Pattern Recognition 46 (2013) 3030–3039

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

A new topological clustering algorithm for interval data

Guénaël Cabanes [a,*,1], Younès Bennani [a], Renaud Destenay [b], André Hardy [b]

[a] LIPN-CNRS, UMR 7030, Université Paris 13, 99 Avenue J-B. Clément, 93430 Villetaneuse, France
[b] Namur Center for Complex Systems (naXys) and Department of Mathematics, University of Namur (FUNDP), 8 Rempart de la Vierge, B 5000 Namur, Belgium

ARTICLE INFO

Article history:
Received 6 February 2012
Received in revised form
4 February 2013
Accepted 29 March 2013
Available online 10 April 2013

Keywords:
Interval data
Clustering
Self-organizing map

ABSTRACT

Clustering is a very powerful tool for automatic detection of relevant sub-groups in unlabeled data sets. In this paper we focus on interval data: i.e., where the objects are defined as hyper-rectangles. We propose here a new clustering algorithm for interval data, based on the learning of a Self-Organizing Map. The major advantage of our approach is that the number of clusters to find is determined automatically; no a priori hypothesis for the number of clusters is required. Experimental results confirm the effectiveness of the proposed algorithm when applied to interval data.

1. Introduction

Unsupervised classification, or clustering, is a very powerful tool for automatic detection of relevant sub-groups (or clusters) in unlabeled data sets, when one does not have prior knowledge about the underlying structure of these data. Patterns in the same cluster should be similar to each other, while patterns in different clusters should not (internal homogeneity and external separation). Clustering plays an indispensable role for understanding various phenomena described by data sets and is considered as one of the most challenging tasks in unsupervised learning. Various approaches have been proposed to solve the problem [35,40,41,25,24].

However, most clustering algorithms are defined to deal with vectorial data in $\mathbb{R}^d$. This kind of representation is frequently used to analyze data from physical measurements, counts or indices, but there are many other kinds of information that cannot be described with vectors. This is the case of complex data described for example with a text, a picture or a hierarchical structure. In this paper we focus on interval data (also known as symbolic interval data). In a vectorial space, interval data are defined by hyper-rectangles. A given data x is thus defined as a closed and bounded interval in $\mathbb{R}^d$, characterized by two vectors, the lower bound $(x_l = [x_{l1}, \ldots, x_{ld}])$ and the upper bound $(x_u = [x_{u1}, \ldots, x_{ud}])$, such that $\forall j \in [1, \ldots, d], x_{lj} \leq x_{uj}$. Intervals are often used to model quantities which vary between two bounds, upper and lower, without further assumptions on the distribution between these bounds [3,21,2,20]. Several clustering methods are available for interval variables. For example, [26] presented an iterative relocation algorithm to partition a set of symbolic objects into classes so as to minimize the sum of the description potential of the classes. De Souza and de Carvalho [18] proposed partitioning clustering methods for interval data based on city-block distances. SCLUST [50] is a partitioning clustering method and a symbolic extension of the well-known Dynamical Clustering method [19]. DIV [10] is a symbolic hierarchical monothetic divisive clustering procedure based on the extension of the within class sum-of-squares criterion. SCLASS [48] and SPART [33] are symbolic hierarchical monothetic divisive methods based on the generalized Hypervolumes clustering criterion. Hardy [32] developed a module called SHICLUST containing the symbolic extensions of four well-known classic hierarchical clustering methods: the single linkage, complete linkage, centroid and Ward methods. The corresponding aggregation indices used the L_1, L_2, Hausdorff and De Carvalho [13] dissimilarity measures [45,23]. The hierarchical component of Hipyr [7] also contains extensions of the four classic hierarchical clustering methods. Other clustering methods for interval data can be found in [11,18,31,4,39,15].

We present here a new clustering algorithm for interval data, based on the learning of a Self-Organizing Map (SOM) [42]. This unsupervised learning algorithm is a popular nonlinear technique for dimensionality reduction and data visualization, with a very low computational cost. It can be seen as a K-means algorithm with topological constraints, usually with a better overall clustering performance [12]. Bock [4,5] proposed a visualization of symbolic interval data by constructing a SOM. In the SODAS software [21], such a map is constructed in the SYKSOM module. SYKSOM assumes a data table of n items that are described by p interval-type variables. The n items are first clustered into a smaller number of mini-clusters (reduction step), and these mini-clusters are

* Corresponding author. Tel.: +61293512267.
E-mail address: guenael.cabanes@lipn.univ-paris13.fr (G. Cabanes).
[1] Permanent address: Behaviour and Genetics of Social Insects Lab, School of Biological Sciences A12, University of Sydney, NSW 2006, Australia.

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.03.023

then assigned to the vertices of a fixed, prespecified rectangular lattice $\mathcal{L}$ of points in the plane such that similar clusters (in the original data space) are represented by neighboring vertices in the lattice $\mathcal{L}$. Other papers concerning SOM algorithms for interval-valued data can be found in the literature [28,29,47,16,22,52]. For example, [22] uses a distance based on Hadamard product and [52] proposes a fuzzy representation based on Gowda and Diday's dissimilarity measure [27]. All these algorithms can be seen as vector quantization and visualization tools for symbolic interval data, and cannot be used directly to obtain a clustering of the data.

The proposed algorithm is a two-level clustering method for interval data. The key idea of the two-level clustering approach based on SOM is to combine the dimension reduction and the fast learning capabilities of SOM in the first level to construct a new reduced space, then to apply a clustering method in this new space to produce a final set of clusters in the second level (see [38,49] for examples with vectorial data). The two-level methods are known to reduce the computational time and allow a visual interpretation of the clustering results [8]. In particular, the use of SOM+K-means or SOM+Hierarchical clustering gives better results than the use of K-means or a Hierarchical clustering alone [8,9]. The major advantage of the new algorithm in comparison to existing methods is that the number of clusters to find is detected automatically, i.e., no a priori hypothesis for the number of clusters is required. This problem, also known as the model selection problem, is one of the most challenging in clustering. Indeed, the existing clustering algorithms for interval data need to have the number of clusters as a user-given parameter [50,26,18,33], which is usually very difficult to determine a priori.

The remainder of this paper is organized as follows. Section 2 presents an adaptation of SOM allowing an automatic two-level clustering. Section 3 describes the new algorithm for interval data. In Section 4 we present the experimental protocol and results are shown in Section 5. In Section 6, we compare the new algorithm with existing methods on artificial and real datasets. Conclusions are given in Section 7.

2. Simultaneous two-level clustering of self-organizing map

Kohonen's Self-Organizing Map (SOM) can be described as a competitive unsupervised learning neural network [42]. When an observation is recognized, the activation of an output cell – competition layer – inhibits the activation of other neurons and reinforces itself. It is said that it follows the so called "Winner Takes All" rule. Actually, neurons are specialized in the recognition of one kind of observation. A SOM often consists of a two-dimensional map of neurons which are connected to n inputs according to n weight connections $w^j=(w^j_1,\ldots,w^j_d)$ and to their neighbors with topological links. A training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; two close observations in the input space would activate two close units of the SOM. An optimal spatial organization is determined by the SOM from the input data, and when the dimension of the input space is lower than three, both the position of weight vectors and direct neighborhood relations between cells can be represented visually. Thus, a visual inspection of the map provides qualitative information about the map and the choice of its architecture. The winner neuron updates its prototype vector, making it more sensitive for later presentation of that type of input. This allows different cells to be trained for different types of data. To achieve a topological mapping, the neighbors of the winner neuron can adjust their prototype vector towards the input vector as well, but at a lesser degree, depending on how far away they are from the winner. Usually a radial symmetric Gaussian neighborhood function K_{ij}, between two neurons i and j, is used for this purpose.

The key idea of the two-level clustering approach based on SOM is to combine the dimension reduction and the fast learning capabilities of SOM in the first level to construct a new reduced vector space, and to apply another clustering method in this new space to produce a final set of clusters in the second level. Although the two-level methods are more interesting than the traditional approaches (in particular by reducing the computational time and by allowing a visual interpretation of the partition result [6,38,49,43]), the data segmentation obtained from the SOM is not optimal, since part of the information is lost during the first stage (dimension reduction). Moreover, this separation in two stages is not suited for a dynamic (incremental) segmentation of data which move in time, in spite of important needs for analysis tools for this type of data. The S2L-SOM algorithm (Simultaneous Two-Levels-SOM, [8]) has been proposed to overcome these problems by simultaneous performing learning and clustering of the SOM from data information.

2.1. The S2L-SOM algorithm

In the S2L-SOM algorithm, it is proposed to associate to each neighborhood connection a real value ν_{ij} which indicates the relevance of the connected neurons i and j. This value is representative of the data distribution between i and j, and can be viewed as the number of data having i and j as the two best representatives neurons. Given the organization constraint of the SOM, both closest prototypes of each data must be connected by a topological connection. This connection "will be rewarded" by an increase of its value, whereas all other connections from the winner neuron "are punished" by a reduction of their values. The values of ν will be used to create sets of connected prototypes; each set not connected to the others is representative of one cluster. Thus, at the end of the training, a set of inter-connected prototypes will be an artificial image of a relevant sub-group of the whole data set.

Connectionist learning is often presented as a minimization of a cost function. In our case, it will be carried out by the minimization of the distance between the input samples and the map prototypes, weighted by a neighborhood function K_{ij}. To do that, we use a gradient algorithm [1]. The cost function to be minimized is defined by

$$\tilde{R}(w)=\frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{M}K_{j,u^*(x^{(k)})}\|w^j-x^{(k)}\|^2 \tag{1}$$

where N represents the number of learning samples, M the number of neurons in the map, $u^*(x^{(k)})$ is the neuron having the closest weight vector to the input pattern $x^{(k)}$, and K_{ij} is a positive symmetric kernel function: the neighborhood function [42]. The relative importance of a neuron i compared to a neuron j is weighted by the value of the kernel function K_{ij} which can be defined as

$$K_{ij}=\frac{1}{\lambda(t)}\times e^{-d_1^2(i,j)/\lambda^2(t)} \tag{2}$$

where $\lambda(t)$ is the temperature function modeling the topological neighborhood extent, defined as

$$\lambda(t)=\lambda_i\left(\frac{\lambda_f}{\lambda_i}\right)^{t/t_{max}} \tag{3}$$

where λ_i and λ_f are the initial and the final temperature respectively. t_{max} is the maximum number allocated to the time (number of iterations for the x learning sample). $d_1(i,j)$ is the Manhattan distance defined between two neurons i and j on the map grid, with coordinates (k, m) and (r, s) respectively

$$d_1(i,j)=\|r-k\|+\|s-m\|. \tag{4}$$

The S2L-SOM training process is highly similar to the Competitive Hebbian Learning (CHL) approach [46]. The difference lies in that the