Normalizing Constrained Symbolic Data for Clustering

Marc Csernel*, F.A.T. de Carvalho **

*INRIA - Rocquencourt, Domaine de Voluceau
Rocquencourt - B. P. 105, 78153 le Chesnay Cedex - France
Marc.Csernel@inria.fr

**Centro de Informatica - CIn/UFPE, Av. Prof Luiz Freire,
s/n,Cidade Universitaria, CEP 50.740-540, Recife - PE BRAZIL
fatc@cin.ufpe.br

Abstract. Clustering is one of the most common operation in data analysis while constrained is not so common. We present here a clustering method in the framework of Symbolic Data Analysis (S.D.A) which allows to cluster Symbolic Data. Such data can be constrained relations between the variables, expressed by rules which express the domain knowledge. But such rules can induce a combinatorial increase of the computation time according to the number of rules. We present in this paper a way to cluster such data in a quadratic time. This method is based first on the decomposition of the data according to the rules, then we can apply to the data a clustering algorithm based on dissimilarities.

1 Introduction.

The aim of cluster analysis is to organize a set of items into clusters such that items within a given cluster have a high degree of similarity, whereas items belonging to different clusters have a high degree of dissimilarity. Cluster analysis can be divided into hierarchical and partitioning methods (Gordon (1999), Everitt (2001)). While hierarchical methods build hierarchies, i.e., a nested sequence of partitions of the input data, partitioning methods try to obtain a partition of the input data into a fixed number of clusters, usually by optimizing a function.

This paper addresses the partitioning of constrained symbolic data into a predefined number of clusters. Symbolic data allows to manage some domain knowledge, provided by relations between the variables. These relations are expressed by rules expressing knowledge among the data. A good description of symbolic data can be found in Bock and Diday (2000).

Symbolic data are expressed by symbolic variables which are defined according to the type of their domain. In this paper we will focus on set-valued variables which take their values in a set of nominal categories and a list-valued variable which take as values list of ordered categories.

Table 1 displays an example of two symbolic descriptions called d_1 and d_2, which are described by three set-valued variables and one list-valued variable (Thorax size). These data can be constrained by the dependencies rules r_1, r_2.

	Wings	Wings_color	Thorax_color	Thorax_size
d_1	{absent,present}	{red,blue}	{blue,yellow}	{small,big}
d_2	{absent,present}	{red,green}	{blue,red}	{small}

TAB. 1 – *Example of Symbolic Description*

$$Wings \in \{absent\} \Longrightarrow Wings\_color = N.A. \qquad (r_1)$$
$$Wings\_color \in \{red\} \Longrightarrow Thorax\_color \in \{blue\} \qquad (r_2)$$

Symbolic Data Analysis (SDA) has provided different clustering tools that differ according to the type of data and the type of clustering considered. We just give references:

Concerning Hierarchical Clustering:

Ichino and Yaguchi (1994) define generalized Minkowski metrics for mixed feature variables and present dendrograms obtained from the application of standard linkage methods for data sets containing numeric and symbolic feature values. Gowda and Ravi (1995b) and Gowda and Ravi (1995a) have presented, respectively, divisive and agglomerative algorithms for symbolic data based on the combined use of similarity and dissimilarity measures. These proximity (similarity or dissimilarity) measures are defined on the basis of the position, span and content of symbolic objects.

Chavent (1998) has proposed a divisive clustering method for symbolic data which provides simultaneously a hierarchy of the data and a monothetic characterization of each cluster.

Concerning partitioning algorithms:

Ralambondrainy (1995) extended the classical k-means clustering method and complemented this method with a characterization algorithm which provides a conceptual interpretation of the clusters.

Bock (2002) proposed several clustering algorithms for symbolic data described by interval variables, and presented a sequential clustering and updating strategy for constructing a Self-Organizing Map (SOM) to visualize interval data. De Carvalho et al. (2006) proposed a dynamic clustering algorithm using an adequacy criterion based on adaptive Hausdorff distances. De Carvalho et al. (2006) introduced a dynamic clustering algorithm using a L_2 distance emphasizing the standardization problem and presenting tools for cluster and partition interpretation.

None of these methods is able to take constraints into account, because it leads usually to a combinatorial growth of the computational time according to the number of rules (see § 4.2.2).

The main contribution of this paper is to present an approach which allows to cluster constrained symbolic data according to the following steps:

– Decompose the data according to the rules following the Normal Symbolic form.
– Compute a dissimilarity between the data to build a dissimilarity matrix.
– Apply a dynamic clustering algorithm on the dissimilarity data matrix.

We use a method, inspired by the third normal form (Codd (1971)) used in database, called Normal Symbolic Form due to Csernel and de Carvalho (1999) to compute a dissimilarity in presence of constraints in a quadratic time, whatever big the number of rules is. Using a dissimilarity function allows to cluster any kind of items provided that a dissimilarity table can

be built upon them. In this paper we propose to cluster the data by applying a dynamic cluster algorithm (see section 3) directly to a dissimilarity table (Lechevallier (1974)).

2 Constrained Symbolic Data

A number of different definitions of symbolic descriptions is available in the literature. Here, we refer for the main part to those given by Bock and Diday (2000): symbolic descriptions can be represented by a vector of feature values $\mathbf{s} = (X^1, \ldots, X^j, \ldots, X^p)$, where a feature value X^j $(j = 1, \ldots, p)$, is a subset of the domain D^j of a variable y^j.

Given a set of symbolic variables $\{y^1, \ldots, y^p\}$, a *symbolic description* is a conjunction of events pertaining to a particular object: $s = [y^1 \in X^1] \wedge \ldots \wedge [y^p \in X^p]$. For example, $s = [color \in \{green, red\}] \wedge [height \in [160, 190]]$ is a symbolic description having the following properties:

a) color is either green or red.

b) height ranges between 160 and 190.

An individual description can be represented by a vector of feature values $\mathbf{z} = (z^1, \ldots, z^p)$ where a feature value z^j $(j = 1, \ldots, p)$ can be a single nominal categorical value, or a single categorical values or a single quantitative value. Given a set of individual descriptions $E = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$, where $\mathbf{z}_i = (z_1^1, \ldots, z_i^p)$, the *extension* of the symbolic description $\mathbf{s}$ is defined as $ext(\mathbf{s}) = \{\mathbf{z}_i \in E : z_i^j \in X^j, i = 1, \ldots, n\ , j = 1, \ldots, p\}$. The *virtual extension* of the symbolic description $\mathbf{s}$ is defined as $vext(\mathbf{s}) = \{\mathbf{z} = (z^1, \ldots, z^p) : (z^1, \ldots, z^p) \in X^1 \times \ldots \times X^p\}$. Of course, the following relation holds: $ext(\mathbf{s}) \subseteq vext(\mathbf{s})$.

Example: With the following individual descriptions

	color	size
Beatle1	Blue	small
Beatle2	Red	medium

and the following symbolic description.

	color	size
specie1	{Blue,Red}	{small,medium}
specie2	{Yellow,Green}	{medium,big}

The extension of *specie1* is $ext(specie1) = \{Beatle1, Beatle2\}$, its virtual extension contains four different elements: *{Blue,small},{Blue,medium},{Red,small}, {Red,medium}*.

2.1 Constraints on Symbolic Descriptions

Symbolic descriptions can be constrained by dependencies between couples of variables expressed by rules. Such rules can be considered as constraints among the description space, they produce some "holes" in it because they forbid some individual descriptions to be considered as a part of the virtual extension of a symbolic description. Each dependence is represented by a rule. We will call premise variable and conclusion variable the variables associated, respectively, with the premise and the conclusion of each rule. We take into account two kinds of dependencies: hierarchical and logical.

Let $\mathcal{C}$ be a set of categories. In the following $\mathcal{P}^*(\mathcal{C})$ will denote the power set of $\mathcal{C}$ without the empty set. Let y_1 and y_2 be two categorical set-valued variables whose domains are respectively $\mathcal{C}_1$ and $\mathcal{C}_2$.

A hierarchical dependence between the variables y_1 and y_2 is expressed by the following kind of rule called hierarchical rule:

$$\text{if } [y_1 \in \mathcal{P}^*(\mathcal{C}_1)] \Longrightarrow [y_2 = N.A.]$$

where the term N.A. means *not applicable* hence the value of variable does not exist. With this kind of dependence, we sometimes speak of mother-daughter variables. Few works have considered the N.A. semantic, but we can mention the paper of Lerat and Lipski (1986) which is mostly in the field of databases. In this paper, we will deal mostly with hierarchical rules. The rule r_1 described hereafter shows an example of such a rule:

$$\text{if } [Wings \in \{absent\}] \Longrightarrow [Wings\_color = N.A.] \qquad (r_1).$$

A **logical dependence** between the variables y_1 and y_2 is expressed by the following kind of rule:

$$\text{if } [y_1 \in \mathcal{P}^*(C_1)] \Longrightarrow [y_2 \in \mathcal{P}^*(C_2)].$$

The rule r_2 described hereafter shows an example of such a rule:

$$\text{if } [Wings\_color \in \{red\}] \Longrightarrow [Thorax\_color \in \{blue\}] \ \ (r_2)$$

Both of these rules reduce the number of individual descriptions belonging to the extension of a symbolic description, but the first kind of rule reduces the number of dimensions of a symbolic description, whereas the second does not. It has been shown in De Carvalho (1998) that computation using rules leads to exponential computation time depending on the number of rules. To avoid this combinatorial explosion of computation time we introduced the Normal Symbolic Form (N.S.F.) (Csernel and de Carvalho (1999), Csernel and de Carvalho (2002), Csernel and de Carvalho (1998)).

2.2 The dependence graph induced by the rules

The different constraints allow us to build a directed graph where the nodes are the variables and the edges are representing the rules. Each edge goes from the premise variable to the conclusion variable. This graph can be not connected. Because NFS induces a decomposition of the description space leaded by the premise variables, we can not deal generally with variables which are conclusion of two different premise variables. Then, it results it is required that the graph induced by the rules forms a tree or a set of trees.

Example: If we consider the three following rules:

- if [Hand $\in \{Absent\}] \Longrightarrow$ [Hand_size = N.A.]
- if [Hand $\in \{Absent\}] \Longrightarrow$ [Finger = N.A.]
- if [Finger $\in \{Absent\}] \Longrightarrow$ [Finger_Size = N.A.]

they induce the following dependence tree between the variables (see Figure 1).

Remark: if two different rules are related to the same variables, they will produce one edge only in the dependence graph.

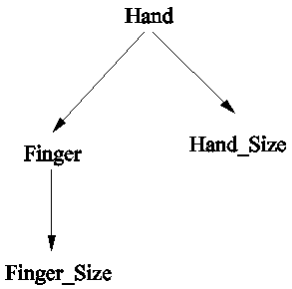

FIG. 1 – *The dependence tree between the variables*

2.3 The mutual interaction of the rules

The different rules in a knowledge base can interact mutually. Hereafter we will focus on one type of interaction due to the kind of inheritance induced by hierarchical dependencies, which leads to the following consequence: if we have the two following rules:

if [Hand $\in \{Absent\}$] $\Longrightarrow$ [Finger = N.A.]

if [Finger $\in \{Absent\}$] $\Longrightarrow$ [Finger_size = N.A.]

then, it's an evidence that:

if Hand [$\in \{Absent\}$] $\Longrightarrow$ [Finger_Size = N.A.]

This constitutes the *Not Applicable propagation*: if a variable is N.A., all its descendant within the dependence graph will also be N.A.

2.4 The notion of Coherence

The combinatorial explosion of computation time induced by the presence of domain knowledge is closely linked to the notion of coherence. Provided a domain knowledge expressed by a set of rules we say that

- An individual description is *coherent* if it respects the rules;
- A symbolic description is *not coherent* or *incoherent* if all the individuals belonging to its virtual extension are not *coherent*;
- The *coherent part* of a symbolic description is its (non empty) virtual extension calculated when taking into account the rules;
- A symbolic description is *fully coherent* if its virtual extension calculated taking the rules into account is equal to its virtual extension calculated without taking the rules into account .
- We say that a symbolic description is *coherent*, if it is neither *incoherent* nor *fully coherent*;

For example, if we have the following rule:

$$\text{if } [Wings \in \{Absent\}] \Longrightarrow [Wings\_color = N.A.]\ (r_1)$$

and the following symbolic descriptions:

description	Wings	Wings_color
d_1	{Absent}	{Blue, Red, Yellow}
d_2	{Absent, Present}	{Blue, Red, Yellow}
d_3	{Present}	{Blue, Red, Yellow}
d_4	{Absent}	{N.A.}

- d_1 is **not coherent** : when $Wing = Absent \Longrightarrow Wings\_color$ should be N.A., according to r_1 and this not the case. The virtual extension of d_1 without taking the rule into account is: $vext(d_1) = \{(absent, blue), (absent, red), (absent, yellow)\}$. None of these individual descriptions belonging to $vext(d_1)$ are coherent;
- d_2 is **coherent** because $Wings = \{Absent, Present\}$ and $Wings\_color$ has a set-value. According to the rule, this set-value is meaningful only when $Wings = Present$, but not when $Wings = Absent$. The virtual extension of d_2 without taking the rule into account is

$$vext(d_2) = \{(absent, blue), (absent, red), (absent, yellow),$$

$$(present, blue), (present, red), (present, yellow)\}$$

  whereas the virtual extension of d_2 taking the rule into account is

$$vext(d_2) = \{(absent), (present, blue), (present, red), (present, yellow)\}$$

  As a consequence, d_2 is neither incoherent nor fully coherent: it is coherent.
- d_3 is **fully coherent** $Wing = \{Present\}$ and $Wing\_color$ has a set-value, the rule does not apply. As a consequence, its virtual extension taking into account the rule is equal to its virtual extension without taking into account the rule:
$$\{(present, blue), (present, red), (present, yellow)\}.$$
- d_4 is **fully coherent** $Wing = \{Absent\}$ and $Wing\_color$ has no set-value, (N.A.) the rule applies. As a consequence, its virtual extension taking into account the rule is equal to its virtual extension without taking into account the rule $\{(absent)\}$.

It is because computations done on constrained symbolic objects need to be done only on the coherent part of a description that the combinatorial explosion of the computation time occurs:

- The different algorithms need to know precisely which part of the symbolic description is coherent in order to make their computations only with this coherent part.
- The idea underlying the N.S.F. is to represent only the fully coherent part of a symbolic description, in order to avoid the previously mentioned calculation.

If we can reach such a goal at a reasonable cost, then all the computation can be done using background knowledge with a reasonable time (as if no background knowledge was used).

3 Dynamic Clustering Algorithms

Partitioning dynamical cluster algorithms (Diday (1973)) are iterative two-step relocation algorithms involving the construction of clusters at each iteration and the identification of a suitable representation or prototype (means, axes, probability laws, groups of elements, etc.) for each cluster by locally optimizing an adequacy criterion between the clusters and their

corresponding representatives (Diday and Simon (1976)). An allocation step is first performed to assign individuals to clusters according to their proximity to the prototypes. This is followed by a representation step where the prototypes are updated according to the assignment of the individuals in the allocation step, until the convergence of the algorithm, when the adequacy criterion reaches a stationary value.

Let Ω be a set of n items indexed by i and described by p variables indexed by j. Each item i is represented by a vector of feature values $\mathbf{x}_i = (x_i^1, \ldots, x_i^p)$. Throughout this paper, we consider the problem of clustering Ω into K disjoint clusters $C_1, ..., C_K$ such that the resulting partition $P = (C_1, ..., C_K)$ is optimum with respect to a given clustering criteria.

By adopting the framework of dynamic clustering (Diday and Simon (1976)), we represent each cluster $C_k \in P$ by a prototype $\mathbf{y}_k$, which is also a vector of feature values. *Note that* $\mathbf{y}_k$ *could be not a member of* Ω. We measure the quality of this cluster by the sum of the dissimilarities $d(\mathbf{x}_i, \mathbf{y}_k)$ between items $i \in C_k$ and the prototype $\mathbf{y}_k$. This measure of quality $\sum_{i \in C_k} d(\mathbf{x}_i, \mathbf{y}_k)$ is called the adequacy criterion of the cluster C_k. The classification problem is to find a partition P and a set L of K prototypes that minimize the following clustering criterion:

$$\Delta(P, L) = \sum_{i=1}^{K} \sum_{i \in C_K} d(\mathbf{x}_i, \mathbf{y}_k) \tag{1}$$

over all partitions $P = (C_1, ..., C_K)$ of Ω and all choices of set $L = (\mathbf{y}_1, ...\mathbf{y}_K)$ of cluster prototypes.

In this context, the dynamic clustering algorithm performs iteratively both a *representation step* and an *allocation step*:

a) Representation step (the partition P is fixed).

Find L that minimises $\Delta(P, \bullet)$ is equivalent to find for $k \in \{1, .., K\}$, the prototype $\mathbf{y}_k$ that minimises the adequacy criterion $\sum_{i \in C_k} d(\mathbf{x}_i, \mathbf{y}_k)$.

b) Allocation step (the set of prototypes L is fixed).

Finding P that minimises $\Delta(\bullet, L)$ is equivalent to find for $k \in \{1, .., K\}$, the cluster $C_k = \{i \in \Omega \mid d(\mathbf{x}_i, \mathbf{y}_k) \leq d(\mathbf{x}_i, \mathbf{y}_m)\ , \forall m \in \{1, .., K\}\}$

Once these two steps properly defined, the partitioning criterion (1) decreases at each iteration and the algorithm converges to a stationary value of this criterion under the two following conditions:

i) Unicity of the "cluster affectation" choice for each item of Ω.

ii) Unicity of the prototype $\mathbf{y}_k$ choice that minimizes the adequacy criterion: $\sum_{i \in C_k} d(\mathbf{x}_i, \mathbf{y}_k)$.

The dynamic clustering algorithm can also be performed using a distances between the items, instead of using the items themselves. This introduces two major changes:

- The prototype is no more a "virtual" item, it must be a real one.
- We don't use any more the data matrix, but a dissimilarity matrix.

3.1 Clustering algorithm on dissimilarity tables

We use a clustering criterion which is based on the sum of dissimilarities between the individuals belonging to the same cluster, and try to minimise this clustering criterion by a

suitable choice of the classes. The aim of the clustering process is to be able to group the objects of a set Ω into k homogeneous clusters on the basis of a dissimilarity table. The proposed approach is an application of the dynamical clustering method to the case of a dissimilarity table (Lechevallier, 1974). The algorithm follows the main principles of the method.

The Algorithm:

- *Initialisation*:
  Let L_o= $\{\mathbf{y}_1^{(o)}, \ldots, \mathbf{y}_K^{(o)}\}$ be the initial prototypes, defined as randomly chosen objects of the set $\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$
- *Allocation step t*:
  An object $\mathbf{x}_i \in \Omega$ is assign to the class $C_k^{(t)}$ $(k = 1, \ldots, K)$, if $d(\mathbf{x}_i, \mathbf{y}_k^{(t-1)})$ is minimum.
- *Representation step t*:
  For $k = 1, \ldots, K$, find a prototype $\mathbf{y}_k^{(t)} \in \Omega$ representing class $C_k^{(t)} \in P^{(t)}$ which minimises $\sum_{\mathbf{x} \in C_k^{(t)}} d(\mathbf{y}_k^{(t)}, \mathbf{x})$.
- *Stopping rule or stability*:
  If $P^{(t+1)} = P^{(t)}$ **then** STOP **else** GO TO *allocation step*

4 A Dissimilarity to compare constrained symbolic data

Many proximity indices for symbolic data have been introduced in the literature, we will describe here some of them. Gowda and Diday (1992) introduced, similarity functions with position, span and content components. Ichino and Yaguchi (1994) presented the generalised Minkowski metrics for mixed feature variables. Similarity and dissimilarity measures between symbolic data constrained by dependency rules between feature values can be found in De Carvalho (1994a). Chavent and Lechevallier (2002) proposed a dynamic clustering algorithm for symbolic interval data where the class representatives are defined by a criterion based on a modified Hausdorff distance. De Carvalho et al. (2006) presented a dynamic clustering algorithm for interval data based on suitable Euclidean distances.

We can see in the literature that the usual way to compute dissimilarities between symbolic descriptions is done comparing the values of each description, variable after variable. The results of these comparisons are then agglomerated according to two different models:

- According to a multiplicative model: Then the rules are naturally taken into account, because all variables are considered together.
- According to an additive model: In this case we can only take the rules into account according to a ponderation which can be related to the rules and to the value of the description on a specific variable.

To define a dissimilarity, we need a measure for each symbolic description: *the description potential*, we also need some operations on symbolic description in order to combine them in a proper way.

4.0.1 Usual operations with symbolic objects

We will recall here two different operations that we can use for dissimilarity computation dealing with symbolic description. For the dissimilarity formula (2) described hereafter, we

only need the join operator (note that the join operator here is quite different from the one used in data base technology).

Let $a = \bigwedge_{i=1}^{p} [y_i \in A_i]$ and $b = \bigwedge_{i=1}^{p} [y_i \in B_i]$ two symbolic descriptions.

– The conjunction is defined as

$$a \wedge b = \bigwedge_{i=1}^{p} [y_i \in A_i \cap B_i]$$

– The join due to Ichino and Yaguchi (1994) between these two symbolic descriptions is defined as

$$a \oplus b = \bigwedge_{i=1}^{p} [y_i \in A_i \oplus B_i]$$

where:

i) if y_i is a symbolic set-valued variable (i.e. A_i and B_i are set of nominal categories);

$$A_i \oplus B_i = A_i \cup B_i$$

ii) if y_i is a symbolic list-valued variable (i.e. $A_i = [low(A_i), up(A_i)]$ and $B_i = [low(B_i), up(B_i)]$ are lists of ordered categories)

$$A_i \oplus B_i = [\min(low(A_i), low(B_i)), \max(up(A_i), up(B_i))]$$

iii) if y_i is an interval-valued variable:
(i.e. $A_i = [low(A_i), up(A_i)]$ and $B_i = [low(B_i), up(B_i)]$ are intervals of $\Re$)

$$A_i \oplus B_i = [\min(low(A_i), low(B_i)), \max(up(A_i), up(B_i))]$$

The figure 2 illustrates these operations for symbolic interval-valued variables.

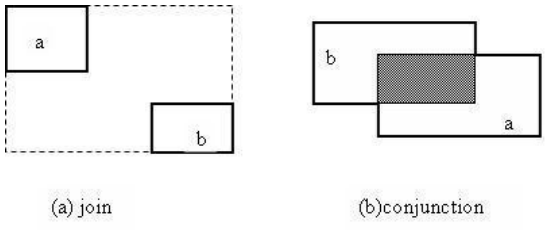

FIG. 2 – *Some operations between symbolic descriptions*

4.1 The proximity function

For our proximity function, we will need the join operator previously defined. We will also use a measure called description potential.

If $d = \wedge_{i=1}^{p} C_i^1$ is a symbolic description, we will denote $\pi(d)$ the description potential of d. When no rules are envolved, the description potential is a measure defined on the Cartesian

product $C_1 \times \ldots \times C_p$, where $C_i \subseteq D_i$, D_i being the domain of the symbolic variable y_i. We will describe this measure more in details in the next section. The dissimilarity function we will use:

$$\delta(a,b) = \frac{1}{2}\left[\pi(a \oplus b) - \pi(a) + \pi(a \oplus b) - \pi(b)\right] \tag{2}$$

is inspired from Ichino and Yaguchi (1994) and has been introduced in De Carvalho (1998) .

Examining more precisely formula (2) we see that this distance consists of two parts, the first part corresponds to the description of a opposed to the description of $a \oplus b$, the second part to the description of b opposed to the description of $a \oplus b$.

We can see on the Figure 3 a diagram corresponding to formula (2). The dissimilarity value corresponds to the doted part of the figure.

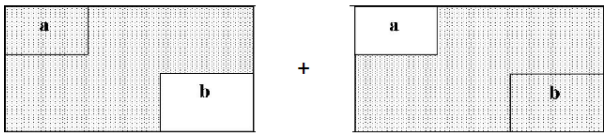

FIG. 3 – *The diagram corresponding to the distance function*

4.2 Computation of Description Potential

The Description Potential (De Carvalho (1998)) is a frequently used measure when dealing with symbolic descriptions, specially in the case of dissimilarity computation.

Formal definition: Let $d = \wedge_{i=1}^{p}[y_i \in C_i]$ be a symbolic description where C_i is a subset of the domain (D_i) of the variable y_i. We denote as $\pi(d)$ the description potential of d. It is a measure defined on the coherent part of d (i.e., the virtual extension of d calculated taking into account the rules). In the following we will describe more precisely how to compute the description potential.

4.2.1 Computation of Description Potential without rules

If d is a symbolic description

$$\pi\left(d\right) = \prod_{i=1}^{p} \mu\left(C_i\right) \tag{3}$$

where

- if y_i is a symbolic set-valued variable (i.e. C_i is a set of nominal categories), $\mu\left(C_i\right)$ is the cardinal of C_i;
- if y_i is a symbolic list-valued variable (i.e. $C_i = (low(C_i), up(C_i))$ is a list of ordered categories), $\mu\left(C_i\right) = (rank(up(C_i)) - rank(low(C_i))) + 1$ where rank is a function which gives the ranking of an ordered category in C_i;

– if y_i is an interval-valued variable (i.e. $C_i = [low(C_i), up(C_i)]$ is an interval of $\Re$), $\mu(C_i)$ is the value of the difference between the upper bound and the lower bound of the interval C_i.

The computation is purely quadratic, and for each symbolic description the computation time is clearly linearly dependent from the number of variables.

4.2.2 Computation of Description Potential with rules

As the "holes" generated by the rules in the description space can intersect, the computation becomes quickly combinatorial according to the number of holes which can intersect, i.e. the number of rules. The "holes" induced by the different rules must be dropped out of the volume, but their intersection two by two must be put back.

Let $d = \wedge_{i=1}^{p}[y_i \in C_i]$ be a symbolic description constrained by a set of rules $\{r_1, \ldots, r_t\}$, each rule expressing a dependency among the variables, then $\pi(d \,|\, \{r_1, \ldots, r_t\})$ expresses the value of the potential of d in presence of the set of rules $\{r_1, \ldots, r_t\}$. It can be proven that:

$$\pi(d \,|\, \{r_1, \ldots, r_t\}) = \prod_{i=1}^{p} \mu(C_i) - \sum_{j=1}^{t} \pi(d \wedge \neg r_j) + \sum_{j<k} \pi((d \wedge \neg r_j) \wedge \neg r_k) + \cdots + (-1)^{t+1} \pi((d \wedge \neg r_1) \wedge \neg r_2) \wedge \cdots \wedge \neg r_t) \quad (4)$$

The complexity of the formula (4) due to De Carvalho (1994b) is exponential according to the number of rules, and linear according to the number of variables. One can remark that this formula is very similar to Poincarre's formula.

Example: Let

$d = [y_1 = \{a1, a2\}] \wedge [y_2 = \{b1, b2\}] \wedge [y_3 = \{c1, c2\}] \wedge [y_4 = \{d1, d2\}]$

be a symbolic description where all the symbolic variables are set-valued. Without rules the computation of the description potential is quite straightforward:

$$\pi(d) = 2 \times 2 \times 2 \times 2 = 16$$

However, if we have the two following rules:

$$\text{if } [y_1 \in \{a_1\}] \Longrightarrow [y_2 \in \{b_1\}] \quad (r_4)$$
$$\text{if } [y_3 \in \{c_1\}] \Longrightarrow [y_4 \in \{d1\}] \quad (r_5)$$

Then we must consider all the individuals belonging to the virtual extension of d to verify if they fit the rules.

Table 2 shows the virtual extension of d. In this table, each line belonging to one half of the array represents an individual description which is numbered. The "coherence" column contains a Y or a N according to the fact that the individual is coherent or not. If the individual is not coherent, the number of the rules by which the incoherence occurs is indicated within parenthesis.

All lines with an N contribute to the first term of the formula (4): $(-\sum_{j=1}^{t} \pi(d \wedge \neg r_j))$.

Line 6 which has N(r_3, r_4) in the coherence column is also corresponding to the second term of the formula (4): $(+\sum_{j<k} \pi((d \wedge \neg r_j) \wedge \neg r_k))$.

N^o	description (individual)	coherence	N^o	description (individual)	coherence
1	(a1, b1, c1, d1)	Y	9	(a2, b1, c1, d1)	Y
2	(a1, b1, c1, d2)	N(r_4)	10	(a2, b1, c1, d2)	N(r_4)
3	(a1, b1, c2, d1)	Y	11	(a2, b1, c2, d1)	Y
4	(a1, b1, c2, d2)	Y	12	(a2, b1, c2, d2)	Y
5	(a1, b2, c1, d1)	N (r_3)	13	(a2, b2, c1, d1)	Y
6	(a1, b2, c1, d2)	N(r_3,r_4)	14	(a2, b2, c1, d2)	N(r_4)
7	(a1, b2, c2, d2)	N(r_3)	15	(a2, b2, c2, d1)	Y
8	(a1, b2, c2, d2)	N (r_3)	16	(a2, b2, c2, d2)	Y

TAB. 2 – *Virtual extension of* d

The value of the potential considering the rules correspond to the number of lines marked by a Y. The value of the potential is 9 instead of 16 without rule.

As, in this example, we get only two rules, we need to consider the two first terms of the formula (4) only. As there is no additivity among the different elements, we must drop out the volume according to each rule. Then put back the volume corresponding to their intersection.

It's to avoid this kind of problem when we are dealing with a dissimilarity computation, that we were induce to introduce the Normal Symbolic Form. The main idea is to split the description space in such a way that we represent only the fully coherent part of the symbolic description. As in the fully coherent description we do not have to check any more if the rules apply or not, the computation is rather quicker and easier.

5 Normal Symbolic Form

In order to provide a better explanation on what the Normal Symbolic Form (N.S.F.) is, we will start our explanation by the following example:

	Wings	Wings_color	Thorax_color	Thorax_size
d_1	{absent,present}	{red,blue}	{blue,yellow}	{big,small}
d_2	{absent,present}	{red,green}	{blue,red}	{small}

TAB. 3 – *Original table.*

In the symbolic data table presented above there are two symbolic descriptions d_1, d_2, and three categorical set-valued variables. The values are constrained by two rules previously seen: a hierarchical one denoted r_1 and a logical one denoted r_2 :

$$[Wings \in \{absent\}] \Longrightarrow [Wings\_color = N.A.] \qquad (r_1)$$
$$[Wings\_color \in \{red\}] \Longrightarrow [Thorax\_color \in \{blue\}] \qquad (r_2)$$

The result of the N.S.F. transformation can be seen in the tables of the figure 4. In these tables the upper left corner contains the table name. A new kind of column appears where each values is a number referring to the corresponding line in the table having the same name as the column, they correspond to *reference variables*.

The first table (with no name) is called the *Main Table*, it refers to the original table (here Table (3)). The other tables are called secondary tables, each of them has a name corresponding to the premise variable of the rule which induced it.

In each secondary table, a double line separates the lines where the first variable verifies the premise, from the lines where the first variable does not verify it. The lines in bold characters correspond to the description of d_1, the table names are in italic.

Each line of a secondary table represents a subset of the coherent part of a symbolic description in the original table. Furthermore, each symbolic description is associated with a subset of lines which constitutes a partition of the coherent part of the symbolic description. Then, only the coherent part of a description is represented under N.S.F.. For example in Color_T table, the number lines 1,2,4, forms a partition of the coherent part of d_1.

	Wings_T	Thorax_size
d_1	**{ 1, 3}**	**{big,small}**
d_2	{2,4}	{small}

Main Table

Wings_T	Wings	**Color_T**
1	**absent**	**4**
2	absent	5
3	**present**	**{ 1, 2 }**
4	present	{ 1, 3 }

Wing_T Table

Color_T	Wings_color	Thorax_color
1	**{ red }**	**{blue }**
2	**{ blue }**	**{ blue, yellow }**
3	{ green }	{blue, red }
4	**N.A.**	**{ blue, yellow }**
5	N.A.	{ blue, red }

Color_T Table

FIG. 4 – *Decomposition of table 3 according to N.S.F.*

We have now three tables instead of a single one, but only the valid parts of the descriptions are represented: now, the tables include the rules r_1 and r_2.

We can remark a growth of the space needed to describe this symbolic descriptions. In the Color_T Table we need 5 lines to describe 2 items. In the original table only two lines where needed. In fact, when there are more items, instead of a memory growth we obtain a memory reduction due to factorization. A complete discussion about this problem can be found in Csernel and de Carvalho (2002).

5.1 Formal Definition

We shall give in this section a more formal definition of the N.S.F. We say that a symbolic data table is under N.S.F. if it fulfills the following conditions.

- **First N.S.F. Condition**: No dependencies exist between variables belonging to the same table, but between the first variable and the others.
- **Second N.S.F. Condition**: For one symbolic description, all the values of a premise variable must lead to the same conclusion.

Most of the time, in order to fulfill the N.S.F. conditions a symbolic array needs to be decomposed, as a relation needs to be decomposed to fulfill Codd's normal forms (Codd (1971)).

Concerning the first N.S.F. condition, one can remark that:

– The reference to the first variable is only for convenience, any other place should have fit, as long as this place is constant.
– The first condition implies that N.S.F. is fully efficient only when the dependences between the variables induced by the rules form a tree or a set of trees.
– We have to decompose the data into different tables.
– Because of the table decomposition due to this condition, we have to introduce new variables called **reference variables**.

The second N.S.F. condition has one main consequence:

– We have to decompose each individual within a table in two parts:
  1) One part where the premise is verified. In this case all the conclusions appearing in the rules apply.
  2) One part where the premise is not verified. The values corresponding to the conclusion variables stay unchanged.

The different tables will form a unique tree according to the dependencies. Each of the dependence between the variables forms a branch of the table tree. The root of the tables tree is the *Main Table*. To refer from one table to another one, we introduce the reference variables, these new variables introduce a small space overhead.

All the tables, but the *Main Table*, are composed in the following way:

– The first variable is a premise variable, all other variables are conclusion variables;
– In each line the premise variable can lead to an unique conclusion for all conclusion variables;
– If we want to represent different conclusions within a table, we need to represent for each object as much lines as we have conclusions.

The main advantage of the N.S.F. is that after this transformation we do not have to worry about the rules any more, we are quadratic ($O(N^2)$ in case of dissimilarity computation) as if there were no rule to consider.

For example, if we got the following rule:

$$\textit{if Wings\_color} \in \{red\} \Longrightarrow Thorax\_color \in \{blue\}$$

N^0	Wings_color	Thorax_color
1	{ red,blue }	{blue,yellow }

becomes

N^0	Wings_color	Thorax_color
1	{ red }	{blue}
2	{ blue }	{ blue, yellow}

One can notice that the union of these two lines gives the initial line.

6 Computation of dissimilarities under N.S.F.

Compute the potential of a symbolic description under N.S.F. is straightforward, because under N.S.F. only the valid parts of the objects are represented. So once under N.S.F. the potential computation is quadratic.

We proceed in a recursive way. Each line of a secondary table describes a coherent symbolic description, and all the lines contributing to the representation of the same object describe symbolic descriptions which do not intersect (by construction). So one has to sum up the potential described by each line of a secondary table.

	Wings_T	Thorax_size	pot
d_1	{1,3}	{big,small}	10
d_2	{2,4}	{small}	5

Main Table

Wings_T	Wings	Color_T	pot
1	absent	4	2
2	absent	5	2
3	present	{1,2}	3
4	present	{1,3}	3

Wing_T table

Color_T	Wings_color	Thorax_color	pot
1	red	{blue }	1
2	blue	{ blue, yellow }	2
3	green	{blue, red }	2
4	N.A.	{ blue, yellow }	2
5	N.A.	{ blue, red }	2

Color_T table

FIG. 5 – *Computation of the description potential using N.S.F.*

On the example described in the figure 5, the potential of each line of the secondary table 2 (Color_T table) has to be computed first, then the lines of the secondary table 1 (Wing_T table), and at last the potential of each object described in the main table. For example line 3 of the secondary table 1, refers to the lines 1 and 2 of the secondary table 2. The potential is the sum of the potential described by these two lines: 1+ 2 = 3. The description potential of d_1 is obtained by multiplying the sum of the potentials of lines 1 and 3 of the secondary table 1 (2 + 3) by the potential due to the variable Thorax_size (2) giving the result 10. In the same way, the description potential of d_2 is obtained giving the result 5.

Note that without rules the results would be $\pi(d_1) = 2 \times 2 \times 2 \times 2 = 16$ and $\pi(d_2) = 2 \times 2 \times 2 \times 1 = 8$ (see figure 3 for the original data).

The computation of the description potential is done recursively, following the tables tree. To compute a dissimilarity between symbolic descriptions, such as the dissimilarity described in formula (2), we will need not only to compute the potential of one symbolic description, but also to compute the potential resulting from the join operation. The results of such operation have been studied by Csernel and de Carvalho (1999), and they demonstrated that the N.S.F. is stable for this operation, i.e. the result of a join operation between two symbolic descriptions under N.S.F. is still under N.S.F. However, this operation, which can create new symbolic descriptions, needs the application of some of the original rules to be achieved correctly. The complexity will remain identical; it has been demonstrated in Csernel and de Carvalho (1998) that the computational time is always linear according to the number of variables.

Figure 6 illustrates why, in some cases, the application of a rule on the result of the operation could be needed again.

The description space is divided in two parts, which on the schema are separated by a black line: in the upper part a rule R applies, and in the lower part the rule does not applies. We have denotes R_A the part of the description space where the rule applies and R_N the part where the rule does not apply. In the part where the rule applies the forbidden zone appears in black. When the join operation is performed between two descriptions d_1 and d_2, if both descriptions

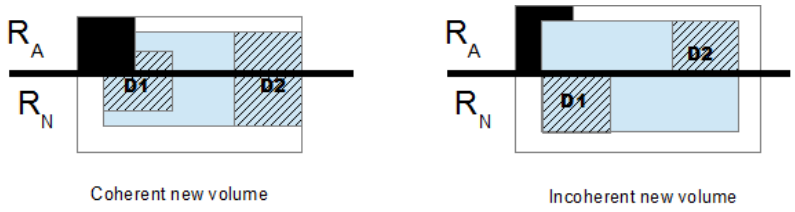

FIG. 6 – *Rule influence on a dissimilarity computation*

have a part in R_A (such as in the left part of the schema) no problem occurs, but we have some problem if one of the description is belongs only to R_A and the other only in R_N (as in the right part of the schema), then the new symbolic description created by the join operation (in pointed grey on the schema) can cover the forbidden area (which represents the rule). To avoid this coverage, the rule must be applied once again, i.e we must compute the coherent part of the description which is a linear in N the number of objects because in each secondary table only one rule is associated. If the description corresponds to the right part of the schema the result is always coherent and no further operation is needed.

If we compute the similarity function using formula 2 (page 10) we must compute the expression $d_1 \oplus d_2$. We start with the *Main Table*, and for each secondary table, as much as possible, we compute apart the potential coming form R and the one coming from R_n and we sum them up. We will denote π_w the potential related to Wing_T table and π_c the potential related to the Color_T table. For each line the first term (2 for the first line) correspond to the value of the description potential computed directly from local variables, the other term correspond to the description potential obtained trough the other secondary tables.

- $\pi(d_1 \oplus d_2) = 2 * (\pi_w(1 \oplus 2) + \pi_w(3 \oplus 4))$;
- $\pi_w(1 \oplus 2) = 1 * (\pi_c(4 \oplus 5)) = 1 * 3 = 3$;
- $\pi_w(3 \oplus 4) = 1 * (\pi_c(1) + \pi_c(2 \oplus 3)) = 1 + 1 * (2 * 3) = 7$;
- $\pi(d_1 \oplus d_2) = 2 * (3 + 7) = 20$.

Note that without the rules the result would be $\pi(d_1 \oplus d_2) = 2 \times 3 \times 3 \times 2 = 36$.

Finally, the dissimilarity between d_1 and d_2, according equation (2) is

$$\delta(d_1, d_2) = \frac{1}{2}((20 - 10) + (20 - 5)) = 12.5$$

Note that without the rules the result would be

$$\delta(d_1, d_2) = \frac{1}{2}((36 - 16) + (36 - 8)) = 24.0$$

Considerations about complexity

At first, we consider the complexity of the N.S.F. transformation. This complexity concerns only the secondary matrices and is in $O(N^2)$, N being the number of objects. Due to factorization,the number N_s of line of a secondary table is generally a lot smaller than N, so the complexity is a lot smaller (see Csernel (1998) for more details).

Concerning space complexity (the size of the secondary table induced by the N.S.F. decomposition), it have been demonstrated in Csernel and de Carvalho (2002) that if N is the number of objects then the number of lines of a secondary table is at most $N + 1$ if it has been

induced by hierarchical rules, and at most $2N$ if it has been induced by logical rules, but that it is generally smaller because of the factorization (as noted previously), and we observed on real examples secondary tables five or ten times smaller than the main tables.

The only drawback induced by computation under N.S.F. is due to the use of the reference variables and the possible additions induced by them.

The complexity of the computation of one distance between two objects is $O(p)$, where p is the number of variables. The complexity of the computation of the dissimilarity matrix is $O(N^2)$. The complexity of the dynamic clustering algorithm, based on dissimilarity table, once the dissimilarity table has been computed, is in the worst case the complexity of a sort ($O(N * Log(N))$, and, in the best case in $0(N)$; it is really smaller than the distance computation, we can repeat the trials without changing the global complexity of the process.

The global complexity for the entire process is therefore in $O(N^2)$. It is mainly due to the distance computation and it behaves as if they were no rule to consider.

7 Experimental Results

To validate the proposed approach, we have conducted several experiments on the following biological symbolic data sets: species of podostemaceae and sub-species of Phlebotominae (genus Lutzomyia) of French Guyana. These experiments have been detailed in De Carvalho et al. (2009), and have two main issues: the execution time was quadratic, and the presence of the rules greatly improve the classification as expressed in the tables 4 and 7.

To compare the clustering results furnished by the clustering algorithm, and the *a priori* we will use the corrected Rand index (CR), as well as error rate of classification ($OERC$)

The corrected Rand (CR) index (Hubert and Arabie (1985)) measures the similarity between an *a priori* partition and a partition furnished by a clustering algorithm. It takes its values within the interval [-1,1] where 1 indicate a perfect agreement between partitions, and values near 0 (or negative values) correspond to cluster agreement found by chance.

The Podostemaceae

The Podostemaceae is a family in the order Malpighiales. It comprises about 50 generae and 250 species of more or less thalloid aquatic herbs. The Podostemaceae data set consists of 12 species of aquatic herbs described by 14 symbolic set-valued variables, 13 symbolic list-valued variables and one nominal variable (Vignes (1991)). There are 9 hierarchical rules. These rules are represented by 6 different connected graphs involving 15 symbolic variables. All the symbolic set-valued and list-valued variables were considered for dissimilarity computation purpose taking or not into account the 9 hierarichical rules, the nominal variable *Genus* (Dalzellia, Ind otristicha, Malaccotristicha, Tristicha and Weddellina) was used as an *a priori* classification.

The dynamic clustering algorithm was applied to the dissimilarity table which was build according to the dissimilarity function (formula 2 page 10) taking or not taking into account the 9 hierarchical rules. The 5-clusters partition obtained with this method applied to dissimilarity tables build with or without taking into account the nine hierarchical rules were compared with the 5-clusters partition known *a priori*. The algorithm best result according to the adequacy criterion was selected. CR and $OERC$ indices were calculated for the best result.

The *a priori* classification is the following:

A-Tristicha(T): 1-Trifaria pulchella 2-Trifaria tlatlayana 3-Australis 12-Trifaria trifaria

B-Indotristicha(I): 4-Ramosissima 5-Tirunelveliana

C-Dalzellia(D): 6-Carinata 7-Diversifolia 8-Sessilis 9-Ceylanica

D-Malaccotristicha(M) : 10-Malayana

E-Weddellina(W) : 11-Squamulosa

Dissimilarity data sets	Without taking into account the rules	Taking into account the rules
Cluster1	1/T	3/T 6/D 7/D 8/D 9/D 10/M
Cluster2	11/W	1/T 2/T
Cluster3	2/T	12/T
Cluster 4	3/T 6/D 7/D 8/D 9/D 10/M 12/T	4/I 5/I
Cluster 5	4/I 5/I	11/W

TAB. 4 – *Clustering Results for the species of podostemaceae*

Table 4 shows clearly that the 5-clusters partitions given by the clustering algorithm applied on the data while taking the rules into account are closer to the 5-classes partition known *a priori* than it is in the 5-clusters partitions obtained by the same algorithms applied without taking rules into account.

The CR and $OERC$ indices corroborate this conclusion: the indices obtained from the results displayed in Table 4 were, respectively, 0.288 and 0.333 without taking the rules into account, and 0.617 and 0.250 when the 9 hierarchical rules were considered.

In conclusion, the algorithm had better performances when it take into consideration the hierarchical rules in the computation of dissimilarities.

The Phlebotominae

The sub-species of Phlebotominae (genus Lutzomyia) data set consists of 70 species of sand flies described by 18 symbolic set-valued variables, 33 symbolic list-valued variables and one nominal variable (Vignes (1991)). There are 4 hierarchical rules. These rules are represented by 2 different connected graphs involving 6 symbolic variables. All the symbolic set-valued and list-valued variables were considered for the dissimilarity computation with and without taking into account the 4 hierarchical rules.

The Phlebotominae subfamily includes numerous genera of blood-feeding flies, including the primary vectors of leishmaniasis, sandfly fever. In the New World, leishmaniasis is spread by sand flies of the genus Lutzomyia, which are common inhabitants of caves, where they feed on bats. The nominal variable *Groups of genus Lutzomyia* (1-Aragoi, 2-Baityi + Cayennensis + Oswaldoi, 3-Dreisbachi + Microps, 4-Evandromyia, 5-Lutzomyia, 6-Migonei + Saulensis, 7-Nyssomyia, 8-Pilosa + Trichopygomyia, 9-Pintomyia + Pressatia, 10-Psathyromyia, 11-Psychodopygus, 12-Trichophoromyia, 13-Verrucarum, 14-Viannamyia) was used as an *a priori* classification.

The clustering algorithm were applied to the dissimilarity table computed with the formula 2 with or without taking the 4 hierarchical rules into account.

The 14-clusters partitions obtained with this method by applied to the dissimilarity data sets with and without the 4 hierarchical rules were compared and with the 14-classes partition known *a priori*.

The *a priori* classification is :

A-Aragoi: 5-aragaoi 7-barrettoi barrettoi 11-brasiliensis 32-inflata

B-Baityi+Cayennensis+Oswaldoi 40-Moucheti 14-cayennensis 38-Micropyga 50-quadrispinosa 46-peresi 51-rorotaensis

64-trinidadensis

C-Dreisbachi+Microps: 22-dreisbachi 27-fluviatilis 57-sordellii

D-Evandromyia:10-brachyphalla 33-infraspinosa 39-Monstruosa 48-pinottii

E-Lutzomyia:13-carvalhoi 29-gomezi 35-lichyi 58-spathotrichia

F-Migonei+Saulensis:43-pacae 54-sericea 68-walkeri 52-saulensis

G-Nyssomyia: 3-anduzei 4-antunesi 25-flaviscutellata 61-sylvicola 67-umbratilis 69-whitmani 70-yuilli pajoti

H-Pilosa+Trichopygomyia: 15-chassigneti 47-pilosa 36-longispina 63-trichopyga

I-Pintomyia+Pressatia: 59-spinosa 16-choti 23-equatorialis 62-triacantha

J-Psathyromyia: 1-abonnenci 12-campbelli 20-dendrophyla 37-Lutziana 49-punctigeniculata 53-scaffi 56-shannoni

K-Psychodopygus: 2-amazonensis 6-ayrozai 8-bispinosa 17-claustrei 18-corossoniensis 19-davisi 21-dorlinsis 30-guyanensis 31-hirsuta 41-nocticola 44-panamensis 45-paraensis 60-squamiventris Maripaensis

L-Trichophoromyia: 9-brachipyga 26-flochi 34-ininii 66-ubiquitalis

M-Verrucarum: 42-odax 55-serrana

N-Viannamyia: 24-fariasi 28-furcata 65-tuberculata

The clustering algorithm was run 500 times and the best result according to the adequacy criterion is selected. CR and $OERC$ indices were calculated for the best result. Table 7 shows the clusters given by the clustering algorithm.

Dissimilarity data sets	Without taking into account the rules	Taking into account the rules
Cluster 1	5/A 6/K 7/A 10/D 19/K 22/C 25/G 27/C 30/K 31/K 35/E 36/H 45/K 64/B 66/L	5/A 7/A 11/A 32/A 37/J
Cluster 2	24/N 28/N	10/D 16/I 23/I 36/H 52/F 54/F 62/I 63/H
Cluster 3	48/D	2/K 6/K 8/K 17/K 18/K 19/K 21/K 30/K 31/K 41/K 44/K 45/K 60/K
Cluster 4	16/I 23/I 62/I	20/J 53/J 56/J
Cluster 5	1/J 4/G 13/E 20/J 56/J	24/N 28/N 65/N
Cluster 6	18/K 21/K 61/G	22/C
Cluster 7	9/L	33/D 35/E 39/D 59/I 68/F
Cluster 8	11/A 26/L 32/A 34/L	3/G 4/G 25/G 61/G 67/G 69/G 70/G
Cluster 9	2/K 17/K 33/D 39/D 41/K 44/K 49/J 58/E 63/H	27/C 57/C
Cluster 10	40/B	12/J
Cluster 11	3/G 15/H 47/H 67/G 70/G	9/L 26/L 34/L 66/L
Cluster 12	12/J 60/K	1/J 49/J
Cluster 13	37/J 54/F 68/F 69/G	14/B 15/H 38/B 40/B 42/M 46/B 47/H 48/D 51/B 55/M 64/B
Cluster 14	8/K 14/B 29/E 38/B 42/M 43/F 46/B 50/B 51/B 52/F 53/J 55/M 57/C 59/I 65/N	13/E 29/E 43/F 50/B 58/E

FIG. 7 – *Clustering Results: Sub-species of Phlebotominae genus Lutzomyia*

Tables 7 shows clearly that the 14-clusters partition given by the clustering algorithms, applied on dissimilarity taking the rules into account, is closer to the 14-classes partition known *a priori* than the 14-clusters partition given by the same clustering algorithms applied without taking into account the rules.

The computation of CR and $OERC$ indices corroborate this conclusion. The CR and $OERC$ indices obtained from the results displayed in Table 7 were, respectively, 0.192 and

0.528 for the dissimilarity without taking into account the hierarchical rules, and 0.750 and 0.228 for the dissimilarity taking into account the rules.

In conclusion, the algorithm had a better performance in clustering species of phlebotominae taking into consideration the hierarchical rules in the computation of dissimilarities.

8 Concluding remarks

The main contribution of this paper is to describe an approach to cluster symbolic data constrained by rules between variables in a quadratic time. We used for this approach a well known clustering algorithm, the dynamic clustering algorithm which performs on dissimilarity table, but the data were decomposed before using the Normal Symbolic Form (N.S.F.). We have detailed all the operations needed to perform this decomposition, and experimental results have proven the usefulness of this approach. We need now to adapt the N.S.F. to histogram variables and use it on other data set directly extracted from data bases.

References

Bock, H.-H. (2002). Clustering algorithms and kohonen maps for symbolic data. *Journal of the Japanese Society of Computational Statistics 15*, 1–13.

Bock, H.-H. and E. Diday (2000). *Analysis of Symbolic Data: Explanatory Methods for Extracting Statistical Information from Complex Data*. Heidelberg: Springer.

Chavent, M. (1998). A monothetic clustering method. *Pattern Recognition Letters 19*(11), 989–996.

Chavent, M. and Y. Lechevallier (2002). Dynamical clustering algorithm of interval data: Optimization of an adequacy criterion based on hausdorff distance. In A. S. et al (Ed.), *Classification, Clustering and Data Analysis*, Berlin, pp. 53–59. Springer.

Codd, E. F. (1971). Further normalization of the data base relational model. *IBM Research Report, San Jose, California RJ909*.

Csernel, M. (1998). On the complexity of computation with symbolic objects using domain knowledge. In M. V. Hans-Hermann Bock, Alfredo Rizzi (Ed.), *Exploratory Data Analysis in Empirical Research,Proceedings of the IFCS-98*, Roma, pp. 403–408. Springer.

Csernel, M. and F. A. T. de Carvalho (1998). On the complexity of computation with symbolic objects using domain knowledge. In M. V. A. Rizzi and H.-H. Bock (Eds.), *New Andvances in Data Science and Classification*, Berlin, pp. 403–408. Springer- Verlag.

Csernel, M. and F. A. T. de Carvalho (1999). Usual operations with symbolic data under normal symbolic form. *Applied Stochastic Models in Business and Industry 15*(4), 241–257.

Csernel, M. and F. A. T. de Carvalho (2002). On memory requirement with normal symbolic form. In M. Schwaiger and O. Opitz (Eds.), *Exploratory Data Analysis in Empirical Research*, Munich, pp. 22–30. Springer.

De Carvalho, F., R. de Souza, M. Chavent, and Y. Lechevallier (2006). Adaptive hausdorff distances and dynamic clustering of symbolic interval data. *Pattern Recognition Letters 27*(3), 167–179.

De Carvalho, F. A. T. (1994a). Proximity coefficients between boolean symbolic objects. In E. D. et al. (Ed.), *New Approaches in Classification and Data Analysis*, Heildeberg, pp. 387–394. Springer-Verlag.

De Carvalho, F. A. T. (1994b). Proximity coefficients between boolean symbolic objects. In E. D. et al. (Ed.), *New Approaches in Classification and Data Analysis*, Heildeberg, pp. 387–394. Springer-Verlag.

De Carvalho, F. A. T. (1998). Extension based proximities between constrained boolean symbolic objects. In C. e. a. Hayashi (Ed.), *Analysis of Symbolic Data, Exploratory Methods for Extracting Statistical Information from Complex Data*, pp. 370 – 378. Springer - Verlag, Tokyo, Japan.

De Carvalho, F. A. T., P. Brito., and H.-H. Bock (2006). Dynamic clustering methods for interval data based on l2 distance. *Computational Statistics 21*(2), 231–250.

De Carvalho, F. A. T., M. Csernel, and Y. Lechevallier (2009). Clustering constrained symbolic data. *Pattern Recognition Letters 30*(11), 1037–1045.

Diday, E. (1973). La méthode des nuées dynamiques. *Revue de Statistique Appliquée 19*(2), 19–34.

Diday, E. and J. C. Simon (1976). Clustering analysis. In K. Fu (Ed.), *Digital Pattern Classification*, Berlin, pp. 47–94. Springer.

Everitt, B. (2001). *Cluster Analysis*. New York: Halsted.

Gordon, A. D. (1999). *Classification*. Boca Raton, Florida: Chapman and Hall/CRC.

Gowda, K. C. and E. Diday (1992). Symbolic clustering using a new dissimilarity. *IEEE Transactions on Systems Man and Cybernetics measure 22*, 368–378.

Gowda, K. C. and T. V. Ravi (1995a). Agglomerative clustering of symbolic objects using the concepts of both similarity and dissimilarity. *Pattern Recognition Letters 16*(6), 647–652.

Gowda, K. C. and T. V. Ravi (1995b). Divisive clustering of symbolic objects using the concepts both similarity and dissimilarity. *Pattern Recognition 28*(8), 1277–1282.

Hubert, L. and p. Arabie (1985). Comparing partitions. *Journal of Classification* (2), 193–218.

Ichino, M. and H. Yaguchi (1994). Generalized minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems, Man, and Cybernetics 24*(4), 698–708.

Lechevallier, Y. (1974). *Optimisation de quelques criteres en classification automatique et application a l'etude des modifications des proteines seriques en pathologie clinique*. Ph. D. thesis, Universite Paris-VI.

Lerat, N. and W. Lipski (1986). Nonapplicable nulls. *Theor. Comput. Sci. 46*(3), 67–82.

Ralambondrainy, H. (1995). A conceptual version of the k-means algorithm. *Pattern Recognition Letters 16*(11), 1147–1157.

Vignes, R. (1991). *Caracterisation automatique de groupes biologiques*. Ph. D. thesis, Universite Paris-VI.