Multi-view Clustering on Relational Data

Francisco de A.T. de Carvalho, Yves Lechevallier,
Thierry Despeyroux, and Filipe M. de Melo

Abstract. Clustering is a popular task in knowledge discovery. In this chapter we illustrate this fact with a new clustering algorithm that is able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. The advantages of this algorithm are threefold: it uses any dissimilarities between objects, it automatically ponderates the impact of each dissimilarity matrice and it provides interpretation tools. We illustrate the usefulness of this clustering method with two experiments. The first one uses a data set concerning handwritten numbers (digitized pictures) that must be recognized. The second uses a set of reports for which we have an expert classification given *a priori* so we can compare this classification with the one obtained automatically.

1 Introduction

Clustering is a popular task in knowledge discovery and it is applied in various fields including data mining, pattern recognition, computer vision, etc. [Gordon, 1999; Jain et al., 1999]. Clustering methods aim 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. A precise definition of the dissimilarity between objects is thus very important.

Some of the clustering techniques are called partitioning methods. Partitioning methods seek to obtain a single partition of the input data into a given number of clusters. Often, such methods look for a partition that optimizes (locally) an adequacy criterion function.

Francisco de A.T. de Carvalho · Filipe M. de Melo
Centro de Informatica -CIn/UFPE - Av. Prof. Luiz Freire, s/n -Cidade Universitaria - CEP 50740-540, Recife-PE, Brazil
e-mail: {fatc,fmm}@cin.ufpe.br

Yves Lechevallier · Thierry Despeyroux
INRIA, Paris-Rocquencourt, 78153 Le Chesnay Cedex, France
e-mail: {Yves.Lechevallier,Thierry.Despeyroux}@inria.fr

F. Guillet et al. (eds.), *Advances in Knowledge Discovery and Management,*
Studies in Computational Intelligence 527,
DOI: 10.1007/978-3-319-02999-3_3,

Two usual representations of the objects upon which clustering can be based are (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 feature data. When each object is described by a vector of sets of categories, intervals or weight histograms, the set of vectors describing the objects can be considered as symbolic (feature) data, according to the Symbolic Data Analysis (SDA) approach[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.

Many methods and algorithms have been proposed in order to cluster (usual or symbolic) feature data[Gordon, 1999; Jain et al., 1999; Kaufman and Rousseeuw, 1990]. However, few clustering models have been proposed for relational data. [Frigui et al., 2007] observed that several applications, as content-based image retrieval, would benefit strongly from clustering methods for relational data. In SDA, many effective dissimilarity measures proposed to the comparison of symbolic data are not differentiable with respect to the prototype parameters and thus, they could not be used in clustering methods for symbolic feature data based on objective functions. For example, in order to cluster constrained symbolic data, [De Carvalho et al., 2009] used the dynamic clustering algorithm for relational data [De Carvalho et al., 2012]. The constraints were taken into account during the computation of a suitable dissimilarity function between the symbolic feature data in order to obtain a relational data set.

In this paper we will focus on relational data. When the representation of an object is not unique, we speak of multi-view data. Multi-view data can be found in many domains such as bioinformatics, marketing, etc. [Cleuziou et al., 2009], and in structural documents. For example, in XML documents with many sections, each of these sections can be interpreted as a different view.

This paper presents a clustering algorithm that is a variant of the one given in [De Carvalho et al., 2012], that is able to partition objects taking simultaneously into account their relational descriptions given by multiple dissimilarity 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].

The 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 central 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 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 RFCM [Hathaway et al., 1989] and FANNY [Kaufman and Rousseeuw, 1990].

The clustering algorithm proposed 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. The method is based on the dynamic hard clustering algorithm for relational data [Lechevallier, 1974; De Carvalho et al., 2008, 2009] and on adaptive distances [Diday and Govaert, 1977; De Carvalho and Lechevallier, 2009]. One of the advantage of the algorithm is that it provides interpretation tools that help in understanding the result.

In order to demonstrate the usefulness of this new clustering algorithm, we apply it on two different applications. The first one concerns the clustering of handwritten digits (0 to 9) that are scanned in binary pictures. The data that are used are available from the "UCI machine learning repository". The second one uses the example given [De Carvalho et al., 2010] taking a document data base for which we have an expert categorization.

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 dissimilarity matrices.

In this new version, the prototype is no more defined as an object, but as a vector of objects from E. For each matrix there is one associated object.

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 $g_k = (g_{k1}, \ldots, g_{kp})$ is the prototype vector of cluster C_k, where each component belongs to the set E, *i.e.* , $\mathbf{g}_k \in E^p \, (k = 1, \ldots, K)$, with $g_{kj} \in E \, (j = 1, \ldots, p)$.

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 vector $\mathbf{g}_k \in E^p$ representing the cluster C_k in P and a weight for each dissimilarity matrix such that the adequacy criterion J is locally optimized. The adequacy criterion is defined as

$$J = \sum_{k=1}^{K} \sum_{e_i \in C_k} d_{\lambda_k}(e_i, \mathbf{g}_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} d_j(e_i, g_{kj}) \tag{1}$$

in which

$$d_{\lambda_k}(e_i, \mathbf{g}_k) = \sum_{j=1}^{p} \lambda_{kj} d_j(e_i, g_{kj}) \tag{2}$$

is the dissimilarity between an example $e_i \in C_k$ and the cluster prototype $g_k \in E^p$ parameterized by the relevance weight vector $\lambda_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kp})$ where λ_{kj} is the weight for the dissimilarity matrix $\mathbf{D}_j$ for the cluster C_k, and $d_j(e_i, g_{kj}))$ is the local dissimilarity d_j between an example $e_i \in C_k$ and the cluster prototype $g_{kj} \in E$.

Our clustering algorithm alternates the three following steps:

- ***Step 1: Definition of the Best Prototype Vectors***
  In this step, the partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the relevance weight matrix λ are fixed.
  For each cluster C_k we compute the prototype vector $\mathbf{g}_k$ which minimizes the clustering criterion J. This vector contains the components g_{kj}, objects of E, that are obtained using :

$$l = argmin_{1 \leq h \leq n} \sum_{e_i \in C_k} \lambda_{kj} d_j(e_i, e_h) \tag{3}$$

- ***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 $\mathbf{g} = (\mathbf{g}_1, \ldots, \mathbf{g}_K)$ are fixed.
  The element j of the relevance weight vector $\lambda_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kp})$, which minimizes the clustering criterion J under $\lambda_{kj} > 0$ et $\prod_{j=1}^{p} \lambda_{kj} = 1$, is calculated by the following expression:

$$\lambda_{kj} = \frac{\left\{\prod_{h=1}^{p} \left[\sum_{e_i \in C_k} d_h(e_i, g_{kh})\right]\right\}^{\frac{1}{p}}}{\left[\sum_{e_i \in C_k} d_j(e_i, g_{kj})\right]} \tag{4}$$

  Remark The more the examples in the cluster C_k are close to the component g_{kj} of the prototype $\mathbf{g}_k$ considering the matrix of dissimilarity $\mathbf{D}_j$, the higher is the value of the weight λ_{kj}.
- ***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 λ are fixed.
  The cluster C_k is updated according to the following allocation rule:

$$C_k = \{e_i \in E : d_{\lambda_k}(e_i, \mathbf{g}_k) < d_{\lambda_h}(e_i, \mathbf{g}_h) \forall h \neq k\} \tag{5}$$

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

It's easy to demonstrate that each previous 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 $J(P, \lambda, \mathbf{g})$ reaches a stationary value representing a local minimum.

Comparing with the initial algorithm [De Carvalho et al., 2012], using a vector prototype allows to optimize the choice of the prototype and of the weight locally, by class and by dissimilarity matrix. The clustering criterion J is decomposed according to the dissimilarity matrices, and according to classes and dissimilarity matrices simultaneously, allowing to interpret the classes against matrices.

3 Interpreting Clusters and Partition

Let T be the criterion corresponding to the criterion J applied to a clustering in a unique class of E. The tools that help to interpret the classes and the partition are based on the decomposition of the criterion T in two parts. The first one corresponds to the dispersion intra-classes W (W corresponds to the clustering criterion J) and the second one corresponds to the dispersion inter-classes B. We use the approach given by [Chavent et al., 2006] that permits to compute this decomposition even if computing the inter-classes dispersion B is impossible (see [De Carvalho et al., 2012]).

Let $P = (C_1, \ldots, C_K)$ the final partition $E = \{e_1, \ldots, e_n\}$ in K classes. Let $\mathbf{g}_k$ the prototype and λ_k the vector of relevance weight of $C_k\,(k = 1, \ldots, K)$. Suppose also that the global prototype is the vector $\mathbf{g} = (g_1, \ldots, g_p)$ where $g_j \in E\,(j = 1, \ldots, p)$.

The global dispersion T of the partition $P = (C_1, \ldots, C_K)$ is defined by

$$T = \sum_{k=1}^{K} \sum_{e_i \in C_k} d_{\lambda_k}(e_i, \mathbf{g}) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} d_j(e_i, g_j) \tag{6}$$

where the global prototype $\mathbf{g}$, that minimizes the global dispersion T, is composed of $g_j = e_l \in E$ computed using :

$$l = argmin_{1 \leq h \leq n} \sum_{k=1}^{K} \sum_{e_i \in C_k} \lambda_{kj} d_j(e_i, e_h) \tag{7}$$

The global dispersion is decomposed in

a) $T = \sum_{k=1}^{K} T_k$ with $T_k = \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} d_j(e_i, g_j)$;
b) $T = \sum_{k=1}^{K} \sum_{j=1}^{p} T_{kj}$ with $T_{kj} = \sum_{e_i \in C_k} \lambda_{kj} d_j(e_i, g_j)$;
c) $T = \sum_{j=1}^{p} T_j$ with $T_j = \sum_{k=1}^{K} \sum_{e_i \in C_k} \lambda_{kj} d_j(e_i, g_j)$

The dispersion intra-classes W is given by the clustering criterion J(see 1):

a) $J = \sum_{k=1}^{K} J_k$ with $J_k = \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} d_j(e_i, g_{kj})$;
b) $J = \sum_{j=1}^{p} J_j$ with $J_j = \sum_{k=1}^{K} \sum_{e_i \in C_k} \lambda_{kj} d_j(e_i, g_{kj})$;
c) $J = \sum_{k=1}^{K} \sum_{j=1}^{p} J_{kj}$ with $J_{kj} = \sum_{e_i \in C_k} \lambda_{kj} d_j(e_i, g_{kj})$

One can easily show that

i) $T \geq J$;
ii) $T_k \geq J_k\,(k = 1, \ldots, K)$;

iii)$T_j \geq J_j\,(j = 1,\ldots,p)$;
iv)$T_{kj} \geq J_{kj}\,(k = 1,\ldots,K; j = 1,\ldots,p)$.

Given the global dispersion, the intra-classes dispersion and their decomposition, the indexes for the help to interpretation of classes and partition introduced by [Chavent et al., 2006] can be easily be adapted to the new algorithm.

The global quality of the final partition is $Q(P) = 1 - \frac{J}{T}$. An index $Q(P)$ close to 1 indicates a partition of better quality (more homogeneous classes).

The global quality of the final partition according to each dissimilarity matrix is given by $Q_j(P) = 1 - \frac{J_j}{T_j}$. A value for $Q_j(P)$ close to 1 indicates a good quality of the partition P according to the dissimilarity matrix $\mathbf{D}_j$. The comparison between $Q_j(P)$ and $Q(P)$ shows that the discriminant power of the dissimilarity matrix $\mathbf{D}_j$ is greater than the average discriminant power of all the dissimilarity matrices.

4 Applications

To illustrate the usefulness of our new algorithm, we use it on two different data sets. The first one is a set of digitized handwritten digits, the second a set of scientific activity reports.

4.1 Handwritten Digits Dataset

Our first example concerns the clustering of "multiple features" data available in the "UCI machine learning repository". This set of data contains handwritten digits (0 to 9) that are scanned in binary pictures. The 2000 handwritten digits (objects) are described by 649 numerical variables. These variables are partitioned in 6 different sets (views):

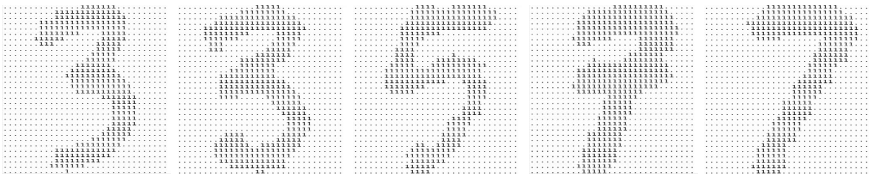

Fig. 1 Digitized handwritten digit '3', '3', '5', '7', '7'

- 76 Fourier coefficients describing the shape of the digits
- 64 Karhunen-Love coefficients
- 240 pixels average in 2 x 3 windows
- 47 Zernike moments
- 6 Morphological characteristics

Theses data are structured in 10 *a priori* classes containing 200 objects, each class corresponding to one digit.

We first consider 7 data tables: one in which the objects are described by all the 649 variables (table "mfeat") and 6 other tables in which the objects are described by one of the 6 different "views", each "view" having respectively 76 (table "mfeatFou"), 216 (table "mfeatFac"), 64 (table "mfeatKar"), 240 (table "mfeatPix"), 47 (table "mfeatZer"), and 6 (table "mfeatMor") variables.

Then 7 relational data tables are obtained from theses 7 data tables using the Euclidean distance. All these tables are then normalized according to their global dispersion [Chavent, 2005] to have the same dispersion. This means that each dissimilarity $d(\mathbf{x}_i,\mathbf{x}'_i)$ in a given relational 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 global dispersion and $g=e_l\in E=\{e_1,\ldots,e_n\}$ is the global prototype, which is computed according to $l=argmin_{1\leq h\leq n}\sum_{i=1}^{n} d(e_i,e_h)$.

Our clustering algorithm has been performed first on the relational data table "mfeat" and then simultaneously in the 6 relational data tables "mfeatFou", "mfeatFac", "mfeatKar", "mfeatPix", "mfeatZer", and "mfeatMor", corresponding to the 6 different "views" to obtain a partition in 10 clusters. The clustering algorithm is run 100 times and the best result according to the adequacy criterion J is selected. Our goal is to compare the partition obtain by our clustering algorithm with the partition in 10 clusters given *a priori*. The comparison criterion that we have chosen are the overall (global) error rate of classification (*OERC*)[Breiman et al., 1984], the corrected Rand index (*CR*)[Hubert and Arabie, 1985], and the F-measure[Van Rijisbergen, 1976].

Results

The values of the *CR*, F-measure and *OERC* indexes, obtained from the final partition computed by our clustering algorithm applied to the relational data table "mfeat", are respectively 0.518, 0.674, and 37.75%.

The values of the same indexes obtained from the final partition computed by our clustering algorithm applied simultaneously to the 6 relational data tables corresponding to the 6 different "views" are respectively 0.762, 0.879 et 12.10%. The table 1 shows the relevance weight matrix of the relational data tables in the clusters.

The table 2 shows the confusion matrix into 10 cluster computed for the final partition.

We can see that the dissimilarity matrix "mfeatMor" is the most pertinent one for defining all the clusters. We also see that the dissimilarity matrix "mfeatFac" has a relevance weight as important that the one for the dissimilarity matrix "mfeatMor" for the cluster 3.

The global quality of the final partition is $Q(P)=1-\frac{J}{T}=0.919$. Closer is the index $Q(P)$ to 1 better is the partition quality (with more homogeneous clusters).

The global quality of the final partition relative to each dissimilarity matrix $Q_j(P)=1-\frac{J_j}{T_j}$ $(j=1,\ldots,6)$ is shown in Table 3. A value of $Q_j(P)$ close to 1 is an indication of a good quality of the partition P relative to the dissimilarity matrix $\mathbf{D}_j$. Comparing $Q_j(P)$ with $Q(P)$ shows that the discriminant power of the

Table 1 Relevance Weight Matrix of the Relational Data Tables in the Clusters

	Relevance Weight of Dissimilarity Matrices					
Clusters	1-mfeatMor	2-mfeatZer	3-mfeatPix	4-mfeatKar	5-mfeatFou	6-mfeatFac
1	6.728	0.713	0.562	0.595	0.533	1.165
2	12.543	0.615	0.515	0.546	0.434	1.059
3	2.891	0.919	0.612	0.646	0.454	2.091
4	3.412	1.083	0.526	0.562	0.513	1.778
5	5.318	0.828	0.573	0.640	0.454	1.361
6	135.631	0.338	0.236	0.252	0.318	1.147
7	54.559	0.484	0.270	0.290	0.393	1.223
8	5.276	0.794	0.547	0.596	0.421	1.733
9	8.163	0.749	0.504	0.559	0.383	1.505
10	8367.671	0.199	0.124	0.134	0.097	0.363

Table 2 Confusion Matrix

	Clusters (Handwritten Digits)									
Clusters	'7'	'1'	'5'	'2'	'4'	'0'	'8'	'3'	'6'	'9'
1	193	15	4	16	6	0	0	30	2	0
2	1	170	0	0	4	0	3	1	5	0
3	0	0	149	1	0	2	0	27	0	0
4	1	0	6	178	0	0	1	3	0	0
5	1	2	1	1	183	0	1	2	3	0
6	0	0	0	0	0	188	18	0	0	0
7	0	11	0	0	0	9	174	0	3	0
8	4	0	40	3	1	1	2	137	1	0
9	0	2	0	1	6	0	1	0	186	0
10	0	0	0	0	0	0	0	0	0	200

Table 3 Global Quality of the Partition P relatively to each Dissimilarity Matrix (%)

	Dissimilarity Matrices					
	1-mfeatMor	2-mfeatZer	3-mfeatPix	4-mfeatKar	5-mfeatFou	6-mfeatFac
$Q_j(P)$	98.44	47.28	43.58	47.16	35.09	65.69

dissimilarity matrix "mfeatMor" is greater that the average discriminant power of all the dissimilarity matrices.

Table 4 shows the heterogeneity index $J(k) = \frac{J_k}{J}$ and the quality index $Q(k) = 1 - \frac{J_k}{T_k}$ for each cluster $k = 1, \ldots, 10$. One can sea, for example, that the cluster 10 (digit '9') is more homogeneous while the cluster 6 (digit '0') is of best quality.

Table 5 shows the index $Q_j(k) = 1 - \frac{J_{kj}}{T_{kj}}$, that gives the quality of the cluster $C_k\,(k = 1, \ldots, 10)$ in the dissimilarity matrix $\mathbf{D}_j\,(j = 1, \ldots, 6)$. Closer to 1 is the value of this index, better is the quality of this cluster in this dissimilarity matrix. While $Q(P)$ is a global index, $Q_j(k)$ is a local one for a given cluster and a given dissimilarity matrix. More, the comparison between the indices $Q_j(k)$ and $Q(k)$

Table 4 Heterogeneity Index and Quality Index of a Cluster(%)

	Cluster k									
	1	2	3	4	5	6	7	8	9	10
Cardinal	266	184	179	189	194	206	197	189	196	200
$J(k)$	17.52	10.20	12.67	10.34	14.07	3.75	6.14	12.38	10.39	2.48
$Q(k)$	88.63	84.80	93.36	93.42	89.42	97.70	93.70	93.43	84.18	88.73

Table 5 Quality of Clusters in the Dissimilarity Matrices (%)

	Dissimilarity Matrix					
Classes	1-mfeatMor	2-mfeatZer	3-mfeatPix	4-mfeatKar	5-mfeatFou	6-mfeatFac
1	97.69	38.36	49.84	52.10	39.84	50.57
2	96.80	32.34	45.94	46.65	30.16	34.07
3	98.79	23.15	34.46	39.43	35.03	39.12
4	98.76	61.18	47.37	52.09	41.66	53.72
5	97.93	13.31	42.37	46.97	20.34	56.26
6	99.58	81.82	60.47	65.07	69.41	77.24
7	98.85	42.33	22.75	26.86	41.98	51.96
8	98.81	25.05	30.37	34.73	20.25	23.20
9	96.50	00.00	45.99	50.50	18.19	68.99
10	03.77	91.28	55.00	53.25	42.25	97.11

gives the dissimilarity matrices that characterize the cluster k. For example, the dissimilarity matrix "mfeatMor" is characteristic of the clusters 1 to 9, while the matrices "mfeatZer" and "mfeatFac" are characteristic of the cluster 10 (digit '9').

4.2 Document Data Base Categorization

As a second application of our algorithm, we use it to categorize a document data base. The document data base is a collection of scientific activity reports produced by each INRIA (The French National Institute for Research in Computer Science and Control) research team in 2007. Theses deliverables are send to the French parliament for public funding assessing and are also made available to its industrial and research partners.

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. Figure 2 gives a view of this categorization. The choice of the themes and the allocation of the teams are mostly related to strategic objectives and scientific closeness between existing teams, however some geographical constraints, such as the desire for a theme to be representative of most INRIA centers are taken into account. Our aim is to compare the *a priori* categorization given by INRIA of the reports with that induced by the clustering algorithm here proposed.

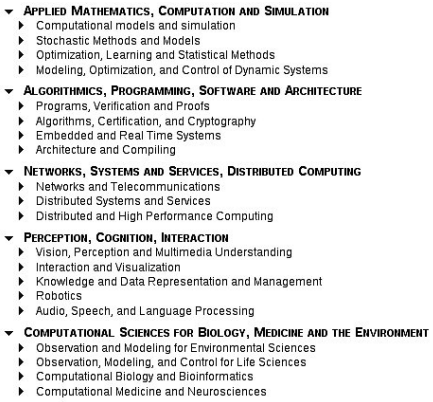

Fig. 2 INRIA research categorization

Each report (RA) is written in English and using LaTeX, it is automatically translated into XML, then to HTML and 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 consider activity reports from 164 INRIA research teams in 2007. The XML version of these documents contains 173 files, a total of 613 000 lines, more than 40 Mbytes of data. Figure 3 gives an example of an activity report summary.

Fig. 3 Example of an activity report summary

In these activity reports, four 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, *scientific foundations* provides the scientific background followed 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, while the *new results* includes the principal results obtained during that year.

In a first step all the texts are preprocessed. Stop-words are removed, and the texts are annotated with part-of-speech and lemma information using treetagger.

Four feature data tables are build, each with 164 objects (the research teams) described by the frequent words (categories) present in one of the four sections. The numbers of frequent words in the sections *overall objectives*, *scientific foundations*, *dissemination*, and *new results* are respectively 220, 210, 404, and 547. Each cell on a data table gives the frequency of a word for the considered activity report section and research team.

Then, four relational data tables have been obtained from the 4 feature data tables through a dissimilarity measure derived from the affinity coefficient [Bacelar-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 global dispersion [Chavent, 2005]: each dissimilarity $d(\mathbf{x}_i,\mathbf{x}'_i)$ in a 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 global dispersion and $g = e_l \in E = \{e_1,\ldots,e_n\}$ is the global prototype, which is computed according to $l = argmin_{1\leq h\leq n} \sum_{i=1}^{n} d(e_i,e_h)$.

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.

Determining the appropriate number of clusters in a partition is a classical problem but no good solution exists[Milligan and Cooper, 1985]. To choose the right number of cluster, our strategy is those of the SPAD software[1]. It consists in choosing the best couple (inter-classes inertia, number of classes). The decrease of the number of classes increases the intra-classes inertia, so to get a partition with a

[1] `http://eng.spad.eu/`

good quality we must identify an important jump of the index. This peak can be found using the second order differences of the clustering criterion [Da Silva, 2009; Charrad et al., 2010].

The discrete first derivative of J according to k is $Df(x) = (f(x+h) - f(x))/h$ and the second one is $D2f(x) = (f(x+h) - 2f(x) + f(x-h))/h^2$. When h tends to 0 this is equivalent to the usual derivative.

A partition in 4 clusters is chosen because at this spot the second derivative is maximal (see Fig. 4). A partition in 11 clusters would also be possible as it is a local maximum.

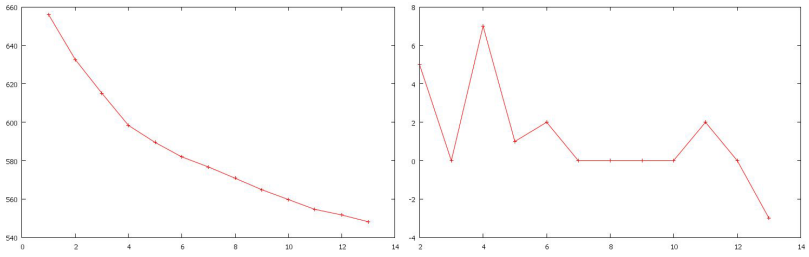

Fig. 4 *J* criteria, second derivative

The 4-clusters partition obtained with the here proposed algorithm was compared with the *a priori* 5-class partition of the INRIA in 2008. INRIA *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)".

The activity reports refer to year 2007, and the expert classification by INRIA has been done in 2008. Between these two years some research teams have been closed and others has evolved. For this reason, only 154 activity reports has been used in the comparison between our automatic clustering and the expert classification done by INRIA.

Table 6 shows that the 4-clusters partition obtained with the clustering algorithm is quite consistent with the *a priori* 5-class categorization, except for the M and C class.

Category 5, Computational Sciences for Biology, Medicine and the Environment, is artificial and is distributed (considering the vocabulary that is used) in two clusters, depending on the fact that the subject is more mathematical or more cognitive. Thus, the cluster C3 could be labelled "Simulation/control/modelisation", and the cluster C4 "Data processing".

The relevance weight matrix for the for variables (sections) used in the activity reports is shown in table 7.

Table 6 Distribution table of 154 reports (2007) in 5 *a priori* categories (2008) (rows) in the 4 clusters (columns)

	C1	C2	C3	C4
M - Applied Mathematics, Computation and Simulation	1	1	**20**	6
A - Algorithmics, Programming, Software and Architecture	**17**	3	1	9
N - Networks, Systems and Services, Distributed Computing	1	**28**	2	2
P - Perception, Cognition, Interaction	5	1	2	**35**
C - Computational Sciences for Biology, Medicine and the Environment	0	0	**11**	**9**

Table 7 Relevance Weight Matrix of the Dissimilarity matrices in the classes

	Relevance Weight of Dissimilarity Matrices			
Clusters	overall objectives	scientific foundations	new results	dissemination
1	0.969026	0.979387	1.000909	1.052727
2	1.019705	0.934093	1.073774	0.977738
3	0.966223	1.068582	1.073115	0.902545
4	0.976156	0.993158	1.026519	1.004837

The values of the *CR*, F-measure and *OERC* indexes, obtained from the final partition computed by our clustering algorithm are respectively 0.360, 0.657 and 27.92%.

5 Conclusion

This paper introduced a new 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 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 are automatically computed at each algorithm iteration and are different from one cluster to another. We also provide tools for the interpretation of the clusters and the partition provided by the algorithm.

Two experiments demonstrate the usefulness of this clustering method.

References

[Bacelar-Nicolau, 2000] Bacelar-Nicolau, H.: The affinity coefficient. In: Bock, H.H., Diday, E. (eds.) Analysis of Symbolic Data, pp. 160–165. Springer, Heidelberg (2000)

[Bock and Diday, 2000] Bock, H., Diday, E.: Analysis of Symbolic Data. Springer, Heidelberg (2000)

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman and Hall/CRC, Boca Raton (1984)

[Charrad et al., 2010] Charrad, M., Lechevallier, Y., Ahmed, M.B., Saporta, G.: On the number of clusters in block clustering algorithms. In: Guesgen, H.W., Murray, R.C. (eds.) FLAIRS Conference. AAAI Press (2010)

[Chavent, 2005] Chavent, M.: 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, pp. 670–677 (2005)

[Chavent et al., 2006] Chavent, M., De Carvalho, F.A.T., Lechevallier, Y., Verde, R.: New clustering methods for interval data. Computational Statistics 21(2), 211–229 (2006)

[Cleuziou et al., 2009] Cleuziou, G., Exbrayat, M., Martin, L., Sublemontier, J.-H.: Cofkm: A centralized method for multiple-view clustering. In: ICDM 2009 Ninth IEEE International Conference on Data Mining, Miami, USA, pp. 752–757 (2009)

[Da Silva, 2009] Da Silva, A.: Analyse de données évolutives: application aux données d'usage Web. PhD thesis, Université Paris-IX Dauphine (2009)

[De Carvalho et al., 2009] De Carvalho, F.A.T., Csernel, M., Lechevallier, Y.: Clustering constrained symbolic data. Pattern Recognition Letters 30(11), 1037–1045 (2009)

[De Carvalho and Lechevallier, 2009] De Carvalho, F.A.T., Lechevallier, Y.: Partitional clustering algorithms for symbolic interval data based on single adaptive distances. Pattern Recognition 42(7), 1223–1236 (2009)

[De Carvalho et al., 2012] De Carvalho, F.A.T., Lechevallier, Y., De Melo, F.M.: Partitioning hard clustering algorithms based on multiple dissimilarity matrices. Pattern Recognition 45(1), 447–464 (2012)

[De Carvalho et al., 2008] De Carvalho, F.A.T., Lechevallier, Y., Verde, R.: Clustering methods in symbolic data analysis. In: Diday, E., Noirhomme-Fraiture, M. (eds.) Symbolic Data Analysis and the SODAS Software, pp. 181–204. Wiley-Interscience, San Francisco (2008)

[De Carvalho et al., 2010] De Carvalho, F.A.T., Despeyroux, T., De Melo, F.M., Lechevallier, Y.: Utilisation de matrices de dissimilarit multiples pour la classification de documents. In: EGC-M 2010, Extraction et Gestion des Connaissances, Alger, Algérie, pp. 1–10 (2010)

[Diday and Govaert, 1977] Diday, E., Govaert, G.: Classification automatique avec distances adaptatives. R.A.I.R.O. Informatique Computer Science 11(4), 329–349 (1977)

[Frigui et al., 2007] Frigui, H., Hwang, C., Rhee, F.C.: Clustering and aggregation of relational data with applications to image database categorization. Pattern Recognition 40(11), 3053–3068 (2007)

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

[Hathaway et al., 1989] Hathaway, R.J., Davenport, J.W., Bezdek, J.C.: Relational duals of the c-means algorithms. Pattern Recognition 22, 205–212 (1989)

[Hubert and Arabie, 1985] Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2(1), 193–218 (1985)

[Jain et al., 1999] Jain, A., Murty, M., Flyn, P.: Data clustering: A review. ACM Comput. Surv. 31(3), 264–323 (1999)

[Kaufman and Rousseeuw, 1990] Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. Wiley, New York (1990)

[Lechevallier, 1974] Lechevallier, Y.: Optimisation de quelques critères en classification automatique et application a l'étude des modifications des protéines sériques en pathologie clinique. PhD thesis, Université Paris-VI (1974)

[Leclerc and Cucumel, 1987] Leclerc, B., Cucumel, G.: Concensus en classification: une revue bibliographique. Mathématique et Sciences Humaines 100, 109–128 (1987)
[Milligan and Cooper, 1985] Milligan, G.W., Cooper, M.C.: An examination of procedures for determining the number of clusters in a data set. Psychometrika 50, 159–179 (1985)
[Pedrycz, 2002] Pedrycz, W.: Collaborative fuzzy clustering. Pattern Recognition Lett. 23, 675–686 (2002)
[van Rijisbergen, 1976] van Rijisbergen, C.J.: Information retrieval. Butterworth-Heinemann, London (1976)