Un algorithme de classification automatique pour des données relationnelles multi-vues

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

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

Résumé. classification automatique (De Carvalho et al., 2012) 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 de fonctions 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. Nous présentons aussi plusieurs outils d'aide à l'interprétation des groupes et de la partition fournie par cette nouvelle méthode. Deux exemples illustrent l'interêt de la méthode. Le premier utilise des données concernant des chiffres manuscrits (0 à 9) numérisés en images binaires provenant de l'UCI. Le second utilise un ensemble de rapports dont nous connaissons une classification experte donnée à priori.

1 Introduction

La classification est une activité courante en extraction de connaissances. Le but de la classification est d'organiser un ensemble d'objets en sous-ensembles appelés clusters ou classes de telle façon que les objets d'une même classe se ressemblent. Pour quantifier cette ressemblance il est important de bien définir cette dissimilarité entre deux objets à partir de leurs représentations.

Dans le cas où la représentation d'un objet n'est pas unique, ces données sont appelées multi-vues. Elles sont présentes dans plusieurs domaines tels que la bioinformatique, le marketing, etc. (Cleuziou et al., 2009). Dans les documents structurés, par exemple les documents XML, il existe plusieurs blocs ou sections, chacun peut être interprété comme une vue.

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 des distances sur différents ensembles de variables. Cet article présente une amélioration de l'algorithme de classification automatique décrit dans (De Carvalho et al., 2012) capable de partitionner des objets en prenant en compte de manière simultanée plusieurs matrices de dissimilarité les décrivant.

L'influence (ou poids) de ces différentes matrices de dissimilarité n'étant pas identique pour chaque classe, la pertinence doit être calculée tout au long du déroulement de l'algorithme par apprentissage .

L'algorithme, décrit dans ce texte, est conçu pour donner une partition, un vecteur de prototypes pour chacune des classes et une pondération à chacune des matrices de dissimilarité par optimisation d'un critère d'adéquation entre ces trois éléments. Cette pondération est différente pour chaque classe. Cet algorithme est basé sur l'algorithme des nuées dynamiques pour des données relationelles décrit par (Lechevallier, 1974; De Carvalho et al., 2009) en utilisant la notion de distance adaptative proposée par (Diday et Govaert, 1977; De Carvalho et Lechevallier, 2009). Dans cette version nous avons modifié l'étape représentation de l'algorithme décrit par (De Carvalho et al., 2012) et nous proposons plusieurs outils d'aide à l'interprétation des classes et de la partition fournies par la méthode.

Dans l'étape de représentation nous avons un vecteur d'objets de E et non un seul objet. A chaque vue et à chaque classe on associe un objet de E ce qui nous permet d'avoir, comme pour l'étape pondération, une approche locale.

Pour montrer l'utilité de cet algorithme, nous l'appliquons à deux exemples d'utilisation. Le premier concerne la catégorisation automatique de données concernant des chiffres manuscrits (0 à 9) numérisés en images binaires (disponible dans UCI machine learning repository) décrits par 6 différents tableaux de variables. Nous proposons aussi de reprendre l'exemple présenté dans (De Carvalho et al., 2010) qui utilisait un ensemble de rapports pour lequel nous disposons d'une classification experte. Dans ce cadre le prototype d'une classe était un élément de E.

2 Une méthode de classification sur plusieurs matrices de dissimilarité

Dans cette section nous introduisons l'amélioration d'un nouvel algorithme des nuées dynamiques pour des données relationnelles (De Carvalho et al., 2012) qui permet de partitionner un ensemble d'objets en fonction d'une description basée sur plusieurs matrices de dissimilarité.

Dans cette nouvelle version, le prototype est donc défini non plus comme un individu de E mais comme un vecteur d'individus de E.

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 vecteur $\mathbf{g}_k = (g_{k1}, \ldots, g_{kp})$ est le vecteur prototype g_k de la classe C_k, où ses composantes appartiennent à l'ensemble E, ***i.e.***, $\mathbf{g}_k \in E^p$ $(k = 1, \ldots, K)$, avec $g_{kj} \in E$ $(j = 1, \ldots, p)$.

Cet algorithme des nuées dynamiques cherche une partition $P = (C_1, \ldots, C_K)$ de E en K classes, le vecteur de prototypes correspondant $\mathbf{g}_k \in E^p$ representant la classe C_k dans P

et une pondération de chaque matrice de dissimilarité de telle façon que le critère d'adéquation J soit localement optimisé.

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

dans lequel

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

est la dissimilarité entre un objet $e_i \in C_k$ et le vecteur prototype de la classe $\mathbf{g}_k \in E^p$ paramétrisé par le vecteur de pondération $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kp})$ où λ_{kj} est le poids de la matrice de dissimilarité $\mathbf{D}_j$ pour la classe C_k, et $d_j(e_i, g_{kj}))$ est la mesure de dissimilarité locale d_j entre un objet $e_i \in C_k$ et le prototype $g_{kj} \in E$.

Notre algorithme alterne les trois étapes suivantes :

– ***Étape 1 : Recherche des meilleurs vecteurs prototypes***
Dans cette étape, la partition $P = (C_1, \ldots, C_K)$ de E en K classes et la matrice de pondération de la pertinence $\boldsymbol{\lambda}$ sont fixés.
Pour chaque classe C_k on recherche le vecteur prototype $\mathbf{g}_k$ qui minimise le critère J.
Ce vecteur prototype possède les composants g_{kj}, objets de E, qui sont obtenus par :

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

– ***Étape 2 : Calcul de la meilleure matrice de pondération***
Dans cette étape, la partition $P = (C_1, \ldots, C_K)$ de E et le vecteur de prototypes $\mathbf{g} = (\mathbf{g}_1, \ldots, \mathbf{g}_K)$ sont fixés.
L'élément j du vecteur de pondération $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kp})$, minimisant le critère J avec les contraintes $\lambda_{kj} > 0$ et $\prod_{j=1}^{p} \lambda_{kj} = 1$, est calculé par :

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

Remarque Plus les objets de la classe C_k sont proches de la composante g_{kj} du prototype $\mathbf{g}_k$ par rapport au tableau de dissimilarité $\mathbf{D}_j$, plus sera grande la valeur de la pondération λ_{kj}.

– ***Étape 3 : Construction de la meilleure partition***
Dans cette étape, le vecteur de prototypes $\mathbf{g} = (g_1, \ldots, g_K)$ et la matrice de pondération $\boldsymbol{\lambda}$ sont fixés.
La classe C_k est construite en utilisant la règle d'allocation suivante :

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

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

Il est facile de montrer que chacune de ces trois étapes fait décroître le critère J.

L'algorithme démarre avec une partition initiale et alterne ces trois étapes jusqu'à convergence. Cette convergence est atteinte quand la valeur du critère $J(P, \boldsymbol{\lambda}, \mathbf{g})$ est stationnaire.

Par rapport à l'algorithme initial (De Carvalho et al., 2012), l'utilisation d'un vecteur prototype permet une optimisation du prototype et de la pondération localement, par groupe et par tableau de dissilimarité. Le critère de convergence J se décompose donc selon les tableaux de dissimilarités et selon les groupes et les tableaux de dissimilarité simultanément, ce qui permet d'interpréter les groupes par rapports aux tableaux.

3 Interprétation des classes et de la partition

Soit le critère T qui correspond au critère J appliqué à la partition en une seule classe de E. Les outils de l'aide à l'interprétation des classes et de la partition sont basés sur la décomposition du critère T en deux parties. L'une correspond à la dispersion intra-classes W et l'autre à la dispersion inter-classes B. Ici nous utiliserons l'approche de (Chavent et al., 2006) qui permet de faire cette décomposition bien que le calcul de la dispersion inter-classes soit impossible.

Soit $P = (C_1, \ldots, C_K)$ la partition finale $E = \{e_1, \ldots, e_n\}$ en K classes. Soit $\mathbf{g}_k$ le prototype et $\boldsymbol{\lambda}_k$ le vecteur de pondération de la pertinence du groupe $C_k\,(k = 1, \ldots, K)$. Supposons aussi que le prototype global est le vecteur $\mathbf{g} = (g_1, \ldots, g_p)$ où $g_j \in E\,(j = 1, \ldots, p)$.

La dispersion globale de la partition $P = (C_1, \ldots, C_K)$ est définie comme étant

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

où le prototype global **g**, qui minimise la dispersion globale T, a les composants $g_j = e_l \in E$ calculés en utilisant :

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

La dispersion globale se décompose
a) $T = \sum_{k=1}^{K} T_k$ avec $T_k = \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj}\, d_j(e_i, g_j)$;
b) $T = \sum_{k=1}^{K} \sum_{j=1}^{p} T_{kj}$ avec $T_{kj} = \sum_{e_i \in C_k} \lambda_{kj}\, d_j(e_i, g_j)$;
c) $T = \sum_{j=1}^{p} T_j$ avec $T_j = \sum_{k=1}^{K} \sum_{e_i \in C_k} \lambda_{kj}\, d_j(e_i, g_j)$
La dispersion intra-groupe J est donnée par l'équation (1).
a) $J = \sum_{k=1}^{K} J_k$ avec $J_k = \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj}\, d_j(e_i, g_{kj})$;
b) $J = \sum_{j=1}^{p} J_j$ avec $J_j = \sum_{k=1}^{K} \sum_{e_i \in C_k} \lambda_{kj}\, d_j(e_i, g_{kj})$;
c) $J = \sum_{k=1}^{K} \sum_{j=1}^{p} J_{kj}$ avec $J_{kj} = \sum_{e_i \in C_k} \lambda_{kj}\, d_j(e_i, g_{kj})$
On peut montrer facilement que
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) $T_{kj} \geq J_{kj}$ $(k = 1, \ldots, K; j = 1, \ldots, p)$.

A partir de la dispersion globale, de la dispersion intra-groupe et de leur décomposition, on peut adapter facilement les indices d'aide à l'interprétation des groupes et de la partition introduits dans (Chavent et al., 2006) pour ce nouvel algorithme.

La qualité globale de la partition finale est $Q(P) = 1 - \frac{J}{T}$. Un indice $Q(P)$ proche de 1 indique une partition de meilleur qualité (des groupes plus homogènes).

La qualité globale de la partition finale par rapport à chaque tableau de dissimilarité est donnée par $Q_j(P) = 1 - \frac{J_j}{T_j}$. Une valeur de $Q_j(P)$ proche de 1 est une indication de la bonne qualité de la partition P par rapport au tableau de dissimilarité $\mathbf{D}_j$. La comparaison de $Q_j(P)$ avec $Q(P)$ montre que le pouvoir discriminant du tableau de dissimilarité $\mathbf{D}_j$ est supérieur au pouvoir discriminant moyen de tous les tableaux de dissimilarité.

4 Application

Pour illustrer notre propos et montrer l'utilité de ce nouvel algorithme, nous l'utilisons sur deux jeux de données. Le premier concernent des chiffres manuscripts numérisés, le second un ensemble de rapports.

4.1 Classification de chiffres manuscripts

Notre premier exemple concerne le partitionnement de données "multiple features" disponible dans "UCI machine learning repository". Ce jeux de données concerne des chiffres manuscrits (0 à 9) numérisés en images binaires. Les 2000 chiffres manuscrits (individus) sont décrits par 649 variables numériques. Ces variables sont partitionnées en 6 différents ensembles ("vues") : i) 76 coefficients de Fourier décrivant la forme des caractères ; ii) 216 corrélations de profil ; iii) 64 coefficients de Karhunen-Love ; iv) 240 moyennes de pixels dans des fenêtres 2 x 3 ; v) 47 moments de Zernike ; et vi) 6 caractéristiques morphologiques. Ces données sont formées par 10 classes a priori de 200 individus, chaque classe correspondant à un chiffre manuscrit.

Initialement 7 tableaux de données sont considerés : un tableau où les individus sont décrits par toutes les 649 variables (tableau "mfeat") et 6 autres tableaux où les individus sont décrits selon chacune des 6 différentes "vues", chaque "vue" ayant respectivement, 76 (tableau "mfeatFou"), 216 (tableau "mfeatFac"), 64 (tableau "mfeatKar"), 240 (tableau "mfeatPix"), 47 (tableau "mfeatZer") et 6 (tableau "mfeatMor") variables. Ensuite, 7 tableaux de données relationnelles sont obtenues à partir de ces 7 tableaux de données au moyen de la distance Euclidienne. Tous ces tableaux de données relationnelles sont normalisés 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 un tableau 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 $l = argmin_{1 \leq h \leq n} \sum_{i=1}^{n} d(e_i, e_h)$.

Notre algorithme de classification a été appliqué d'abord sur le tableau de données relationelle "mfeat" et ensuite simultanément sur les 6 tableaux de données relationnelles "mfeatFou", "mfeatFac", "mfeatKar", "mfeatPix", "mfeatZer" et "mfeatMor", correspondant aux 6 différentes "vues" pour obtenir une partition en 10 groupes. L'algorithme est déroulé 100 fois

et le meilleur résultat vis à vis du critère d'adéquation J est selectionné. Notre but est de comparer le partitionnement obtenu de façon automatique par cet algorithme avec le partitionnement a priori des données en 10 classes. Les critères de comparaison choisis sont le taux global d'erreur de classification ($OERC$), l'indice de Rand corrigé (CR) et la F-measure.

Résultats

Les valeurs des indices CR, F-measure et $OERC$, obtenus à partir de la partition finale donnée par l'algorithme appliqué sur le tableau de données relationelle "mfeat", sont respectivement 0.518, 0.674 et 37.75%.

Les valeurs de ces mêmes indices, obtenus à partir de la partition finale donnée par l'algorithme appliqué simultanément sur les 6 tableaux de données relationnelles correspondant aux 6 différentes "vues", sont respectivement 0.762, 0.879 et 12.10%. Le Tableau 1 montre la matrice des poids de pertinence des tableaux de données relationnelles dans les groupes.

TAB. 1 – *Matrice des poids de pertinence des tableaux de dissimilarité dans les groupes*

	Poids de pertinence des tableaux de dissimilarités					
Groupes	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
	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

Le Tableaux 2 montre la matrice de confusion en 10 groupes calculée pour la partition finale.

TAB. 2 – *Matrice de confusion*

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

On peut voir, par exemple, que le tableau de dissimilarité “mfeatMor” est le plus pertinent dans la définition de tous les groupes. Entre autre, le tableau de dissimilarité “mfeatFac” a un poids de pertinence presque aussi important que celui du tableau de dissimilarité “mfeatMor” pour le groupe 3.

La qualité globale de la partition finale est $Q(P) = 1 - \frac{J}{T} = 0.919$. Un indice $Q(P)$ proche de 1 indique une partition de meilleur qualité (des groupes plus homogênes).

La qualité globale de la partition finale par rapport à chaque tableau de dissimilarité $Q_j(P) = 1 - \frac{J_j}{T_j}$ $(j = 1, \ldots, 6)$ est montré dans le Tableau 3. Une valeur de $Q_j(P)$ proche de 1 est une indication de la bonne qualité de la partition P par rapport au tableau de dissimilarités $\mathbf{D}_j$. La comparaison de $Q_j(P)$ avec $Q(P)$ montre que le pouvoir discriminant du tableau de dissimilarité “mfeatMor” est supérieur au pouvoir discriminant moyen de tous les tableaux de dissimilarité.

TAB. 3 – *Qualité globale de la partition P par rapport à chaque tableau de dissimilarité (%)*

	Tableau de dissimilarités					
	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

Le Tableau 4 montre l’indice d’hétérogénéité $J(k) = \frac{J_k}{J}$ et l’indice de qualité $Q(k) = 1 - \frac{J_k}{T_k}$ pour chaque groupe $k = 1, \ldots, 10$. On peut voir, par exemple, que le groupe 10 (chiffre ’9’) est le plus homogène tandis que le groupe 6 (chiffre ’0’) est celui de meilleure qualité.

TAB. 4 – *Indice de hétérogénéité et indice de qualité d’un groupe(%)*

	Groupe 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

Le Tableau 5 montre l’indice $Q_j(k) = 1 - \frac{J_{kj}}{T_{kj}}$, qui donne la qualité du groupe C_k $(k = 1, \ldots, 10)$ dans le tableaux de dissimilarité $\mathbf{D}_j$ $(j = 1, \ldots, 6)$. Plus la valeur de cet indice est proche de 1, meilleure est la qualité de ce groupe dans ce tableau de dissimilarité. En plus, la comparaison entre les indices $Q_j(k)$ et $Q(k)$ donne les tableaux de dissimilarité qui caractérisent le groupe k. Par exemple, le tableau de dissimilarité “mfeatMor” est caractéristique des groupes 1 à 9, tandis que les tableaux “mfeatZer” et “mfeatFac” sont caractéristiques du groupe 10 (chiffre ’9’).

4.2 Classification de rapports

La base utilisée 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

TAB. 5 – *Qualité des groupes dans les tableaux de dissimilarité (%)*

	Tableau de dissimilarité					
Groupes	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

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. Ces documents sont homogènes et leur structure est définie par une DTD XML qui contient des sections obligatoires et d'autres optionnelles.

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.

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

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 $l = argmin_{1\leq h\leq n} \sum_{i=1}^{n} d(e_i,e_h)$.

4.2.1 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.

La détermination du nombre approprié de classes dans une partition est un problème classique mais il n'existe aucune bonne solution. Notre stratégie pour la détermination du bon nombre de classes est celle qui est proposée dans le logiciel SPAD. Elle consite à sélectionner le meilleur couple (inertie intraclasse, nombre de classes). Comme la diminution du nombre de classes fait augmenter l'inertie intra-classe il faut repérer un saut important de l'indice pour avoir une partition de bonne qualité. Ce coude est repéré à l'aide des dérivées premières et secondes (Da Silva, 2009).

La dérivée première discrète de J au nombre de classe k est $Df(x) = (f(x+h)-f(x))/h$ et la dérivée seconde discrète est $D2f(x) = (f(x+h) - 2f(x) + f(x-h))/h^2$. Lorsque h tend vers 0, on retrouve la dérivée usuelle.

La partition en 4 classes est retenue car la dérivée seconde est maximale (voir Fig. 1), la partition en 11 classes pourrait aussi être retenue car elle est un maximum local.

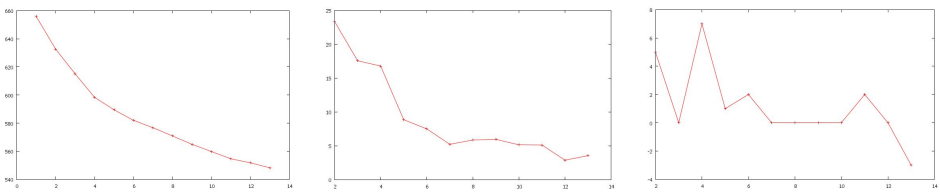

FIG. 1 – *critère J, dérivée première , dérivée seconde*

Cette partition en 4 clusters obtenues par notre algorithme a été comparée avec la catégorisation en 5 classes des équipes de recherches données *a priori* par l'Inria en 2008. 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, mais nous ne nous sommes intéressés qu'aux catégories de premier niveau.

Entre 2007, années sur laquelle porte l'analyse des rapports et la classification faite par l'Inria en 2008, certaines équipes de recherche ont évoluées, voire disparues. Pour cette raison seulement 154 rapports seulement ont pus être pris en compte pour comparer la classification automatique avec celle de l'Inria.

La classification en 4 classes que nous avons générée est cohérente avec celle en 5 classes donnée a priori et cela demande quelques explications.

La répartition est donnée dans le tableau 6.

TAB. 6 – *Répartition de la classification de 154 rapports (2007) en 5 thèmes (2008) dans les 4 clusters*

	C1	C2	C3	C4
M - Mathématiques appliquées, calcul, simulation	1	1	**20**	6
A - Architectures, logiciels, systèmes, programmation et algorithmique	**17**	3	1	9
R - Réseaux, systèmes et services, calcul distribué	1	**28**	2	2
P - Perception, cognition, interaction	5	1	2	**35**
S - STIC pour les sciences de la vie et de l'environnement	0	0	**11**	**9**

TAB. 7 – *Matrice des poids de pertinence des tableaux de dissimilarité dans les groupes*

	Poids de pertinence des tableaux de dissimilarité			
Classes	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

Le tableau 6 montre clairement que le thème 5, STIC pour les sciences de la vie et de l'environnement, est artificiel et qu'il se réparti (au point de vue du vocabulaire utilisé) dans deux clusters, suivant qu'il traite d'aspects plutôt mathématiques ou plutôt cognitifs. Ainsi le cluster C3 pourrait être labellisé "Simulation/contrôle/modélisation", et le cluster C4 "Traitement de l'information".

Un classement différent peut s'expliquer aussi suivant qu'on veuille appuyer davantage sur l'aspect fondamental ou sur l'aspect application. Ainsi, l'équipe SIGNES est classée dans le groupe P dans la classifiaction experte. Mais quand on regarde le libellé en français, "Signes linguistiques, grammaire et sens : algorithmique logique de la langue" on comprend bien que le rapport va beaucoup parler d'algorithmes et que son vocabulaire le classe plus naturellement dans le cluster C1 qui est très proche du thème A.

Une mauvaise affectation peut provenir d'un langage ambigüe ou commun à des disciplines différentes. Par exemple l'équipe EDELWEISS a été classée dans le cluster C2, proche du thème réseau. Cette équipe s'occupe en effet de réseaux, mais des réseaux sociaux.

La matrice des poids de pertinence des 4 sections utilisées dans les rapports est présentée dans le tableau 7.

5 Conclusion

Cet article introduit un nouvel algorithme de classification capable de partitionner un 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 autre. Plusieurs outils d'aide à l'interprétation des groupes et de la partition fournies par cet algorithme a pu aussi être introduit.

L'utilité de cet algorithme est montré en utilisant deux jeux de données : - Des données concernant des chiffres manuscrits (0 à 9) numérisés en images binaires d'une part ; les bons résultats obtenus sur ces données selon plusieurs indices d'évaluation de la partition finale et l'utilité des indices d'aide à l'interprétation liés à cet algorithme donnent des indications du potentiel de la méthod. - Des rapports d'autre part, avec là aussi une bonne pertinence de la classification générée.

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.

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.

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

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

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

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

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 F. M. De Melo (2012). Partitioning hard clustering algorithms based on multiple dissimilarity matrices. *Pattern Recognition 45*(1), 447–464.

De Carvalho, F. A. T., D. T., F. M. De Melo, et Y. Lechevallier (2010). Utilisation de matrices de dissimilarité multiples pour la classification de documents. In *EGC-M'2010, Extraction et gestion des connaissances*, Alger, Algérie, pp. 1–10.

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

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 an improvement of a clustering algorithm (De Carvalho et al., 2012) 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 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. Moreover, various tools for the partition and cluster interpretation furnished by this new algorithm are also presented. Two experiments demonstrate the usefulness of this clustering method and the merit of the partition and cluster interpretation tools. The first one uses a data set from UCI machine learning repository concerning handwritten numbers (digitalized pictures). The second uses a set of reports for which we have an expert classification given a priori.