Pattern Recognition 45 (2012) 447–464

Contents lists available at ScienceDirect

Pattern Recognition

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

Partitioning hard clustering algorithms based on multiple dissimilarity matrices

Francisco de A.T. de Carvalho [a,*], Yves Lechevallier [b], Filipe M. de Melo [a]

[a] Centro de Informática, Universidade Federal de Pernambuco, Av. Prof. Luiz Freire, s/n - Cidade Universitária, CEP 50740-540 Recife (PE), Brazil
[b] INRIA—Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France

ARTICLE INFO

Article history:
Received 9 September 2010
Received in revised form
25 May 2011
Accepted 28 May 2011
Available online 7 June 2011

Keywords:
Partitioning clustering algorithms
Relational data
Relevance weight
Multiple dissimilarity matrices

ABSTRACT

This paper introduces hard clustering algorithms that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. These matrices have been generated using different sets of variables and dissimilarity functions. These methods are designed to furnish a partition and a prototype for each cluster as well as to learn a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fitting between the clusters and their representatives. These relevance weights change at each algorithm iteration and can either be the same for all clusters or different from one cluster to another. Experiments with data sets (synthetic and from UCI machine learning repository) described by real-valued variables as well as with time trajectory data sets show the usefulness of the proposed algorithms.

1. Introduction

Clustering methods organize a set of items into clusters such that items within a given cluster have a high degree of similarity, whereas those of different clusters have a high degree of dissimilarity. These methods have been widely applied in fields such as taxonomy, image processing, information retrieval and data mining. The most popular clustering techniques are hierarchical and partitioning methods [1,2].

Hierarchical methods yield complete hierarchy, i.e., a nested sequence of partitions of the input data. Hierarchical methods can be agglomerative [3–7] or divisive [8–12]. Agglomerative methods yield a sequence of nested partitions starting with trivial clustering in which each item is in a unique cluster and ending with clustering in which all items are in the same cluster. A divisive method starts with all items in a single cluster and performs a splitting procedure until a stopping criterion is met (usually upon obtaining a partition of singleton clusters).

Partitioning methods seek to obtain a single partition of the input data into a fixed number of clusters. These methods often look for a partition that optimizes (usually locally) an objective function. To improve cluster quality, the algorithm is run multiple times with different starting points and the best configuration obtained from the total runs is used as the output clustering. Partitioning methods can be divided into hard clustering [13–17] and fuzzy clustering [18–22]. Hard clustering furnishes a partition in which each object of the data set is assigned to one and only one cluster. Fuzzy clustering generates a fuzzy partition that furnishes a degree of membership of each pattern in a given cluster. This gives the flexibility to express that objects belong to more than one cluster at the same time.

There are two common representations of the objects upon which clustering can be based: feature data and relational data. When each object is described by a vector of quantitative or qualitative values, the set of vectors describing the objects is called a *feature data*. Alternatively, when each pair of objects is represented by a relationship, then it is called *relational data*. The most common case of relational data is when one has (a matrix of) dissimilarity data, say $R=[r_{kl}]$, where r_{kl} is the pairwise dissimilarity (often a distance) between objects k and l. Clustering of relational data is very useful when the objects cannot be described by a vector of feature values, when the distance measure does not have a closed form, etc. [10,23–26]. Recently, Frigui et al. [27] proposed CARD, a relational fuzzy clustering algorithm that is able to partition objects taking into account multiple dissimilarity matrices and that learns a relevance weight for each dissimilarity matrix in each cluster. CARD is mainly based on the well-known fuzzy clustering algorithms for relational data NERF [26] and FANNY [10]. As remarked by [27], several applications can benefit from relational clustering algorithms based on multiple dissimilarity matrices. In image data base categorization, the relationship among the objects may be described by multiple dissimilarity matrices and the most effective dissimilarity measures do not have a closed form or are not differentiable with respect to prototype parameters.

This paper extends the dynamic hard clustering algorithm for relational data [23,24], into hard clustering algorithms that are able to partition objects taking into account simultaneously their

* 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), Yves.Lechevallier@inria.fr (Y. Lechevallier), fmm@cin.ufpe.br (F.M. de Melo).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.05.016

relational descriptions given by multiple dissimilarity matrices. The main idea is to obtain a collaborative role of the different dissimilarity matrices [28] to obtain a final partition. These dissimilarity matrices could have been generated using different sets of variables and a fixed dissimilarity function (in this case, the final partition is given according to different views (i.e., different sets of variables) describing the objects), or using a fixed set of variables and different dissimilarity functions (in this case, the final partition is given according to different dissimilarity functions) or using different sets of variables and dissimilarity functions. As pointed out by [27], the influence of the different dissimilarity matrices cannot be equally important in the definition of the clusters in the final partition. Thus, to obtain a meaningful partition from all dissimilarity matrices, the relational hard clustering algorithms given in this paper are designed to give a partition and a prototype for each cluster as well as to learn a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fitting between the clusters and their representatives. These relevance weights change at each algorithm's iteration and can either be the same for all clusters or different from one cluster to another.

This paper is organized as follows. Section 2 first reviews a partitioning dynamic hard clustering algorithm based on a single dissimilarity matrix (Section 2.1) and then introduces partitioning dynamic hard clustering algorithms based on multiple dissimilarity matrices with relevance weight for each dissimilarity matrix either estimated locally (Section 2.2.1) or estimated globally (Section 2.2.2). Section 3 gives empirical results to show the usefulness of these relational clustering algorithms. Finally, Section 4 gives final remarks and comments.

2. Partitioning hard clustering algorithms based on multiple dissimilarity matrices

This section introduces partitioning dynamic hard clustering algorithm for relational data that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices.

2.1. Dynamic hard clustering algorithm based on a single dissimilarity matrix

There are several relational clustering algorithms based on a single dissimilarity matrix in the literature like SAHN (sequential agglomerative hierarchical non-overlapping) [1] and PAM (partitioning around medoids) [10] but the paper starts with a brief description of the partitioning dynamic hard clustering algorithm for relational data based on a single dissimilarity matrix [23,24] (denote here *SRDCA*) because the algorithms here are based on it.

Let $E=\{e_1,\ldots,e_n\}$ be a set of n objects and let a dissimilarity matrix $\mathbf{D}=[d(e_i,e_l)]$, where $d(e_i,e_l)$ measures the dissimilarity between objects e_i and e_l $(i,l=1,\ldots,n)$. A particularity of this method is that it assumes that the prototype G_k of cluster C_k is a subset of fixed cardinality $1\leq q\ll n$ of the set of objects E (even if, for a matter of simplicity, very often $q=1$), i.e., $G_k\in E^{(q)}=\{A\subset E:|A|=q\}$. It looks for a partition $P=(C_1,\ldots,C_K)$ of E into K clusters and the corresponding prototypes $G_1,\ldots,G_K$ representing the clusters in P such that it is (locally) optimized an adequacy criterion (objective function) measuring the fit between the clusters and their prototypes.

The adequacy criterion measures the homogeneity of the partition P as the sum of the homogeneities in each cluster. It is defined as

$$J=\sum_{k=1}^{K}\sum_{e_i\in C_k}D(e_i,G_k)=\sum_{k=1}^{K}\sum_{e_i\in C_k}\sum_{e\in G_k}d(e_i,e) \tag{1}$$

where $J_k=\sum_{e_i\in C_k}D(e_i,G_k)$ is the homogeneity in cluster C_k $(k=1,\ldots,K)$ and

$$D(e_i,G_k)=\sum_{e\in G_k}d(e_i,e) \tag{2}$$

measures the matching between an example $e_i\in C_k$ and the cluster prototype $G_k\in E^{(q)}$.

The *SRDCA* relational clustering algorithm sets an initial partition and alternates two steps until convergence, when the criterion J reaches a stationary value representing a local minimum. This algorithm is summarized as follows.

Algorithm. Dynamic hard clustering algorithm for relational data

(1) *Initialization.*
Fix the number K of clusters;
Fix the cardinality $1\leq q\ll n$ of the prototypes G_k $(k=1,\ldots,K)$;
Set $t=0$;
Randomly select K distinct prototypes $G_k^{(0)}\in E^{(q)}$ $(k=1,\ldots,K)$;
Assign each object e_i to the closest prototype to obtain the partition $P^{(0)}=(C_1^{(0)},\ldots,C_K^{(0)})$ with $C_k^{(0)}=\{e_i\in E:D(e_i,G_k^{(0)})\leq D(e_i,G_h^{(0)}),(h=1,\ldots,K)\}$.

(2) *Step*1: *computation of the best prototypes.*
Set $t=t+1$;
The partition $P^{(t-1)}=(C_1^{(t-1)},\ldots,C_K^{(t-1)})$ is fixed.
Compute the prototype $G_k^{(t)}=G^*\in E^{(q)}$ of cluster $C_k^{(t-1)}$ $(k=1,\ldots,K)$ according to: $G^*=\mathrm{argmin}_{G\in E^{(q)}}\sum_{e_i\in C_k^{(t-1)}}D(e_i,G)=\mathrm{argmin}_{G\in E^{(q)}}\sum_{e_i\in C_k^{(t-1)}}\sum_{e\in G}d(e_i,e)$

(3) *Step*2: *definition of the best partition.*
The prototypes $G_k^{(t)}\in E^{(q)}$ $(k=1,\ldots,K)$ are fixed.
$test\leftarrow 0$
$P^{(t)}\leftarrow P^{(t-1)}$
for $i=1$ to n do
  find the cluster $C_m^{(t)}$ to which e_i belongs
  find the winning cluster $C_k^{(t)}$ such that
    $k=\mathrm{argmin}_{1\leq h\leq K}D(e_i,G_h^{(t)})=\mathrm{argmin}_{1\leq h\leq K}\sum_{e\in G_h}d(e_i,e)$
  if $k\neq m$
    $test\leftarrow 1$
    $C_k^{(t)}\leftarrow C_k^{(t)}\cup\{e_i\}$
    $C_m^{(t)}\leftarrow C_m^{(t)}\backslash\{e_i\}$

(4) *Stopping criterion.* If $test=0$ then STOP; otherwise go to 2 (Step 1).

Let $E=\{e_1,\ldots,e_n\}$ be the set of n objects and let p dissimilarity matrices $\mathbf{D}_j=[d_j(e_i,e_l)]$ $(j=1,\ldots,p)$, where $d_j(e_i,e_l)$ gives the dissimilarity between objects e_i and e_l $(i,l=1,\ldots,n)$ on dissimilarity matrix $\mathbf{D}_j$.

The *SRDCA* relational clustering algorithm can be changed into the "dynamic hard clustering algorithm based on multiple dissimilarity matrices" (denoted here *MRDCA*) to take into account simultaneously these p dissimilarity matrices $\mathbf{D}_j$. For that, the adequacy criterion of the *SRDCA* relational clustering algorithm is modified into

$$J=\sum_{k=1}^{K}\sum_{e_i\in C_k}D(e_i,G_k)=\sum_{k=1}^{K}\sum_{e_i\in C_k}\sum_{j=1}^{p}D_j(e_i,G_k)$$
$$=\sum_{k=1}^{K}\sum_{e_i\in C_k}\sum_{j=1}^{p}\sum_{e\in G_k}d_j(e_i,e) \tag{3}$$

in which

$$D(e_i,G_k)=\sum_{j=1}^{p}D_j(e_i,G_k)=\sum_{j=1}^{p}\sum_{e\in G_k}d_j(e_i,e) \tag{4}$$