
Lise Ceremade, Université de Paris IX Dauphine,
Place du Maréchal De Lattre de Tassigny, 75016 Paris, France

Abstract

We present a two steps process to analyze complex data called symbolic objects. At the first step we find the best symbolic description representing groups of individuals retrieved from a relational database. We define operators to retrieve these groups of data and we present a generalization/specialization process to describe each of them. At the second step, we perform a classification of these groups according to their symbolic description. We define a divisive clustering method in the particular case of symbolic data.

Key Words : Symbolic Data Analysis, Relational Database, Generalization/Specialization, Divisive hierarchical clustering.

1 Introduction

First, we consider initial data stored in a relational database which can be voluminous. Users can be either interested in analyzing these data or in some cases they want to perform more sophisticated analysis based on aggregation into groups. Such analyses can't be performed with classical methods. We present here a two steps process that enables easily a user to answer to more sophisticated questions.

The first step consists in retrieving groups of individuals and to define a good generalization which summarizes each group of items. In the area of Knowledge Discovery in Databases, some authors deal with a similar problem, called query summarization. Hwang and al. (see [8]) refer to several approaches to summarize efficiently a set of tuples in relational databases (using generalization trees on domains of attributes). In the area of Machine Learning, Michalski (see [10]) has defined generalization rules to summarize

[1] Proceedings of the conference on Knwoledge Extraction and Symbolic Data Analysis : KESDA'98, Eurostat's collection, Theme 9 : Research and development, 105–117

a set of classical descriptions. The way we build descriptions of groups is guided by a volume criterion which reduces over-generalization. Indeed, the quality of the descriptions produced is of great importance as they are used as input for the clustering method.

The input of the clustering step are the symbolic data performed by the generalization step. Data are called symbolic when they are described on each variable with a set of values or with a set of weighted values. We propose a divisive clustering algorithm which reverses the process of agglomerative hierarchical clustering by starting with all objects in one cluster and subdividing successively each cluster into smaller ones. Williams and Lambert (see [14]) proposed a first monothetic divisive clustering method in the particular case of binary data. A cluster is called monothetic if a conjunction of logical properties is both necessary and sufficient for membership in the cluster (see [11]). More recently, a parametric monothetic clustering method has been proposed for quantitative data (see [4]).

In this paper, we use a non parametric monothetic clustering method (see [3]) in cases of symbolic data, which aims at finding at each step a partition and a symbolic interpretation of the corresponding clusters.

2 Generalization of groups of individuals retrieved from a relational database

2.1 Selection of a population from the database

A statistical population is a set of units (also called individuals) having common properties. In our process, we consider the initial population, called Ω, to be the result of a user-query which defines the task-relevant data and its kind of grouping. Each individual belongs to a class, according to its value observed on the second attribute C of the select clause. So we associate to Ω a structure into J groups and the attribute C is such that :

$$\forall \omega \in \Omega,\ j \in C(\omega) \Leftrightarrow \omega \in G_j$$

Ω is described by a list of variables $Y = (Y_1, \ldots, Y_i, \ldots, Y_p)$. Each $Y_i : \Omega \rightarrow O_i$ corresponds to an attribute in the select clause and O_i is the space of observations of Y_i, deduced from the associated column of the extension of the query (Table 1).

Dealing with small datasets, we perform the user-query and all the selected information is stored in working memory. When the amount of data

	C	Y_1	$\ldots$	Y_i	$\ldots$	Y_p
ω_1	$\ldots$			$\ldots$		
$\ldots$	$\ldots$			$\ldots$		
ω_j	$C(\omega_j)$	$\ldots$	$\ldots$	$Y_i(\omega_j)$		
$\ldots$						
ω_N						

Table 1: Result of the sql query

is great, we have considered the use of random sampling. As we don't previously know the size of the population, we perform a random sampling using reservoir algorithm defined by Waterman (see [13]). The advantage of such an algorithm is that it uses only one pass to sample the result of the query.

We adapt this kind of algorithm to sample individuals from each group $G_1, \ldots, G_J$. A reservoir of size m is defined for each G_j. The random sampling is performed independently for each group $G_1, \ldots, G_J$ in the same pass.

2.2 Representation of a group G_j by an assertion

The symbolic description model used as output of our generalization is the model of symbolic objects introduced by E. Diday (see [5]). Here we focus on building particular kinds of symbolic objects, called assertions, which deal with classes (or sets) of individuals.

The output of the generalization/specialization process is the set $\mathcal{A} = \{a_1, \ldots, a_J\}$ where a_j is the assertion built from G_j. An assertion a_j is described on p variables $(Y_1, \ldots, Y_i, \ldots, Y_p)$, where :

$$Y_i \,:\, \mathcal{A} \rightarrow \mathcal{D}_i$$

and $\mathcal{D}_i$ can be $\mathcal{P}(O_i)$ or the set of probability distributions defined on O_i. An assertion a_j is described on $Y = (Y_1, \ldots, Y_p)$ by a vector :

$$Y(a_j) = (Y_1(a_j), \ldots, Y_i(a_j), \ldots, Y_p(a_j))$$

also noted :

$$\delta_j = (\delta_j^1, \ldots, \delta_j^i, \ldots, \delta_j^p)$$

	Y_1	$\ldots$	Y_i	$\ldots$	Y_p
a_1			$\ldots$		
$\ldots$			$\ldots$		
a_j	$\ldots$	$\ldots$	$Y_i(a_j) = \delta_j^i$		
$\ldots$					
a_J					

Table 2: Data matrix obtained by the generalization process

The output of our generalization/specialization process can be represented by the following data matrix (Tab. 2) where $\delta_j^i \in \mathcal{D}_i$.

For example, we want to cluster positions in a factory according to informations on employees' careers. Let Y_1 = Training, Y_2 = YearsService and Y_3 = Age, be the variables defined in the select clause of the sql-query. Ω is defined as the set of employees stored in the database. $\mathcal{A}$ is the set of assertions corresponding to positions in the factory. An assertion $a_j \in \mathcal{A}$ corresponds to the set of employees belonging to the same j^{th} position. The description δ_j of assertion a_j could be the following :

$$\delta_j = (\{\text{University, Engineering school}\},\ [7, 15],\ [34, 40])$$

This assertion can also be represented as a conjunction of properties :

a_j : [Training $\in$ {University, Engineering school}] $\wedge$
[YearsService $\in$ [7,15]] $\wedge$ [Age $\in$ [34,40]]

This conjunction of properties is interpreted as follows : the employees observed at the j^{th} position have a University or Engineering school training, they have been in the factory between 7 and 15 years and they are between 34 and 40 years old. Frequencies can also be added for each modality of a nominal variable :

a_j : [Training $\in$ {University (0.98), Engineering school (0.02)}] $\wedge$
[YearsService $\in$ [7,15]] $\wedge$ [Age $\in$ [34,40]]

2.3 Generalization/specialization process

We present the way we generalize each group of data by a description which enables to express the variation within the group. This process is performed in an unsupervised way. Each group is considered independently from each other. Indeed, we look for an homogeneous description of a group and not a discriminant one. The first step consists in generalizing each group G_j by an assertion a_j. This generalization is performed for each variable, using the following rules :

$$Y(a_j) \; = \; (\delta_j^1, \ldots, \delta_j^i, \ldots, \delta_j^p)$$

$$\text{where } \delta_j^i = \oplus(\{Y_i(\omega) \;\mid\; \omega \in G_j\}) \;\; i = 1, \ldots, p$$

$$\text{and } \oplus\left(\{Y_i(\omega)\}_{G_j}\right) = \begin{cases} [\min(\{Y_i(\omega)\}_{G_j}), \max(\{Y_i(\omega)\}_{G_j})] & Y_i \text{ numeric} \\ \{Y_i(\omega)\}_{G_j} & Y_i \text{ nominal} \end{cases}$$

However, untypical individuals make the generalization too broad, adding a lot of values rarely observed in the class. For example, let us define two groups described by the following assertions a_1 and a_2 :

a_1 : [Training ∈ {University, Engineering school}] ∧
[YearsService ∈ [7,10]] ∧ [Age ∈ [34,40]]

a_2 : [Training ∈ {University, Engineering school}] ∧
[YearsService ∈ [8,15]] ∧ [Age ∈ [34,40]]

In the first group, employees are mostly younger than 36 years old and 98% have an Engineering school training. In the second group, employees are mostly older than 38 years old and 98% have a University training. If we compare a_1 and a_2, they are similar while initial groups aren't the same at all. In so far as untypical individuals are less informative regarding a homogeneous description of the class, we propose a specialization step, where the final description is more characteristic.

To perform the specialization step, we adapt a volume criterion (see [2]), which measures a generality index of an assertion a_j :

$$vol(a_j) = \prod_{i=1}^{p} Et(\delta_j^i)$$

$$\text{where } Et(\delta_j^i) = \begin{cases} card(\delta_j^i) & \text{nominal case} \\ \max(\delta_j^i) - \min(\delta_j^i) & \text{numeric case} \end{cases}$$

However, this volume criterion can't be applied with both nominal and numeric variables because of scale problems. Indeed, this criterion measures in a different way a generalization on a numeric variable (length of an interval) and on a nominal one (set of values). These two different scales may involve an unbalanced reduction giving preference to the loss of numeric values. To overcome problems of scale between nominal and numeric variables, we code numeric variables into ordinal ones. This coding allows us to have a homogeneous criterion among all variables without giving preference to one particular kind. This coding is performed by a recursive partitioning. It aims at finding a set of intervals under uniform hypothesis. Indeed, we search for a uniform distribution of observed data on each interval found.

After coding the numeric variables, we reduce the initial assertion into a more homogeneous one. We fix a threshold α, which is the minimum covering power of the assertion. The covering power of an assertion a_j is based on the extension of a_j, which is the set of individuals verifying the description δ_j :

$$ext_{G_j}(a_j) \;=\; \{\omega \in G_j \;\mid\; \forall i \in \{1,\ldots,p\},\; Y_i(\omega) \in \delta_j^i\}$$

The covering power of an assertion a_j, noted $Rec(a_j)$ is defined as :

$$Rec(a_j) = \frac{card(ext_{G_j}(a_j))}{card(G_j)}$$

We define $\mathcal{V} = \mathcal{P}(\delta_j^1 \cup \ldots \cup \delta_j^p)$. The covering set of an element $V \in \mathcal{V}$ on G_j is defined as :

$$cov(V) = \cup_{v \in V} \; \{\omega \in G_j \;\mid\; Y_i(\omega) = v\}$$

where Y_i is the variable such that $v \in \delta_j^i$. If Y_i is a quantitative variable, we rather tests if $Y_i(\omega) \in v$ where v is an interval obtained by the discretization step performed on δ_j^i.

The algorithm of specialization consists in computing all admissible values set removals $V \in \mathcal{V}$. V is called an *admissible removal* if the new assertion a_j^* obtained after removing V from δ_j is such that :

$$Rec(a_j^*) \geq \alpha$$

We significantly reduce the complexity by computing bests admissible removals, according to our volume criterion. This computation is performed using simplification rules in an iterative search. At each step k, we build L_k which is the set of all admissible removals such that :

$$\forall V \in L_k,\ card(V) = k \text{ and } \frac{card(cov(V))}{card(G_j)} \leq (1-\alpha)$$

We associate $V^* \in \mathcal{V}$ to V. V^* is equal to :

$$V^* = \cup_{i=1,\dots,p} \{v \in \delta_j^i \setminus \oplus(\{Y_i(\omega) \mid \omega \in G_j \setminus cov(V)\})\}$$

For each $V \in \mathcal{V}$, we can define the corresponding assertion $a_{j,*}$ which is described by $\delta_{j,*} = (\delta_{j,*}^1, \dots, \delta_{j,*}^p)$ such that :

$$\delta_{j,*}^i = \delta_j \setminus V^* \ \ i = 1, \dots, p$$

In our algorithm the complexity of computing V^* from V is bypassed according to several propositions (see [12]).

We define the basic lines of our iterative search. The first step consists in computing L_1. L_1 corresponds to all singletons such that :

$$L_1 = \{V \in \mathcal{V} \ \mid\ card(V) = 1 \text{ and } \frac{card(cov(V))}{card(G_j)} \leq (1-\alpha)\}$$

For each $V \in L_1$, we associate an element $V^* \in \mathcal{V}$ which is initialized to V.

Given this first step, we iterate by building L_2 which corresponds to a subset of the set of couples values $V = \{v_k,v_l\}$, where $\{v_k\}, \{v_l\} \in L_1$. Let us define $V_1 = \{v_k\}$ and $V_2 = \{v_l\}$. We define the following rules to decrease the complexity of the reduction algorithm (see [12]) :

- if $cov(V) = cov(V_1)$ (resp. $cov(V_2)$), we update V_1^* (resp. V_2^*) :

  $$V_1^* \leftarrow V_1^* \cup \{v_l\} \text{ (resp. } V_2^* \leftarrow V_2^* \cup \{v_k\})$$

- else if $cov(V) \leq 1-\alpha$, V is added to L_2 and V^* is initialized to $V_1^* \cup V_2^*$.

We iterate considering the removal due to three and more combination values by building $L_3, \dots, L_k, \dots, L_m$ where L_k corresponds to all admissible removal of k values from a_j. L_k is computed using the previous rules from L_{k-1}. The search is stopped after m iterations when $L_m = \emptyset$. The end of the specialization step consists in choosing the assertion $a_{j,l}$ corresponding to V_l in $\{L_1 \cup \ldots \cup L_{m-1}\}$ which provides the minimum volume :

$$a_j \leftarrow \arg\min(\{vol(a_{j,*}) \ \mid \ V \in L_1 \cup \ldots \cup L_{m-1}\})$$

We give a small example with simulated numeric data (Fig. 1).

Figure 1: Specialization of the initial description

The generalization step gives the following description for the group :

$$[Y_1 \in [4.21, 16.66]] \wedge [Y_2 \in [1.88, 8.02]]$$

First we perform a coding of the two numeric variables. After the specialization step, keeping a covering power of 90%, we describe the same group as follows :

$$[Y_1 \in [5.84, 13.96]] \wedge [Y_2 \in [3.26, 6.528]]$$

Taking the employees careers example, this step of specialization provides two new assertions. Rarely seen employees' profiles have been removed from the final description :

a_1 : [Training =Engineering school] $\wedge$ [YearsService $\in$ [7,8]] $\wedge$ [Age $\in$ [34,37]]

a_2 : [Training =University] $\wedge$ [YearsService $\in$ [10,15]] $\wedge$ [Age $\in$ [36,40]]

At the end of our generalization/specialization step, an output assertion $a_j \in \mathcal{A}$ corresponds to the minimal volume of description whose power covering on G_j is α.

3 Clustering of symbolic data by a divisive approach

Let $\mathcal{A} = \{a_1, \ldots, a_J\}$, the J assertions obtained by the generalization/specialization process. Each assertion is described on p variables $Y_1, \ldots, Y_p$ by a vector :

$$\delta_j = (\delta_j^1, \ldots, \delta_j^p) \in \mathcal{D} = \mathcal{D}_1 \times \ldots \times \mathcal{D}_p$$

Let $P_K = (C_1, \ldots, C_K)$ be a K-clusters partition of $\mathcal{A}$:

$$\begin{aligned} C_k \cap C_{k'} &= \emptyset \qquad (1) \\ \bigcup_{k=1,\ldots,K} C_k &= \mathcal{A} \qquad (2) \end{aligned}$$

At each step of the divisive algorithm, a new (K+1)-clusters partition is obtained by splitting a cluster $C_k \in P_K$ in two new clusters C_k^1 and C_k^2.

The general algorithm is the following :

```
Initialization: $P_1 = \mathcal{A}$; set K=1;
If $K \leq J - 1$ then:
    (i) Choose $C_k \in P_K$ such that the split $(C_k^1, C_k^2)$ of $C_k$ maximizes
        $\Delta(C_k) = I(C_k) - I(C_k^1) - I(C_k^2)$    (3)
    (ii) $P_{K+1} = P_K \cup \{C_k^1, C_k^2\} - \{C_k\}$
    (iii) $K \leftarrow K + 1$
```

The algorithm stops after $J - 1$ iterations if the J assertions have different descriptions in $\mathcal{D}$. Usually, the users are interested in few clusters partitions and the algorithm stops after $L - 1$ iterations, $L < J$. In this case, the singletons of the hierarchy are the L clusters of the partition obtained in the last iteration of the algorithm.

In order to define step (i), we will define :

- the quality criterion I of a cluster C_k. It will be defined as an extension of the inertia criterion to the case of a dissimilarity matrix
- how to split a cluster C_k and how to find its best split (C_k^1, C_k^2)

3.1 Extension of the inertia criterion

Let $D = \{d_{jj'}\}$ a dissimilarity matrix defined on $\mathcal{A} = \{a_1, \ldots, a_J\}$:

$$d_{jj'} = d(a_j, a_{j'}) = d(a_{j'}, a_j) \geq 0,\ d_{jj} = 0$$

Each assertion is weighted by a real value p_j ($j = 1, \ldots, J$), for instance $p_j = \frac{1}{J}$.

Definition 1 *The quality I of a cluster C_k is defined by :*

$$\begin{aligned} I(C_k) &= \sum_{a_j \in C_k} \sum_{a_{j'} \in C_k} \frac{p_j p_{j'}}{2\mu_k} d_{jj'}^2 \qquad (4) \\ \mu_k &= \sum_{a_j \in C_k} p_j \end{aligned}$$

In the particular case of quantitative data, $\delta_j \in \mathbf{R}^p$ and the criterion $I(C_k)$ is the inertia of the cluster C_k.

In the general case of symbolic data, d is a distance or a dissimilarity between symbolic descriptions (see [7], [9]).

3.2 Splitting a cluster

In our approach, we split a cluster C according to a binary question of the form

$$\{\text{Is } Y_i \leq \ \text{c ?}\}$$

where $c \in O_i$ is called the cut value.

Breiman, Friedman, Olshen and Stone (see [1]) defined the notion of binary question in the case of classical data.

Here we propose an extension of this definition in the case of symbolic data e.g. in the case of objects described by an interval of values or by set of weighted values :

$$Y_i : \mathcal{A} \rightarrow \mathcal{D}_i$$

where $\mathcal{D}_i$ can be :

- the set of closed and bounded intervals of **R**

- the set of probability distributions on O_i

An assertion $a \in C$ answers "yes" or "no" to the binary question according a binary function $Q_c : \mathcal{A} \rightarrow \{true, false\}$. The split (C_1, C_2) of C is induced by the binary question {Is $Y_i \leq c$?} as follows :

$$\begin{aligned} C_1 &= \{a \in C \ / \ Q_c(a) = true\} \\ C_2 &= \{a \in C \ / \ Q_c(a) = false\} \end{aligned}$$

• If the assertions are described on the variable Y_i by a real interval and

- $O_i = \mathbf{R}$,

- $\mathcal{D}_i$ is the set of closed and bounded intervals of **R**,

- $Y_i(a) = [i_a, s_a] \in \mathcal{D}_i$ and $m_a = \frac{i_a + s_a}{2}$ is the middle of $[i_a, s_a]$,

then the assignment rule is the following :

$$\begin{aligned} Q_c(a) = 1 \quad &\text{if} \quad m_a \leq c \\ Q_c(a) = 0 \quad &\text{if} \quad m_a > c \end{aligned}$$

See for instance Fig. **??**.

Figure 2: a is assigned to C_1 because $m_a = 171 \leq 172$

• If the assertions are described on the variable Y_i by a discrete probability distribution and

- O_i is a finite and ordered set,

- $\mathcal{D}_i$ is the set of probability distributions on O_i,

- $Y_i(a) = \delta_a$ and δ_a is a function defined from O_i to $[0, 1]$ such that :

$$\sum_{x \in O_i} \delta_a(x) = 1$$

then, the assignment rule is the following :

$$\begin{aligned} Q_c(a) = 1 \quad & \text{if} \quad \sum_{x \leq c} \delta_a(x) \geq 1/2 \\ Q_c(a) = 0 \quad & \text{if} \quad \sum_{x \leq c} \delta_a(x) < 1/2 \end{aligned}$$

3.3 Choice of the best split

Let C be a set of n assertions. The goal is to find the split $C = (C_1, C_2)$ of smallest "within inertia":

$$\begin{aligned} W(C_1, C_2) &= I(C_1) + I(C_2) \\ &= \sum_{a_j \in C_1} \sum_{a_{j'} \in C_1} \frac{p_j p_{j'}}{2\mu_1} d_{jj'}^2 + \sum_{a_j \in C_2} \sum_{a_{j'} \in C_2} \frac{p_j p_{j'}}{2\mu_2} d_{jj'}^2 \qquad (5) \end{aligned}$$

In the Edward and Cavalli-Sforza method (see [6]) one chooses the optimal split (C_1, C_2) of C among the $2^{n-1} - 1$ possible splits. It is clear that the amount of calculation needed when n is large will be prohibitive.

In our approach, to reduce the complexity, we choose the best split among all the splits induced by the set of binary questions.

- If a variable Y_i is described by real intervals, there will be at most $z_i = n - 1$ different splits (C_1, C_2) induced by this variable. Indeed, whatever the cut value c between two consecutive m_a may be, the split induced is the same. In order to ask only $n-1$ questions to generate all these splits, we decide to use the $n-1$ cut values c, chosen as the middle of two consecutive m_a. Indeed, if the n values m_a are different, there are $n-1$ cut values on Y_i.

- If a variable Y_i is described by a discrete probability distribution on O_i (finite and ordered) and $M = card(O_i)$, there is $M - 1$ different binary questions and at most $z_i = M - 1$ different splits (C_1, C_2) induced by this variable.

Finally, if there are p variables, we choose among the $z_1 + \ldots + z_p$ corresponding splits (C_1, C_2), the split of smallest "within inertia".

3.4 The output

The output of the divisive algorithm is a hierarchy whose clusters C are indexed by $\Delta(C) = I(C) - I(C_1) - I(C_2)$. The number $L-1$ of iterations is chosen by the user and the singletons of the hierarchy are the L clusters of the last partition. This hierarchy is also a decision tree. The L clusters are the leaves and the nodes are the binary questions selected by the algorithm. Each cluster is characterized by a production rule defined according to the binary questions leading from the root to the corresponding leaves.

4 A simple example

The following example is an illustration of the two step process.

The Table 3 gives an example of a data matrix resulting from a sql query on 1000 employees. The attribute C associates to each employee a number between 1 and 50, corresponding to the position of the employee in the factory.

	C	Training	YearsService	Age
ω_1	2	University	15	40
$\vdots$		$\vdots$	$\vdots$	$\vdots$
ω_{1000}	43	Engineering school	7	29

Table 3: Result of the sql query on 1000 employees

The Table 4 gives the 50 assertions (corresponding to the 50 previous factory position), obtained by the generalization/specialization process.

	Training	YearsService	Age
a_1	University (1), Engineering school (0)	[7,8]	[34,37]
$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_{50}	University (0.3), Engineering school (0.7)	[2,5]	[25,28]

Table 4: Assertions obtained by the generalization/specialization process

The Fig. 3 gives the hierarchy obtained with the 50 previous assertions after two splits.

The two binary questions constructing the hierarchy are :

Figure 3: The hierarchy and the three clusters obtained by the clustering process

$$\{ \text{ Is YearsServices} \leq 6 \text{ ?}\}$$
$$\{ \text{ Is Age} \leq 31 \text{ ?}\}$$

Those binary questions are also two binary functions from $\mathcal{A}$ in $\{true, false\}$ noted :

$$Q_1\text{(a)=[YearsServices(a)} \leq 6 \text{]}$$
$$Q_2\text{(a)=[Age(a)} \leq 31 \text{]}$$

Those functions are defined according to the assignment rules given section 3.2.

According to those binary functions, production rules can be defined for each cluster. For instance the production rule of cluster 1 is :

If [YearsServices(a) $\leq$ 6]=true and [Age(a) $\leq$ 31]=true
then a $\in$ Cluster 1

Each cluster is also an assertion (representing a set of assertions) characterized by a conjunction of properties. For instance, Cluster 1 is described by :

$$\text{[YearsServices} \leq 6 \text{]} \wedge \text{[Age} \leq 31 \text{]}$$

References

[1] Breiman, L., J.H. Friedman, R.A. Olshen, C.J. Stone : Classification and regression Trees. Wadsworth & Brooks/Cole Advanced books & software (1984)

[2] Brito, P. : Use of Pyramids in Symbolic Data Analysis. New Approaches in Classification and Data Analysis. Springer-Verlag (1994) 378–386

[3] Chavent, M. : Analyse des Données Symboliques. Une méthose divisive de classification. PhD Thesis Université Paris-IX Dauphine, France (1997).

[4] Ciampi, A. : Classification and Discrimination : the RECPAM Approach. proc. of COMPSTAT'94 (1994) 129-147

[5] Diday, E. : Des objets de l'Analyse des Données à ceux de l'Analyse des connaissances. Induction symbolique et numérique à partir des données, Y.Kodratoff and E.Diday Eds, Cepadues (1991)

[6] Edwards, A.W.F. and Cavalli-Sforza, L.L. : A method for cluster analysis. Biometrics **21** (1965) 362-375

[7] Gowda, K.C. and Ravi, T.V. : Agglomerative clustering of symbolic objects using the concept of both similarity and dissimilarity. Pattern Recognition Letters **16** (1995) 647-652

[8] Hwang H.Y. and Fu, W.C. : Efficient Algorithms for Attribute-Oriented Induction. First International Conference on Knowledge Discovery and Data Mining, Montreal, Quebec, Canada (1995) 168–173

[9] Ichino, M. and H. Yaguchi : General minkowsky metrics for mixed feature type data analysis. IEEE Transaction on System, Man and Cybernetics **24** (1994) 698-708

[10] Michalski, R. : A theory and methodolody of inductive learning. Machine Learning, an artificial intelligence approach **1** (1986) 83–134

[11] Sneath, P.H. and R.R. Sokal : Numerical Taxonomy. Freeman and company, San Francisco (1973)

[12] Stéphan, V. : Construction d'objets symboliques par synthèse des résultats de requêtes SQL PhD Thesis Université Paris-IX Dauphine, France (1998).

[13] Vitter J.S. :. Random Sampling with a reservoir. ACM Transactions on Mathematical Software **11** (1985) 37–57

[14] Williams, W.T. and J.M. Lambert : Multivariate methods in plant ecology I: association analysis in plant communities. J. Ecology **50** (1959) 775-802