Utilisation de matrices de dissimilarité multiples pour la classification de documents

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

Utilisation de matrices de dissimilarité multiples pour la classification de documents

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

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

Résumé. Cet article introduit l'algorithme de classification donné dans (De Carvalho et Lechevallier, 2007) capable de partitionner des objets en prenant en compte de manière simultanée plusieurs matrices de dissimilarité qui les décrivent. Ces matrices peuvent avoir été générées en utilisant différents ensembles de variables et une fonction de dissimilarité unique, un ensemble de variables donné et différentes fonctions de dissimilarité ou bien différents ensembles de variables et de fontions de dissimilarité. Cette méthode, basée sur l'algorithme de nuées dynamiques est conçu pour fournir une partition et un prototype pour chaque classe tout en découvrant une pondération pertinante pour chaque matrice de dissimilarité en optimisant un critère d'adéquation entre les classes et leurs représentants. Ces pondérations changent à chaque itération de l'algorithme et sont différentes pour chacune des classes.

Ce papier se focalise sur une expérience utilisant un ensemble de documents, dont nous connaisons une classification donnée *a priori* par des experts servant de référence, et montre l'utilité de cette méthode de partitionnement.

1 Introduction

La classification est une activité courante en extraction de connaissances, que ce soit dans les domaines de fouille de données, de reconnaissance de motifs, de vision par ordinateur, etc. (Gordon, 1999; Jain et al., 1999). Le but de la classification est d'organiser un ensemble d'objets en sous-ensembles appelés clusters de telle façon que les éléments d'un même cluster aient un haut niveau de similarité alors qu'à l'inverse deux éléments de deux clusters différents aient un haut niveau de dissimilarité. Les deux techniques de classification les plus populaires sont les méthodes hiérarchiques et les méthodes de partitionnement. Le but des méthodes de partitionnement est d'obtenir une partition unique des données d'entrée en un nombre fixe de clusters. Beaucoup de ces méthodes recherchent une partition qui optimise (localement) une fonction-critère d'adéquation.

Il y a deux représentations courantes des objets sur lesquels travaillent les méthodes de classification : les tableaux de données et les tableaux de proximités (en anglais relational data).

Dans le premier cas, si les données sont décrites à l'aide de vecteurs de valeurs quantitatives ou qualitatives, on parle de données non symboliques (feature data), si les données sont décrites à l'aides de vecteurs de valeurs complexes tels que des intervalles ou des histogrammes on parle de données symboliques (symbolic feature data) (Bock et Diday, 2000). Dans le cas de données relationnelles, les objets sont décrits par une relation, dont le cas le plus courant est représenté par une matrice de dissimilarité qui stocke une mesure de dissimilarité deux à deux entre objets (souvent une distance). Il est à noter que si beaucoup de méthodes ont été étudiées concernant les tableaux de données (symboliques ou non), peu de modèles existent concernant les données relationelles, en dépit du fait que de nombreuses applications (comme par exemple l'identification de contenus visuels) en auraient besoin (Frigui et al., 2007).

Cet article introduit l'algorithme de classification donné dans (De Carvalho et Lechevallier, 2007) capable de partitionner des objets en prenant en compte de manière simultanée plusieurs matrices de dissimilarité qui les décrivent. L'idée générale est que chaque matrice de dissimilarité ait un rôle collaboratif (Pedrycz, 2002) dans le but d'arriver à un consensus sur une partition (Leclerc et Cucumel, 1987). Ces matrices peuvent avoir été générées en utilisant différents ensembles de variables et une fonction de dissimilarité unique, un ensemble de variables donné et différentes fonctions de dissimilarité ou bien différents ensembles de variables et de fontions de dissimilarité. L'influence (ou poids) de ces différentes matrices de dissimilarité n'est pas identique pour définir les clusters de la partition finale et cette pertinence doit être calculée grâce à un apprentissage tout au long du déroulement de l'algorithme.

(Frigui et al., 2007) propose CARD, un algorithme capable de partitionner des objets en prenant en compte de manière simultanée plusieurs matrices de dissimilarité et d'apprendre un poids pour chaque matrice de dissimilarité dans chaque cluster en fonction de sa pertinence. CARD est basé sur les algorithmes très connus des nuées dynamiques pour des données relationnelles RFCM (Hathaway et al., 1989) et FANNY (Kaufman et Rousseeuw, 1990).

L'algorithme décrit dans ce papier est conçu pour donner une partition et un prototype pour chacun des clusters tout en apprenant quel poids donner à chacune des matrices de dissimilarité par optimisation d'un critère d'adéquation qui mesure l'adéquation entre un cluster et son représentant. Cette pondération change à chaque itération de l'algorithme et est différent pour chaque cluster. Il est basé sur l'algorithme des nuées dynamiques pour des données relationelles décrit par (Lechevallier, 1974; De Carvalho et al., 2008) ainsi que celui utilisant les distances adaptatives (Diday et Govaert, 1977; De Carvalho et Lechevallier, 2009).

Pour montrer l'utilité de cet algorithme, nous l'appliquons à la catégorisation automatique d'un ensemble de documents homogènes rédigés en anglais pour lequel nous connaissons déjà une partition données *a priori* par des experts.

La section 2 de cet article présente l'algorithme proposé. La section 3 montre l'application de l'algorithme à un cas concret dans le but de mettre en évidence son utilité. Enfin, la section 4 tire quelques conclusions.

2 Un algorithme de partitionnement prenant en compte plusieurs matrices de dissimilarité

Dans cette section nous introduisons une extension de l'algorithme des nuées dynamiques pour des données relationelles (De Carvalho et al., 2008) qui permet de partitionner en ensem-

ble d'objets et prenant en compte une description utilisant plusieurs matrices de dissimilarité.

Soit $E = \{e_1, \ldots, e_n\}$ un ensemble de n exemples et soit p matrices de dissimilarité $n \times n$ $(\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_p)$ où $\mathbf{D}_j[i,l] = d_j(e_i, e_l)$ donne la dissimilarité entre les objets e_i et e_l dans la matrice de dissimilarité $\mathbf{D}_j$. Supposons que le prototype g_k du cluster C_k appartienne à l'ensemble d'exemples E, *i.e.*, $g_k \in E \ \forall k = 1, \ldots, K$.

L'algorithme des nuées dynamique avec pondération pour chaque matrice de dissimilarité cherche une partition $P = (C_1, \ldots, C_K)$ de E en K clusters ainsi que le prototype correspondant $g_k \in E$ representant le cluster C_k dans P de telle façon que le critère d'adéquation mesurant l'adéquation entre un cluster et son prototype soit localement optimisé. Le critère d'adéquation est défini par

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

dans lequel

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

est la dissimilarité entre un exemple $e_i \in C_k$ et le prototype du cluster $g_k \in E$ paramétrisé par le vecteur de pondération $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^j, \ldots, \lambda_k^p)$ où λ_k^j est le poids de la matrice de dissimilarité $\mathbf{D}_j$ pour le cluster C_k, et $d_j(e_i, g_k)$ est la mesure de dissimilarité locale d_j entre un exemple $e_i \in C_k$ et le prototype du cluster $g_k \in E$.

La matrice de pondération de la pertinence $\boldsymbol{\lambda}$ composée de K vecteurs de pondération $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^j, \ldots, \lambda_k^p)$ change à chaque itération, *i.e.*, ils ne sont pas déterminés de façon absolue et sont différents pour chaque cluster.

Notre algorithme alterne les trois étapes suivantes :

– ***Étape 1 : Definition des meilleurs prototypes***
Dans cette étape, la partition $P = (C_1, \ldots, C_K)$ de E en K clusters et la matrice de pondération de la pertinence $\boldsymbol{\lambda}$ sont fixés.
Proposition 1 Le prototype $g_k = e_l \in E$ du cluster C_k, qui minimise le critère J, est calculé en utilisant :

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

– ***Étape 2 : Définition de la meilleure matrice de pondération de la pertinence***
Dans cette étape, la partition $P = (C_1, \ldots, C_K)$ de E et le vecteur de prototypes $\mathbf{g} = (g_1, \ldots, g_K)$ sont fixés.

Proposition 2 L'élément j du vecteur de pondération de la pertinence $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^p)$, qui minimise le critère J avec $\lambda_k^j > 0$ et $\prod_{j=1}^p \lambda_k^j = 1$, est calculé en utilisant l'expression suivante :

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

– ***Étape 3 : Définition de la meilleure partition***
Dans cette étape, le vecteur de prototypes $\mathbf{g} = (g_1, \ldots, g_K)$ et le matrice de pondération de la pertinence $\boldsymbol{\lambda}$ sont fixés.
Proposition 3 Le cluster C_k, qui minimise le critère J, est mis à jour en utilisant la règle d'allocation suivante :

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

Si le minimum n'est pas unique, e_i est affecté à la classe qui possède le plus petit index.

Il est facile de montrer que chacune de ces trois étapes fait décroître le critère J. L'algorithme des nuées dynamiques avec pondération de la pertinence de chacune des matrices de dissimilarité démarre avec une partition initiale et alterne trois étapes jusqu'à convergence, quand le critère $J(P, \boldsymbol{\lambda}, \mathbf{g})$ atteint une valeur stationnaire qui représente un minimum local.

Cet algorithme est résumé ci après.

L'algorithme des nuées dynamiques avec pondération de la pertinence des matrices de dissimilarité

1. *Initialisation*.
   Fixer le nombre K de clusters ;
   Sélectionner de manière aléatoire K objets distincts $g_k \in E$;
   Fixer la matrice de pondération de la pertinence $\boldsymbol{\lambda}$ où $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^p) = (1, \ldots, 1)$;
   Assigner chaque objet e_i au prototype le plus proche de manière à obtenir une partition $P = (C_1, \ldots, C_K)$ où C_k est construit en utilisant la règle (5).
2. *Étape 1 : définition des meilleurs prototypes*.
   La partition $P = (C_1, \ldots, C_K)$ et la matrice de pondération de la pertinence $\boldsymbol{\lambda}$ sont fixés.
   Trouver le prototype $g_k \in E$ du cluster C_k en utilisant (3)
3. *Étape 2 : définition de la meilleure matrice de pondération de la pertinence*.
   Le vecteur de prototypes $\mathbf{g}$ et la partition $P = (C_1, \ldots, C_K)$ sont fixés.
   Pour chaque k calculer le composant de vecteur de pondération $\boldsymbol{\lambda}_k$ en utilisant (4)

4. *Étape 3 : définition de la meilleure partition.*
   Le vecteur de prototypes $\mathbf{g}$ et la matrice de pondération de la pertinence $\boldsymbol{\lambda}$ sont fixés.
   Construire la nouvelle partition $P' = (C'_1, \ldots, C'_K)$ avec la règle donnée par (5) et controler la convergence avec :
   $test \leftarrow 0$;
   pour $i = 1$ jusqu'à n faire
      e_i appartenait à la classe C_m et appartient au cluster gagnant C'_k
      si $k \neq m$ alors $test \leftarrow 1$;
   $P \leftarrow P'$;
5. *Critère d'arrêt.*
   Si $test = 0$ alors STOP, sinon aller en 2 (Étape 1).

3 Application : partitionnement d'une base documentaire

Pour illustrer notre propos et montrer l'utilité de ce nouvel algorithme, nous l'utilisons pour partitionner une base de données de documents. Cette base est la collection des rapports d'activité produits par les différentes équipes de recherche de l'Inria (Institut National de Recherche en Informatique et Automatique) en 2007. Les activités de recherche de l'Inria sont organisées en thèmes de recherche. Ces thèmes de recherche ne correspondent pas à une structure organisationnelle mais permettent seulement de faciliter la présentation des activités de l'Inria et son évaluation. Le choix de ces thèmes de recherche et l'affectation des différentes équipes dans l'un de ces thèmes prennent en compte les objectifs stratégiques de l'institut, la proximité scientifique entre équipes, mais aussi d'autres contraintes plus politiques comme la volonté de faire apparaître des thématiques fortes dans certaines zones géographiques. Notre but est de comparer le partitionnement optenu de façon automatique par l'algorithme que nous avons décrit avec la présentation officielle, que nous décrirons comme experte, donnée *a priori* par l'Inria.

Ces rapports d'activité sont rédigés en anglais. Les sources sont des documents laTeX qui sont traduits de façon automatique en XML afin d'être publié sur le Web. Dans la suite de cet article nous ferons implicitement toujours référence à la version XML de ces rapports d'activité. Ces documents sont homogènes et leur structure est définie par une DTD XML qui contient des sections obligatoires et d'autres optionnelles.

Nous donnons à titre d'illustration un extrait de cette DTD.

```
<!ELEMENT raweb (header, moreinfo?, members, presentation,
                 foundation?, domain?, software?,  results,
                 contracts?, international?, dissemination?,
                 biblio) >
<!ATTLIST raweb year CDATA #IMPLIED >
```

Dans cette application nous considérons les rapports d'activité de 164 équipes de recherche de l'Inria portant sur l'année 2007.

La version XML de ces rapports représentent au total plus de 613 000 lignes de source, soit plus de 40 Moctets de données.

FIG. 1 – *Exemple d'un sommaire de rapport d'activité.*

Dans ces rapports, 4 sections ont été sélectionnées pour décrire l'activité des équipes de recherche : *overall objectives*, *scientific foundations*, *dissemination* and *new results*. La section *overall objectives* décrit les objectifs scientiques de l'équipe, alors que la section *scientific foundations* décrit les fondements de la discipline ainsi que tout le matériel scientifique qui va être utile pour l'atteinte des objectifs. La section *Dissemination* contient les activités d'enseignement, l'implication dans la communauté scientifique (comités de programme, conférence éditoriale, organisation de séminaires, workshop et conférences). La partie *new results* décrit les principaux résultats ou avancées obtenus pendant l'année.

Dans un premier temps le contenu des rapports d'activité est traité pour supprimer certains mots non significatifs (stop-words), puis le texte est passé dans un lemmatiseur afin de supprimer les flexions et remplacer chaque mot par sa forme de référence (lemme ou forme canonique).

Puis 4 tables de données (feature data tables) sont construites chacune avec 164 individus (les équipes de recherche de l'Inria) décrites par les mots fréquents (catégories) présents dans une des 4 variables. Le nombre de mots fréquents dans la section *overall objectives* est de 220, 210 pour *scientific foundations*, 404 pour *dissemination* et 547 pour *new results*. Chaque

FIG. 2 – *Classification officielle (a priori) des équipes de recherche de l'INRIA.*

cellule dans une table de donnée donne la fréquence d'un mot dans la section concernée du rapport d'activité concerné pour une équipe de recherche.

Ensuite, 4 tables de données relationnelles sont obtenues à partir des 4 tables de données (feature data tables) au moyen d'une mesure de dissimilarité dérivée du coefficient d'affinité (Bacelar-Nicolau, 2000). Nous supposons que chaque individu est décrit par une variable multivaluée ("presentation", etc.) qui a m_j modalités (ou catégories) $\{1, \ldots, m\}$. Un individu e_i est décrit par $\mathbf{x}_i = (n_{i1}, \ldots, n_{im})$ où n_{ij} est la fréquence de la modalité j. La dissimilarité entre une paire d'individus e_i et $e_{i'}$ est donnée par :

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

Toutes ces tables de données relationnelles sont normalisées suivant leur dispersion totale (Chavent, 2005) de telle manière qu'elles aient la même dispersion. Ceci veut dire que chaque dissimilarité $d(\mathbf{x}_i, \mathbf{x}'_i)$ dans une table de données relationnelles a été normalisée en $\frac{d(\mathbf{x}_i, \mathbf{x}'_i)}{T}$ où $T = \sum_{i=1}^{n} d(e_i, g)$ est la dispersion totale et $g = e_l \in E = \{e_1, \ldots, e_n\}$ est le prototype global, calculé suivant $l = argmin_{1 \leq h \leq n} \sum_{i=1}^{n} d(e_i, e_h)$.

3.1 Résultats

Notre algorithme de classification a été appliqué simultanément sur les 4 tables de données relationnelles ("presentation", "foundation", "dissemination" et "bibliography") pour obtenir une partition en $K \in \{1, \ldots, 15\}$. Pour un nombre de cluster donné K, l'algorithme est déroulé 100 fois et le meilleur résultat vis à vis du critère d'adéquation est selectionné.

Pour déterminer le nombre de clusters, nous utilisons l'approche décrite par (Da Silva, 2009), qui consiste à choisir les pics sur le graphe des "différences de second ordre" du critère de classification (équation (1)) : $J^{(K-1)} + J^{(K+1)} - 2J^{(K)}$, $K = 2, \ldots, 14$. En suivant cette approche, nous avons fixé le nombre de clusters à 4 et 9.

Ces partitions en 4 et 9 clusters obtenues par notre algorithme ont été comparées avec la catégorisation en 5 classes des équipes de recherches données *a priori* par l'Inria. Cette catégorisation en 5 classes connue *a priori* est la suivante : "Mathématiques appliquées, calcul et simulation (M)", "Algorithmique, programmation, logiciels et architectures (A)", "Réseaux, systèmes et services, calcul distribué (R)", "Perception, cognition, interaction (P)", "STIC pour les sciences de la vie et de l'environnement (V)". Ces 5 catégories sont elles-même divisées en sous-categories.

Sous plusieurs aspects, nous avons retrouvé avec la classification automatique en 9 clusters la présentation faite *a priori* par l'Inria. Par exemple, la sous-catégorie "Réseaux et télécommunications" de R rentre exactement dans un unique cluster. Les deux sous-catégories "Systèmes et services distribués" et "Calcul distribué et applications à très haute performance" sont fusionnées dans un cluster unique, suggérant par là que du point de vue du langage utilisé la distinction entre ces deux catégories était artificielle.

Nous assistons aussi à des "migrations" de certaines équipes. Par exemple, la classification automatique en 9 clusters montre que le langage utilisé en cryptographie (présenté dans une sous-catégorie de A dans la présentation *a priori*) est beaucoup plus proche du langage utilisé en maths (catégorie M).

Certaines migrations portent un sens politique. C'est le cas en particulier pour les équipes pluri-disciplinaires où il peut être interessant de mettre en avant un thème porteur, même s'il n'est pas central. Cela se voit dans la classification automatique en 4 clusters dans laquelle on détecte très clairement ce genre de phénomène. Cette opinion est confortée par le fait que la présentation faite par l'Inria est revue avec une périodicité variable. Certaines divergences entre notre classification automatique et la présentation officielle n'existait pas dans une version précédence de cette présentation.

Enfin, il apparaît que les équipes classées dans la catégorie V utilisent le même langage que les équipes classées en M ou en P. Ceci indique que dans ces rapports d'activité, le poids de la section "scientific foundations" est très important au détriment des applications, ce fait étant plus visible en regardant le partitionnement en 4 clusters, mais apparaît aussi dans le partitionnement en 9 clusters.

4 Conclusion

Cet article introduit un nouvel algorithme de classification capable de partitionner en ensemble d'objets en tenant compte de manière simultanée de leurs descriptions relationnelles données à l'aide de plusieurs matrices de dissimilarité. Ces matrices peuvent avoir été générées

en utilisant différents ensembles de variables et différentes fonctions de dissimilarité. L'algorithme exhibe une partition et un prototype pour chacun des clusters ainsi qu'une pondération de la pertinence pour chacune des matrices de dissimilarité par optimisation d'un critère d'adéquation qui mesure l'adéquation entre un cluster et son représentant. Cette pondération de la pertinence change à chaque itération de l'algorithme et diffère d'un cluster à un aute.

L'utilité de cet algorithme est montré en utilisant une base de documents contenant des rapports d'activité de l'Inria, le but étant de comparer le résultat obtenu par l'algorihme de classification automatique avec une présentation experte donnée *a priori*. Cette catégorisation *a priori* a pu être retrouvée de façon automatique, des divergences mineures pouvant être expliquées par des choix politiques dans la présentation faite par l'Inria.

Références

Bacelar-Nicolau, H. (2000). The affinity coefficient. In H. H. Bock et E. Diday (Eds.), *Analysis of Symbolic Data*, pp. 160–165. Speinger, Heidelberg.

Bock, H. et E. Diday (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, pp. 670–677.

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

De Carvalho, F. et Y. Lechevallier (2007). Une méthode de partionnement sur un ensemble de tableaux de distances. In *Actes des XIVes Rencontres de la Société Francophone de Classification*, Paris, pp. 79–82. Société Francophone de Classification : ENST.

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

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

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

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

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

Hathaway, R. J., J. W. Davenport, et J. C. Bezdek (1989). Relational duals of the c-means algorithms. *Pattern Recog. 22*, 205–212.

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

Kaufman, L. et P. J. Rousseeuw (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*. Ph. D. thesis, Université Paris-VI.

Leclerc, B. et G. Cucumel (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.

Summary

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.