Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Clustering constrained symbolic data

Francisco de A.T. de Carvalho [a,*], Marc Csernel [b,c], Yves Lechevallier [b]

[a] Centro de Informatica, CIn/UFPE, Av. Prof Luiz Freire, s/n, Cidade Universitaria, CEP 50.740-540, Recife, PE, Brazil
[b] INRIA, Rocquencourt, Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 le Chesnay Cedex, France
[c] University of Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France

ARTICLE INFO

Article history:
Received 20 July 2007
Received in revised form 9 October 2008
Available online 3 May 2009

Communicated by W. Pedrycz

Keywords:
Symbolic Data Analysis
Clustering algorithms
Normal symbolic form
Constraints
Dissimilarity functions

ABSTRACT

Dealing with multi-valued data has become quite common in both the framework of databases as well as data analysis. Such data can be constrained by domain knowledge provided by relations between the variables and these relations are expressed by rules. However, such knowledge can introduce a combinatorial increase in the computation time depending on the number of rules. In this paper, we present a way to cluster such data in polynomial time. The method is based on the following: a decomposition of the data according to the rules, a suitable dissimilarity function and a clustering algorithm based on dissimilarities.

1. Introduction

Cluster analysis is an exploratory data analysis field the aim of which is to organise 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 techniques can be divided into hierarchical and partitional methods (Jain et al., 1999; Gordon, 1999): hierarchical methods yield a complete hierarchy, i.e. a nested sequence of partitions of the input data, whereas partitioning methods seek to obtain a single partition of the input data into a fixed number of clusters, usually by optimizing an objective function.

In cluster analysis, the patterns to be grouped are usually represented as a vector of single quantitative or qualitative measurements, for which each column represents a variable. However, this model is too restrictive for representing complex data. In order to take into account the variability and/or uncertainty inherent to the data, variables must assume sets of categories or intervals, possibly even with frequencies or weights.

This paper addresses the partitioning of constrained symbolic data in a predefined number of clusters. This kind of data has been mainly studied in Symbolic Data Analysis (SDA), a domain related to multivariate analysis, pattern recognition and artificial intelligence. The aim is to provide suitable methods (clustering, factorial techniques, etc.) for managing data described through multi-valued variables, in which there are sets of categories, intervals or weight (probability) distributions in the cells of the data table (Bock and Diday, 2000). This allows the management of domain knowledge provided by relations between the variables and these relations are expressed by rules.

In a symbolic data table, the rows are the symbolic description and the columns are the symbolic variables. A symbolic variable is defined according to its type of domain. For example, for a symbolic description, an interval variable takes an interval of $\Re$ (set of real numbers) as its value. A set-valued variable takes a set of nominal categories as its value and a list-valued variable takes a list of ordered categories as its value. Table 1 displays an example of two symbolic descriptions described by three set-valued variables (Hand, Hand color and Finger) and one list-valued variable (Thorax size).

The previous array represents two symbolic descriptions called d_1 and d_2. The following dependency rules, r_1 and r_2, constrain the data.

$$\text{Hand} \in \{\text{absent}\} \Rightarrow \text{Hand\_Color} = \text{N.A.} \qquad (r_1)$$

$$\text{Hand} \in \{\text{absent}\} \Rightarrow \text{Finger} = \text{N.A.} \qquad (r_2)$$

SDA has provided suitable tools for clustering symbolic data: agglomerative hierarchical methods (Gowda and Diday, 1991, 1992; Ichino and Yaguchi, 1994; Gowda and Ravi, 1995a,b, 1999a; El-Sonbaty and Ismail, 1998a; Guru et al., 2004; Guru and Kiranagi, 2005), divisive hierarchical methods (Chavent, 2000),

* Corresponding author. Tel.: +55 81 21268430; fax: +55 81 21268438.
E-mail addresses: fatc@cin.ufpe.br, francisco.carvalho@pq.cnpq.br (F.A.T. de Carvalho), Marc.Csernel@inria.fr (M. Csernel), Yves.Lechevallier@inria.fr (Y. Lechevallier).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.04.009

Table 1
Example of symbolic descriptions.

	Hand	Hand_Color	Finger	Thorax_size
d_1	{absent, present}	{red, blue}	{absent, present}	{small, big}
d_2	{absent, present}	{red, green}	{absent, present}	{small}

hard cluster partitioning algorithms (Ralambondrainy, 1995; Bock, 2002; Chavent and Lechevallier, 2002; Mali and Mitra, 2003; Souza and DeCarvalho, 2004; Chavent et al., 2006; De Carvalho et al., 2006a; De Carvalho et al., 2006b; De Carvalho and Lechevallier, 2009) and fuzzy cluster partitioning algorithms (El-Sonbaty and Ismail, 1998b; De Carvalho, 2007). These methods differ with regard to the type of symbolic data considered, cluster structures and/or the clustering criteria considered. However, none of these methods is able to take constraints into account, because this usually leads to a combinatorial computation time due to the number of rules. The main contribution of this paper is to present an approach that is able to cluster constrained symbolic data. In this approach, constraints are taken into account during the computation of the dissimilarity between the symbolic descriptions. Using a dissimilarity function allows clustering all kinds of items, provided that a dissimilarity table can be built from them. Thus, we can cluster individuals that differ from the usual vector of $\Re^p$ representation. The constrained symbolic data are then clustered using clustering algorithms applied to a dissimilarity data matrix. We describe a suitable dissimilarity function and we use a method inspired by database technology to compute this function in the presence of constraints in a polynomial time, regardless of the number of rules. This method, called normal symbolic form (Csernel and De Carvalho (1999)), is fully efficient when the variables connected by the constraints form a tree or a set of trees It gives its best results with set-valued or list-valued variables. Using this method, we can compute a distance in a polynomial time in the presence of constraints and then cluster the constrained symbolic data.

This paper is organised as follows: Section 2 presents constrained symbolic data and Section 3 describes a dissimilarity function for computing dissimilarities on constrained symbolic data efficiently thanks to the normal symbolic form. Section 4 presents clustering algorithms with a dissimilarity matrix as input, applied to a biological data set that includes constraints. Finally, Section 5 gives the conclusions and final remarks.

2. Constrained symbolic data

A number of different definitions of symbolic descriptions are available in the literature. Here, we follow those given by Diday (1988), Gowda and Diday (1991) and Bock and Diday (2000): symbolic descriptions are defined by a logical conjunction of events linking values and variables, in which the variables can take one or more values and all objects need not be defined by the same variables.

Given a symbolic variable y, an *event* $e = [y \in X]$ is a value-variable pair that links feature variables and feature values of objects. For example, $e = [\text{color} \in \{\text{blue}, \text{red}\}]$ is an event that indicates that the set-valued variable color takes either a blue or a red value. Given a set of symbolic variables $\{y^1, \ldots, y^p\}$, a *symbolic description* is a conjunction of events of a particular object: $s = [y^1 \in X^1] \wedge \ldots \wedge [y^p \in X^p]$. For example, $s = [\text{color} \in \{\text{blue}, \text{red}\}] \wedge [\text{height} \in [160, 190]]$ is a symbolic description with the following properties: (a) color is either blue or red; (b) height ranges between 160 and 190.

According to Bock and Diday (2000), each symbolic description can be represented by a vector of feature values $\boldsymbol{x} = (X^1, \ldots, X^j, \ldots, X^p)$, where a feature value $X^j\,(j = 1, \ldots, p)$ can be either a set of nominal categorical values, a list of ordinal categorical values or an interval, if the symbolic variable y_j is, respectively, set-valued, list-valued or interval-valued. For example, the symbolic description $s = [\text{color} \in \{\text{green}, \text{red}\}] \wedge [\text{height} \in [160, 190]]$ can be represented by the vector of feature values $\boldsymbol{x} = (\{\text{green}, \text{red}\}, [160, 190])$.

An individual description can be represented as a vector of feature values $\boldsymbol{z} = (z^1, \ldots, z^p)$, where a feature value $z^j\,(j = 1, \ldots, p)$ can be a single nominal categorical value, a single ordinal categorical value or a single quantitative value. Given a set of individual descriptions $E = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, where $\boldsymbol{z}_i = (z_i^1, \ldots, z_i^p)$, the *extension* of the symbolic description $\boldsymbol{x}$ is defined as $ext(\boldsymbol{x}) = \{\boldsymbol{z}_i \in E : z_i^j \in X^j, j = 1, \ldots, p\}$. The *virtual extension* of the symbolic description $\boldsymbol{x}$ is defined as $vext(\boldsymbol{x}) = \{\boldsymbol{z} = (z^1, \ldots, z^p) : (z^1, \ldots, z^p) \in X_1 \times \ldots \times X_p\}$. Of course, the following relation holds: $ext(\boldsymbol{x}) \subseteq vext(\boldsymbol{x})$.

Example. In the following individual descriptions (left side) and in the following symbolic descriptions (right side)

	color	size
Beetle1	blue	small
Beetle2	red	medium

	color	size
species1	{blue,red}	{small,medium}
species2	{yellow,green}	{medium,big}

the extension of specie1 is $ext(\text{specie1}) = \{\text{Beetle1}, \text{Beetle2}\}$. The virtual extension of *species1* has four individual descriptions: {(blue, small), (blue, medium), (red, small), (red, medium)}.

2.1. Constraints on symbolic descriptions

Symbolic descriptions can be constrained by dependencies between pairs of variables expressed by rules. Such rules can be considered as constraints in the description space; they produce "holes" in the space because they forbid some individual descriptions to be considered as a part of the virtual extension of a symbolic description. Each dependency is represented by a rule. We call the variables associated with the premise and the conclusion of each rule premise variable and conclusion variable, respectively.

Let $\mathscr{D}$ be a set (or a ordered list) of categories. In the following equation, $\mathscr{P}^*(\mathscr{D})$ denotes the power set of $\mathscr{D}$ without the empty set. Let y_1 and y_2 be set-valued (or list-valued) variables, the domains of which are $\mathscr{D}_1$ and $\mathscr{D}_2$, respectively. *A hierarchical dependency* between the variables y_1 and y_2 is expressed by the following kind of rule called a hierarchical rule:

$$\text{if } [y_1 \in \mathscr{P}^*(\mathscr{D}_1)] \Rightarrow [y_2 = \text{N.A.}]$$

where the term N.A. means *not applicable*, hence, the variable does not exist. The rule r_1 described below is an example of such a rule:

$$\text{if } [\text{Wings} \in \{\text{absent}\}] \Rightarrow [\text{Wings\_color} = \text{N.A.}]. \qquad (\text{r}_1)$$

This rule reduces the number of individual descriptions belonging to the extension of a symbolic description as well as the number of dimensions of a symbolic description. It was shown in De Carvalho et al. (1998) that computation using rules leads to an exponential computation time depending on the number of rules. In order to avoid this combinatorial computation time, we have introduced the normal symbolic form (Csernel and De Carvalho, 1999).

2.2. Mutual interaction of rules

The different rules can interact mutually. One can say that hierarchical dependencies induce a kind of inheritance. This induces the following consequence: If we have the two following rules: