Clustering of Multiple Dissimilarity Data Tables for Documents Categorization

Yves Lechevallier, Francisco De Carvalho, Thierry Despeyroux, Filipe De Melo

Clustering of Multiple Dissimilarity Data Tables for Documents Categorization

Yves Lechevallier[1], Francisco de A. T. de Carvalho[2], Thierry Despeyroux[1], and Filipe M. de Melo[2]

[1] INRIA, Paris-Rocquencourt
78153 Le Chesnay cedex, France,
{*Yves.Lechevallier,Thierry.Despeyroux*}*@inria.fr*
[2] Centro de Informatica -CIn/UFPE
Av. Prof. Luiz Freire, s/n -Cidade Universitaria, CEP 50740-540, Recife-PE, Brésil, {*fatc,fmm*}*@cin.ufpe.br*

Abstract. This paper introduces a clustering algorithm that is able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. These matrices could have been generated using different sets of variables and a fixed dissimilarity function, using a fixed set of variables and different dissimilarity functions or using different sets of variables and dissimilarity functions. This method, which is based on the dynamic hard clustering algorithm for relational data, is designed to provided 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 fit between clusters and their representatives. These relevance weights change at each algorithm iteration and are different from one cluster to another. Experiments aiming at obtaining a categorization of a document data base demonstrate the usefulness of this partitional clustering method.

Keywords: Clustering Analysis, Relational Data, Documents Categorization

1 Introduction

Clustering is a popular task in knowledge discovering and it is applied in various fields including data mining, pattern recognition, computer vision, etc (Gordon (1999), Jain et al (1999)). Clustering methods aims at organizing a set of objects into clusters such that items within a given cluster have a high degree of similarity, while items belonging to different clusters have a high degree of dissimilarity. The most popular clustering techniques are hierarchical and partitioning methods. Partitioning methods seek to obtain a single partition of the input data into a fixed number of clusters. Such methods often look for a partition that optimizes (locally) an adequacy criterion function.

There are two common representations of the objects upon which clustering can be based : (usual or symbolic) 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. When each (complex) object is described by a vector of sets of categories, intervals or weight histograms, the set of vectors describing the objects is called a symbolic feature data. Symbolic data has been mainly studied in Symbolic Data Analysis (SDA) (Bock and Diday (2000)). Alternatively, when each pair of objects is represented by a relationship, then we have relational data. The most common case of relational data is when we have (a matrix of) dissimilarity data, say $R = [r_{il}]$, where r_{il} is the pairwise dissimilarity (often a distance) between objects i and l.

This paper gives a clustering algorithm that is able to partition objects taking simultaneously into account their relational descriptions given by multiple dissimilaritity matrices. The main idea is to obtain a collaborative role of the different dissimilarity matrices (Pedrycz (2002)) in order to obtain a final consensus partition (Leclerc and Cucumel (1987)). These dissimilarity matrices could have been generated using different sets of variables and a fixed dissimilarity function (the final partition gives a consensus between different views (sets of variables) describing the objects), using a fixed set of variables and different dissimilarity functions (the final partition gives the consensus between different dissimilarity functions) or using different sets of variables and dissimilarity functions. Moreover, the influence of the different dissimilarity matrices is not equally important in the definition of the clusters in the final consensus partition. Thus, in order to obtain a meaningful partition from all dissimilarity matrices, it is necessary to learn cluster-dependent relevance weights for each dissimilarity matrix.

Frigui et al (2007) proposed CARD, a clustering algorithm that is able to partition objects taking into account multiple dissimilaritity matrices and that learns a relevance weight for each dissimilarity matrix in each cluster. CARD is mainly based on the well know fuzzy clustering algorithms for relational data RFCM (Hathaway et al (1989)) and FANNY (Kaufman and Rousseeuw (1990)).

The clustering algorithm given in this paper is 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 fit between clusters and their representatives. These relevance weights change at each algorithm iteration and are different from one cluster to another. However, this method is based on the dynamic hard clustering algorithm for relational data (Lechevallier (1974), De Carvalho et al (2008), De Carvalho et al (2009)) as well as on the dynamic clustering method based on adaptive distances (Diday and Govaert (1977), De Carvalho and Lechevallier(2009)). The adaptive dynamic clustering method gives a partition as well as a prototype for each cluster and learns a relevance weight for each variable in each cluster. In order to demonstrate the usefulness of this clustering algorithm,

experiments were designed in order to obtain a categorization of a document data base.

This paper is organized as follows. Section 2 presents a partitioning clustering algorithm based on multiple dissimilarity matrices. In order to illustrate the usefulness of this clustering method, section 3 shows the application of this algorithm in order to obtain a categorization of a document data base. Finally, section 4 presents the conclusions.

2 A Dynamic Clustering Algorithm Based on Multiple Dissimilarity Matrices

In this section, we introduce an extension of the dynamic clustering algorithm for relational data (De Carvalho et al (2008)) which is able to partition objects taking simultaneously into account their relational descriptions given by multiple dissimilaritity matrices.

Let $E = \{e_1, \ldots, e_n\}$ be a set of n examples and let p dissimilarity $n \times n$ matrices $(\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_p)$ where $\mathbf{D}_j[i,l] = d_j(e_i, e_l)$ gives the dissimilarity between objects e_i and e_l on dissimilarity matrix $\mathbf{D}_j$. Assume that the prototype g_k of cluster C_k belongs to the set of examples E, *i.e.*, $g_k \in E \ \forall k = 1, \ldots, K$.

The dynamic hard clustering algorithm with relevance weight for each dissimilarity matrix looks for a partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the corresponding prototype $g_k \in E$ representing the cluster C_k in P such that an adequacy criterion (objective function) measuring the fit between the clusters and their prototypes is locally optimized. The adequacy criterion is defined as

$$J = \sum_{k=1}^{K} \sum_{e_i \in C_k} d^{(k)}(e_i, g_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_k^j d_j(e_i, g_k) \tag{1}$$

in which

$$d^{(k)}(e_i, g_k) = \sum_{j=1}^{p} \lambda_k^j d_j(e_i, g_k) \tag{2}$$

is the dissimilarity between an example $e_i \in C_k$ and the cluster prototype $g_k \in E$ parameterized by relevance weight vector $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^j, \ldots, \lambda_k^p)$ where λ_k^j is the weight between the dissimilarity matrix $\mathbf{D}_j$ and the clusters C_k, and $d_j(e_i, g_k)$ is the local dissimilarity d_j between an example $e_i \in C_k$ and the cluster prototype $g_k \in E$.

The relevance weight matrix $\boldsymbol{\lambda}$ composed by K relevance weight vectors $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^j, \ldots, \lambda_k^p)$ changes at each iteration, *i.e.*, they are not determined absolutely, and are different from one cluster to another. Our clustering algorithm alternates the three following steps:

Step 1: Definition of the Best Prototypes

In this step, the partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the relevance weight matrix $\boldsymbol{\lambda}$ are fixed.

Proposition 1. *The prototype $g_k = e_l \in E$ of cluster C_k, which minimizes the clustering criterion J, is computed according to:*

$$l = \arg \min_{1 \leq h \leq n} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_h^j \, d_j(e_i, e_h) \tag{3}$$

Step 2: Definition of the Best Relevance Weight Matrix

In this step, the partition $P = (C_1, \ldots, C_K)$ of E and the vector of prototypes $\mathbf{g} = (g_1, \ldots, g_K)$ are fixed.

Proposition 2. *The element j of the relevance weight vector $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^p)$, which minimizes the clustering criterion J under $\lambda_k^j > 0$ and $\prod_{j=1}^{p} \lambda_k^j = 1$, is calculated by the following expression:*

$$\lambda_k^j = \frac{\left\{\prod_{h=1}^{p} \left[\sum_{e_i \in C_k} d_h(e_i, g_k)\right]\right\}^{\frac{1}{p}}}{\left[\sum_{e_i \in C_k} d_j(e_i, g_k)\right]} \tag{4}$$

Step 3: Definition of the Best Partition

In this step, the vector of prototypes $\mathbf{g} = (g_1, \ldots, g_K)$ and the relevance weight matrix $\boldsymbol{\lambda}$ are fixed.

Proposition 3. *The cluster C_k, which minimize the criterion J, is updated according to the following allocation rule:*

$$C_k = \{e_i \in E : d^{(k)}(e_i, g_k) < d^{(h)}(e_i, g_h) \forall h \neq k \ \} \tag{5}$$

If the minimum is not unique, e_i is assigned to the class having the smallest index

It's easy to demonstrate that each preview step decreases the criterion J. The dynamic hard clustering algorithm with relevance weight for each dissimilarity matrix sets an initial partition and alternates three steps until convergence, when the criterion $J(P, \boldsymbol{\lambda}, \mathbf{g})$ reaches a stationary value representing a local minimum. This algorithm is summarized below.

The Dynamic Hard Clustering Algorithm with Relevance Weight Matrix

1. *Initialization.*
   Fix the number K of clusters;
   Randomly select K distinct objects $g_k \in E$;
   Set the Relevance Weight Matrix $\boldsymbol{\lambda}$ where $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^p) = (1, \ldots, 1)$;
   Assign each object e_i to the closest prototype in order to obtain the partition $P = (C_1, \ldots, C_K)$ where C_k is constructed by the rule (5).

2. *Step 1: definition of the best prototypes.*
   The partition $P = (C_1, \ldots, C_K)$ and the relevance weight matrix $\boldsymbol{\lambda}$ are fixed.
   Compute the prototype $g_k \in E$ of cluster C_k according to equation (3)
3. *Step 2: definition of the best relevance weight matrix.*
   The vector of prototypes $\mathbf{g}$ and the partition $P = (C_1, \ldots, C_K)$ are fixed. For each k compute the component of the weight vector $\boldsymbol{\lambda}_k$ according to equation (4)
4. *Step 3: definition of the best partition.*
   The vector of prototypes $\mathbf{g}$ and the relevance weight matrix $\boldsymbol{\lambda}$ are fixed. Construct the new partition $P' = (C'_1, \ldots, C'_K)$ with the rule given by (5) and control the convergence by:
   $test \leftarrow 0$;
   for $i = 1$ to n do
      e_i belonged to the class C_m and belongs to the winning cluster C'_k
      if $k \neq m$ then $test \leftarrow 1$;
   $P \leftarrow P'$;
5. *Stopping criterion.* If $test = 0$ then STOP, otherwise go to 2 (Step 1).

3 Application: document data base categorization

To illustrate the usefulness of the proposed clustering algorithm, we use it to categorize a document data base. The document data base is a collection of reports produced by every Inria (The French National Institute for Research in Computer Science and Control) research team in 2007. Research teams are grouped into scientific *themes* that do not correspond to an organizational structure (such as departments or divisions), but act as a virtual structure for the purpose of presentation, communication and evaluation. Choice of themes and team allocation are mostly related to strategic objectives and scientific closeness between existing teams, but also take in account some geographical constraints, such as the desire for a theme to be representative of most Inria centers. Our aim is to compare the categorization given automatically by the clustering algorithm introduced in this paper with the *a priori* expert categorization given by INRIA.

These reports are written in English. The sources are LaTeX documents, and are automatically translated into XML, then to HTML to be published on the Web. In the rest of the paper we implicitly refer to the XML version of the Activity Report. The logical structure of the RA is defined by an XML DTD with a few mandatory sections and some optional parts.

In this application we considered activity reports from 164 INRIA research teams in 2007. On each activity report, 4 sections have been selected to describe a research team: *overall objectives*, *scientific foundations*, *dissemination* and *new results*. The *overall objectives* part defines the research objectives and *scientific foundations* provides the scientific background followed

Fig. 1. INRIA research categorization and example of the Activity Report summary

by potential applications of the research domain. *Dissemination* includes any teaching activity, involvement with the research community (program committees, editorial boards, conference and workshop organization) and seminars. The *new results* includes the principal results obtained during this year.

From these activity reports we initially obtained 4 feature data tables, each table with 164 individuals (INRIA research team) described by the frequent words (categories) present in one of 4 variables The number of frequent words in *overall objectives* section is 220, 210 for *scientific foundations*, 404 for *dissemination* and 547 for *new results* sections. Each cell on a data table gives the frequency of a word for the considered activity report section and research team.

Then, 4 relational data tables have been obtained from the 4 feature data tables through a dissimilarity measure derived from the affinity coefficient (Barcelar-Nicolau (2000)). We assume that each individual is described by one set-valued variable ("presentation", etc.) which has m_j modalities (or categories) $\{1, \ldots, m\}$. An individual e_i is described by $\mathbf{x}_i = (n_{i1}, \ldots, n_{im})$ where n_{ij} is the frequency of modality j. The dissimilarity between a pair of individuals e_i and $e_{i'}$ is given by:

$$d(\mathbf{x}_i, \mathbf{x}'_i) = 1 - \sum_{j=1}^{m} \sqrt{\frac{n_{ij}}{n_{i\bullet}} \frac{n_{i'j}}{n_{i'\bullet}}} \quad where \quad n_{i\bullet} = \sum_{j=1}^{m} n_{ij}.$$

All these relational data tables were normalized according to their overall dispersion (Chavent (2005)) to have the same dispersion. This means that each dissimilarity $d(\mathbf{x}_i, \mathbf{x}'_i)$ in a given relation data table has been normalized as $\frac{d(\mathbf{x}_i, \mathbf{x}'_i)}{T}$ where $T = \sum_{i=1}^{n} d(e_i, g)$ is the overall dispersion and $g = e_l \in E = \{e_1, \ldots, e_n\}$ is the overall prototype, which is computed according to $l = argmin_{1 \leq h \leq n} \sum_{i=1}^{n} d(e_i, e_h)$.

3.1 Results

The clustering algorithm has been performed simultaneously on these 4 relational data tables ("presentation", "foundation", "dissemination" and "bibliography") in order to obtain a partition in $K \in \{1, \ldots, 15\}$. For a fixed number of clusters K, the clustering algorithm is run 100 times and the best result according to the adequacy criterion is selected.

In order to determine the number of clusters, we used the approach described by Da Silva (2009), which consists on the choice of the peaks on the graph of the "second order differences" of the clustering criterion (equation (1)): $J^{(K-1)} + J^{(K+1)} - 2J^{(K)}$, $K = 2, \ldots, 14$. According to this approach, we fixed the number of clusters in 4 and 9.

The 4-cluster and the 9-cluster partitions obtained with this clustering algorithm were compared with the INRIA research team categorization 5-class partition known a priori. The 5-class a priori categorization is as follows: "Applied Mathematics, Computation and Simulation (M)", "Algorithmics, Programming, Software and Architecture (A)", "Networks, Systems and Services, Distributed Computing (N) ", "Perception, Cognition, Interaction (P)" and "Computational Sciences for Biology, Medicine and the Environment (C)". These 5 categories are themselves divided into several sub-categories. In many points we retrieve in the 9-cluster partition the categorization done a priori by INRIA. For example the sub-category "Networks and Telecommunications" of N fits exactly in one cluster. The two sub-categories "Distributed Systems and Services" and "Distributed and High Performance Computing" are merged into a unique cluster, indicating that from the language used point of view the distinction between these two categories is artificial. Some teams have also migrate. For example it seems that the language used in Cryptography (that is part of A in the a priori categorization) is closer to the language used in math (M). Looking at the 4-cluster partition, some migrations are also clearly detected, which have a political sense, in particular when the concerned team is found in a cluster corresponding to the "right" category in a former categorization done by INRIA. Finally, is seems that teams in the C category share the same language as teams in M or in P stressing the fact that in the activity report the weight of the scientific foundations is important, and this fact showing up in both partitions is however clearer in the 4-cluster partition than in the 9-cluster one.

4 Concluding Remarks

This paper introduced a clustering algorithm that is able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilaritity matrices. These matrices could have been generated using different sets of variables and dissimilarity functions. This algorithm provides a partition and a prototype for each cluster as well as a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that

measures the fit between clusters and their representatives. These relevance weights change at each algorithm iteration and are different from one cluster to another. The usefulness of this algorithm was illustrated comparing the categorization of INRIA research teams given by the clustering algorithm with the a priori expert categorization given by INRIA. The clustering algorithm was able to retrieve the a priori categorization, the observed minor divergences being explained by political choices of INRIA.

References

BACELAR-NICOLAU, H. (2000): The affinity coefficient. In: H.H Bock and E. Diday (Eds.): *Analysis of Symbolic Data*. Springer, Heidelberg, 160–165.

BOCK, H.H. and DIDAY, E. (2000): *Analysis of Symbolic Data*. Springer, Heidelberg.

CHAVENT, M. (2005): Normalized k-means clustering of hyper-rectangles. In: *Proceedings of the XIth International Symposium of Applied Stochastic Models and Data Analysis (ASMDA 2005), Brest, France*, 670–677.

DE CARVALHO, F. A. T. and LECHEVALLIER, Y. and VERDE, R. (2008): Clustering methods in symbolic data analysis. In: Edwin Diday; Monique Noirhomme-Fraiture. (Eds.): *Symbolic Data Analysis and the SODAS Software*. Wiley-Interscience, San Francisco, 181–204.

DE CARVALHO, F. A. T. and CSERNEL, M. and LECHEVALLIER, Y. (2009): Clustering constrained symbolic data *Pattern Recognition Letters, 30 (11), 1037–1045*.

DE CARVALHO, F. A. T., LECHEVALLIER, Y. (2009): Partitional clustering algorithms for symbolic interval data based on single adaptive distances. *Pattern Recognition, 42 (7), 1223–1236*.

DA SILVA, A. (2009): Analyse de données évolutives: application aux données d'usage Web. *Thèse de Doctorat. Université Paris-IX Dauphine*.

DIDAY, E., GOVAERT, G. (1977): Classification automatique avec distances adaptatives. *R.A.I.R.O. Informatique Computer Science 11 (4), 329–349*.

FRIGUI, H., HWANG, C. and RHEE, F. C. (2007): Clustering and aggregation of relational data with applications to image database categorization. *Pattern Recog., 40 (11), 3053–3068*.

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

HATHAWAY, R. J., DAVENPORT, J. W. and BEZDEK, J. C. (1989): Relational duals of the c-means algorithms. *Pattern Recog., 22, 205–212*.

JAIN, A.K., MURTY, M.N. and FLYN, P.J. (1999): Data clustering: A review. *ACM Comput. Surv. 31 (3), 264–323*.

KAUFMAN, L. and ROUSSEEUW, P. J. (1990): *Finding Groups in Data*. NewYork: Wiley.

LECHEVALLIER, Y. (1974): Optimisation de quelques critères en classification automatique et application a l'étude des modifications des protéines sériques en pathologie clinique. *Thèse de 3eme cycle. Université Paris-VI*.

LECLERC, B. and CUCUMEL, G. (1987): Concensus en classification : une revue bibliographique. *Mathématique et sciences humaines, 100, 109–128*

PEDRYCZ, W. (2002): Collaborative fuzzy clustering. *Pattern Recognition Lett., 23, 675–686*.