2011

XVIIIèmes Rencontres de la Société Francophone de Classification

28- 30 Septembre 2011
Orléans

Avant-Propos

La Société Francophone de Classification (SFC) organise du 28 au 30 septembre 2011 ses traditionnelles journées 'Rencontres', SFC'11, à l'Université de cette belle ville d'Orléans, dans la région des célèbres 'Châteaux de la Loire'.

L'organisation de ces rencontres a été prise en charge par des enseignants-chercheurs du laboratoire d'informatique (LIFO) et du laboratoire de mathématiques (MAPMO). Heureuse coïncidence, tant la Classification est devenue tributaire de ces deux grandes disciplines qu'on aurait tort de vouloir dissocier.

Les membres du comité de programme de SFC'11 ont voulu, par la qualité et la diversité des travaux des trois conférenciers invités ainsi que par le choix des sessions, continuer à souligner les liens entre la Classification et divers autres domaines : apprentissage, bioinformatique, co-clustering, dissimilarités, données temporelles, données symboliques, fouille de données, flux de données, images et textes, optimisation, réseaux sociaux, treillis, visualisation des données.

Je remercie vivement:

- le comité d'administration de la SFC pour leur très important support scientifique et matériel;
- les organismes qui ont aidé, par un apport financier, à la réalisation de SFC'11 : le CNRS INS2I, l'Université d'Orléans, la Région Centre, le Conseil Général du Loiret, la Ville d'Orléans;
- tous les membres du comité de programme et tous les relecteurs supplémentaires pour la qualité de leur travail et leur contribution à l'élaboration de ces actes;
- tous les membres du comité d'organisation pour l'important travail qu'ils ont accompli.

Je remercie vivement et tout particulièrement Guillaume Cleuziou du LIFO, Président du comité d'organisation, pour son extraordinaire travail dans les deux comités de SFC'11.

Richard Emilion

MAPMO

Université d'Orléans

Comité de Programme

Président

- Richard Emilion (MAPMO, Univ. Orléans)

Membres

- R. Abdesselam (ERIC - Univ. Lyon 2)
- H. Azzag (LIPN - Univ. Paris 13)
- S. Ben Yahia (Fac. sciences de Tunis, TUNISIE)
- Y. Bennani (LIPN - Univ. Paris 13)
- P. Bertrand (Ceremade - Univ. Paris Dauphine)
- C. Biernacki (LIFL - Univ. Lille)
- G. Bisson (TIMC-IMAG - Univ. Joseph Fourrier Grenoble)
- H. Bock (Univ. Aachen, ALLEMAGNE)
- P. Brito (Univ. Porto, PORTUGAL)
- F. Brucker (LIF - Univ. Marseille)
- D. Chauveau (MAPMO - Univ. Orléans)
- M. Chavent (IMB - Univ. Bordeaux)
- B. Crémillieux (GREYC - Univ. Caen)
- G. Cucumel (Univ. Montréal, CANADA)
- F. De Carvalho (UFPE, BRESIL)
- L. Delsol (MAPMO - Univ. Orléans)
- J. Diatta (LIM – Univ. de La Réunion)
- J.P. Domenger (LaBRI - Univ. Bordeaux)
- F. D'alché-Buc (IBISC - Univ. Evry)
- P. Gançarski (LSIIT - Univ. Strasbourg)
- G. Govaert (UTC - Compiègne)
- A. Guénoche (IML - Univ. Marseille)
- A Gély (LITA - Univ. Metz)
- A. Hardy (Faculté de Namur, BELGIQUE)
- P. Kuntz (LINA - Univ. Nantes)
- M. Lebbah (LIPN - Univ. Paris 13)
- Y. Lechevallier (INRIA - Rocquencourt)
- M.J. Lesot (LIP6 - Univ. Paris 6)
- L. Martin (LIFO - Univ. Orléans)
- C. Maugis (INSA Toulouse)
- E. Mephu Nguifo (LIMOS - Univ. Clermont-Fd)
- M. Nadif (LIPADE - Univ. Paris Descartes)
- A. Napoli (LORIA - Nancy)
- C. Nédellec (INRA - Jouy en Josas)
- N. Niang (CNAM - Paris)
- G. Ritschard (Univ. Genève, SUISSE)
- R. Verde (Univ. Naples, ITALIE)
- M. Vichi (Univ. Rome, ITALIE)
- Ch. Vrain (LIFO - Univ. Orléans)

Relecteurs supplémentaires

- Khalid Benabdeslem
- Lydia Boudjeloud-Assala
- Guénael Cabanes
- Nicolas S. Müller
- Jacques-Henri Sublemontier

Comité d'Organisation

Président

- Guillaume Cleuziou (LIFO - Univ. Orléans)

Membres

- Sylvie Billot (LIFO - Univ. Orléans)
- Davide Buscaldi (LIFO - Univ. Orléans)
- Matthieu Exbrayat (LIFO - Univ. Orléans)
- Vincent Levorato (LIFO - Univ. Orléans)
- Matthieu Lopez (LIFO - Univ. Orléans)
- Lionel Martin (LIFO - Univ. Orléans)
- Yannick Parmentier (LIFO - Univ. Orléans)
- Damien Poirier (LIFO - Univ. Orléans)
- Jacques-Henri Sublemontier (LIFO - Univ. Orléans)
- Isabelle Tellier (LIFO - Univ. Orléans)

Table des matières

Theory of k-means clustering

Christian Sohler

Department of Computer Science
Technische Universität Dortmund
christian.sohler@tu-dortmund.de

The k-means algorithm is a simple, yet effective clustering heuristic to optimize the sum of squared error (SSE) clustering criterion, i.e. to find a set of k centers such that the sum of squared distances from the input points to the nearest center is minimized. It is well known that the k-means algorithm converges to a local optimum. However, the ratio between the local optimum and the global optimum can be arbitrarily large. Since the k-means algorithm is so prominent, we also call the problem to minimize the SSE the k-means clustering problem. In my talk I will survey recent developments in theoretical computer science to analyze the performance of the k-means algorithm as well as other approaches to the k-means problem.

In particular, I will address new ways to give performance guarantees that can be achieved by clever seeding algorithms (algorithm that compute the starting solution for the k-means algorithm). These seeding procedures guarantee a bounded ratio between the cost of the local optimum computed by the k-means algorithm and the cost of the global optimum. Furthermore, I will survey some approaches to make k-means scalable to massive data sets using so-called core-sets, i.e. small weighted subsets of the input data the approximate the input data with respect to the k-means clustering problem.

Computing factors in binary and ordinal data

Radim Belohlavek

Palacky University
Olomouc, Czech Republic
radim.belohlavek@acm.org

The talk will provide an overview of recent developments in computing decompositions of binary matrices and, more generally, matrices with entries from residuated lattices such as the real unit interval $[0, 1]$. The problem that will be discussed consists in finding a decomposition of an $n \times m$ matrix I into a product (Boolean, or more generally, sup-t-norm product) of an $n \times k$ matrix A and a $k \times m$ matrix B. I represents a relationship between n objects and m attributes, the entry I_{ij} represents a degree (such as 0, 1, 0.8, etc.) to which attribute j applies to object i. Matrices A and B represent relationships between the objects and k new attributes, called factors, discovered by finding the decomposition and between the k factors and the attributes.

We show how the structures coming from formal concept analysis, namely Galois connections, closure operators, and concept lattices, may be used to compute optimal decompositions, i.e. decompositions with the least number of factors possible. Furthermore, we present illustrative examples and a greedy approximation algorithm for computing suboptimal decompositions.

The talk will also include open problems and possible research directions.

Classification croisée à base de modèles

Gérard Govaert

HEUDIASYC, UMR CNRS 6599
Université de Technologie de Compiègne
gerard.govaert@utc.fr

Ces dernières années, la classification croisée ou classification par blocs, c'est-à-dire la recherche simultanée d'une partition des lignes et d'une partition des colonnes d'un tableau de données, est devenue un outil très utilisé en fouille de données. Dans cette présentation, nous étudions le problème de la classification croisée en nous appuyant sur un modèle de mélange probabiliste. Différents types de données et de modèles sont envisagés et plusieurs algorithmes de classification sont développés. Des résultats sur des données simulées et des données réelles illustrent et confirment l'efficacité et l'intérêt de cette approche.

Comparaison des partitions par la distance de transfert pour la coloration de graphes

Daniel Cosmin Porumbel*

*Univ. Lille-Nord de France, UArtois, LGI2A, Rue de l'Université, 62400 Béthune, France

La distance de transfert, étudiée initialement par S. Régnier [5] est bien connue en classification pour comparer des partitions [3, 4]. Rappelons brièvement, qu'étant données deux partitions P_1 et P_2 d'un ensemble S, la distance de transfert $d(P_1, P_2)$ est définie comme le nombre minimal d'éléments qui doivent être transférés entre les classes de P_1 pour obtenir une partition égale à P_2. En classification, elle est souvent utilisée pour valider un algorithme en mesurant l'écart entre le résultat de l'algorithme et une solution connue, ou pour comparer les résultats de différentes approches. Dans cette communication, nous l'utilisons dans un contexte différent : celui de l'analyse d'espaces de recherche ("fitness landscapes") pour un problème d'optimisation combinatoire classique : la k-coloration de graphes.

L'objectif général est d'utiliser des informations sur les espaces de recherche pour améliorer les performances des algorithmes méta-heuristiques en optimisation combinatoire. En effet, il est bien connu que les performances des méta-heuristiques dépendent étroitement du choix de paramètres qui est souvent effectué de façon ad-hoc. Une meilleure compréhension des comportements des processus de recherche et des structures des espaces de recherche associés reste nécessaire pour rendre leurs stratégies "mieux informées". L'intérêt de l'apprentissage et de la fouille de données en optimisation combinatoire est en plein essor comme l'attestent outre des publications (e.g. [2, 1]) la récente conférence LION (Learning and Intelligent Optimization).

Dans cette présentation, nous allons insister sur trois points :

- La classification de solutions candidates pour le problème de coloration de graphe ;
- Sa mise en oeuvre via le calcul de la distance de transfert entre colorations (partitions) ;
- Des différentes stratégies d'optimisation et des résultats expérimentaux sur des instances du benchmark DIMACS de coloration de graphe.

Remerciements
Ce travail a été mené en collaboration avec Jin-Kao Hao (LERIA, Angers) et Pascale Kuntz (LINA, Nantes).

Références

[1] R. Battiti, R Brunato, and F. Mascia. *Reactive Search and Intelligent Optimization*. Springer, 2008.

[2] J. Boyan, W. Buntine, and A. Jagota. Statistical machine learning for large-scale optimization. *Neural Computing Surveys*, 3(1) :1–58, 2000.

[3] W.H.E. Day. The complexity of computing metric distances between partitions. *Mathematical Social Sciences*, 1 :269–287, 1981.

[4] L. Denœud and A. Guénoche. Comparison of distance indices between partitions. In V. Batagelj et al., editors, *Data Science and Classification*, pages 21–28. Springer, Berlin, Germany, 2006.

[5] S. Régnier. Sur quelques aspects mathématiques des problèmes de classification automatique. *Mathématiques et Sciences Humaines*, 82 :20, 1983 et 1965. (reprint of *ICC Bulletin*, 4, 175- 191, Rome, 1965).

Recherche de classes dans des réseaux sociaux

Amine Louati**,***
, Rania Soussi *, Marie-Aude Aufaure* Hajer Baazaoui**,Yves Lechevallier***

*Laboratoire MAS Ecole Centrale de Paris, Grande Voie des Vignes
Chatenay-Malabry, France
Rania.Soussi,Marie-Aude.Aufaure@ecp.fr,
**Laboratoire RIADI-GDL, Ecole Nationale des Sciences de l'Informatique,
Campus Universitaire de la manouba,La Manouba 2010
Amine.louati,hajer.baazaouizghal@riadi.rnu.tn,
***INRIA-Rocquencourt,Domaine de Voluceau 78150 Rocquencourt
Amine.Louati,Yves.Lechevallier@inria.fr

Résumé. Les réseaux sociaux permettent d'avoir une vision globale des acteurs et de leurs interactions, facilitant ainsi l'analyse et la recherche d'information. Le réseau a souvent une taille importante ce qui rend son analyse et sa visualisation difficiles ainsi l'étape d'agrégation est une tâche nécessaire. Dans ce travail, nous proposons une méthode d'agrégation basée sur l'algorithme k-SNAP qui produit un graphe résumé en fonction des attributs et des relations sélectionnés par l'utilisateur.

1 Introduction

Les réseaux sociaux jouent un rôle important dans le partage et la recherche d'information. Un réseau social est un ensemble d'entités reliées entre elles par des liens ou des interactions, il est généralement modélisé par une structure de graphe. Les *sommets* désignent les individus ou les organisations, ces sommets sont reliés entre eux par des relations qui forment les *arêtes* de ce graphe.

Le réseau construit peut avoir une taille très importante, aussi il devient difficile d'exploiter et surtout d'interpréter l'information de son graphe par une simple visualisation. D'où la nécessité de disposer de méthodes efficaces d'agrégation produisant un graphe résumé qui conserve non seulement les principales caractéristiques structurelles mais surtout améliore les performances d'analyse et d'interprétation.

2 Agrégation des réseaux sociaux

L'agrégation des graphes est une méthode qui permet de mettre en évidence les communautés présentes dans le réseau, facilitant ainsi l'interprétation et allégeant sa visualisation. La plupart des travaux existants utilisent des procédés statistiques, tels que *degree distributions* (Newman, 2003), *hop-plots* (Chakrabarti et al., 2007) et *clustering coefficients* (Watts

et Strogatz, 1998) ; les résultats obtenus sont souvent utiles mais difficiles à exploiter, d'autres emploient des algorithmes de partitionnement hiérarchique de graphe comme *superGraph* (Rodrigues Jr. et al., 2006) pour visualiser les graphes larges, cependant ces techniques ignorent totalement la description du contenu des nœuds (attributs) dans leurs processus d'agrégation ce qui rend l'interprétation délicate.

Enfin, certains algorithmes tel que *SNAP* ou *k-SNAP* (Tian et al., 2008) utilisent un ensemble de variables qualitatives, appelées "attributs", associé aux nœuds et les relations entre ces nœuds pour agréger le graphe. Cet algorithme est basé sur la notion Groupement A-*compatible* d'attributs et la notion de Groupement (A, R) *compatible* d'attributs et de relations.

A partir d'un ensemble V de sommets et un ensemble de relations $R = \{R_1, R_2, ..., R_r\}$ sur V, on défini $G = (V, E)$ le graphe où $E = \{E_1, E_2, ..., E_r\}$ est l'ensemble des arêtes tel que $(u, v) \in E_i$ si uR_iv.

Les éléments de V sont caractérisés par un ensemble d'attributs (variables qualitatives) $\Lambda(G)$. Pour un ensemble d'attributs $A \subseteq \Lambda$(G), une fonction Φ sur V est dite *groupement A-compatible* si elle vérifie la condition suivante : $\forall u, v \in V, si\, \Phi(v) = \Phi(u), alors\, \forall a_i \in A, a_i(u) = a_i(v)$, elle sera notée Φ_A. Cette fonction Φ_A induit une partition $P = (C_1, C_2, ..., C_k)$ sur V où chacune de ses classes C_i est formée par l'ensemble de nœuds qui ont exactement les mêmes valeurs sur toutes les variables de A.

Sur chaque relation R_i, on note $N_{R_i}(v) = \{u \in V | (u, v) \in E_i\}$ l'ensemble des nœuds voisins de v et $NG_{\Phi_A,R_i}(v) = \{C \in P | \exists u \in C \cap N_{R_i}(v)\}$ l'ensemble des classes associées à Φ_A voisines de v, *i.e* au moins un des individus de chaque classe est voisin de v.

Cette fonction Φ_A sur V est dite *groupement* (A, R)*-compatible* si elle vérifie la condition suivante $\forall u, v \in V, si\, \Phi_A(u) = \Phi_A(v), alors\, \forall R_i \in R$, on a $NG_{\Phi_A,R_i}(u) \equiv NG_{\Phi_A,R_i}(v)$. Dans chaque classe d'un *groupement (A,R) compatible*, les nœuds sont homogènes aussi bien en termes d'attributs de A que de relations de R. En d'autre termes, tous les nœuds d'une même classe ont les même valeurs d'attributs A et sont en relation avec les même classes.

L'objectif de *SNAP* est de construire tous les groupements (A, R)*-compatibles*. *k-SNAP* a été introduit pour améliorer *SNAP* en relaxant le critère d'homogénéité des relations ; pour chaque relation entre deux classes, on n'exige plus que tous les nœuds de ces deux classes y participent. Cependant, on maximise le taux de participation, tout en maintenant le critère *A-compatible*. Pour cela, *k-SNAP* utilise une mesure d'évaluation notée Δ qui permet à déterminer à chaque itération la meilleure classe à diviser jusqu'à ce que le nombre de classes soit égal à K.

3 Notre approche

Nous allons utiliser le principe de *k-SNAP* en maintenant l'étape *A-compatible* mais en modifiant la nouvelle mesure d'évaluation Δ relative à l'*(A,R)-compatibilité* et le principe de découpage. En fait, la division de la classe sélectionnée est réalisée en utilisant la notion de *sommet central d'une classe*.

Cette nouvelle mesure d'évaluation Δ d'une partition P sera basée sur la distance de Jaccard définie sur les voisins communs. Elle est définie comme suit :

$$\Delta(P) = \sum_{R_t \in R} \Delta_t(P) = \sum_{R_t \in R} \sum_{1 \leq i \leq |P|} \delta_i^t \text{ avec } \delta_i^t = \sum_{m \in C_i} \sum_{n \in C_i} d^t(m, n)$$

où $d^t(m,n) = (b+c)/(a+b+c)$ est la distance de Jaccard sur la relation R_t avec $a = |N_{R_t}(m) \cap N_{R_t}(n)|$, $b = |N_{R_t}(m)| - a$ et $c = |N_{R_t}(n)| - a$.

La fonction Φ_p étant la fonction d'affection d'un sommet v de V dans la partition P, le degré du sommet v associé à la relation R_t et à la partition P est égal à $D_{R_t,P}(v) = |N_{R_t}(v) \cap C_{\phi_p(v)}|$. Le sommet central v_d d'une classe C_i de la partition P est défini par : $d = \arg\max_{v \in C_i} D_{R_t,P}(v)$

Partant d'un groupement *A-compatible*, notre procédure consiste à chercher à chaque itération la relation R_t et la classe à diviser C_i qui maximisent la mesure d'évaluation δ_i^t jusqu'à ce que le cardinal de la partition soit égal à K. Le mécanisme de division consiste à déterminer le sommet central v_d de la classe C_d à diviser dont le degré $D_{R_t,P}$ est le plus élevé, on la découpe en deux sous-classes selon la stratégie suivante : l'une contient les voisins du sommet central, l'autre le reste de la classe.

Algorithm 1

Entrée : G un graphe ; K le nombre de classes ; $A \subseteq \Lambda(G)$ un ensemble de variables ; $R = \{R_1, R_2, ..., R_r\}$ un ensemble de relations. *Sortie* : un graphe agrégé en K classes.
1 : P est la partition $A - compatible$ basée sur les valeurs des attributs de A ; $\Delta = 0$.
2 : tant que $|P| < K$ faire
3 : pour chaque $R_j \in R$ faire ; pour chaque $C_i \in P$ faire
5 : calculer δ_i^j valuation de la classe C_i pour la relation R_j.
7 : fin pour ; fin pour
8 : calculer $\delta_d^t = \max_{1 \leq i \leq |P|} \max_{1 \leq l \leq r} \delta_i^j$ et sélectionner la relation R_t et la classe C_d.
9 : rechercher le sommet v_d vérifiant $D_{R_t,P}(v_d) = max_{v \in C_d} D_{R_t,P}(v)$.
10 : conserver tous les sommets de l'ensemble $v \in N_{R_t}(v_d) \cup \{v_d\}$ dans la classe C_d
11 : mettre les autres dans la nouvelle classe $C_{|P|+1}$.
12 : fin tant que.

Nous allons appliquer l'algorithme 1 sur le graphe extrait à partir du réseau social classique connu sous le nom du réseau *karaté club* (Donetti et Munoz, 2004). Au cours d'une étude réalisée par le Sociologue Wayne Zachary, le club a traversé une période de turbulences due à une controverse entre l'administrateur du club et son entraîneur sur la question de l'augmentation des honoraires des adhérants du club conduisant à la création de deux classes et qui constitue une décomposition *a priori* des sommets.

La classe 1 (points noirs) du graphe (Fig 1.a) représente les individus soutenant l'entraineur (le sommet 1) qui est le sommet central de cette classe, la classe 2 (points blancs) représente les individus soutenant l'administrateur (sommet 34) sommet central de la classe. Au bout de deux itérations (Fig 1.b), l'opération de division donne naissance à deux nouvelles classes : la classe 3 est constituée d'un individu supportant l'entraineur sans interagir directement avec lui ; la classe 4 est formée par deux individus qui soutiennent l'administrateur sans être en relation directe avec lui. Les deux sommets centraux des classes 1 et 2 sont l'entraineur et l'administrateur ce qui est naturel. L'intérêt de l'introduction du concept du sommet central, est de pouvoir résumer une classe en un seul sommet représentant ce qui facilite énormément la visualisation. Concernant l'agrégation au niveau des arêtes, on peut évaluer le graphe pour construire une relation entre deux sommets autrement dit, uR_iv si la liaison entre eux est supérieure à un seuil donné.

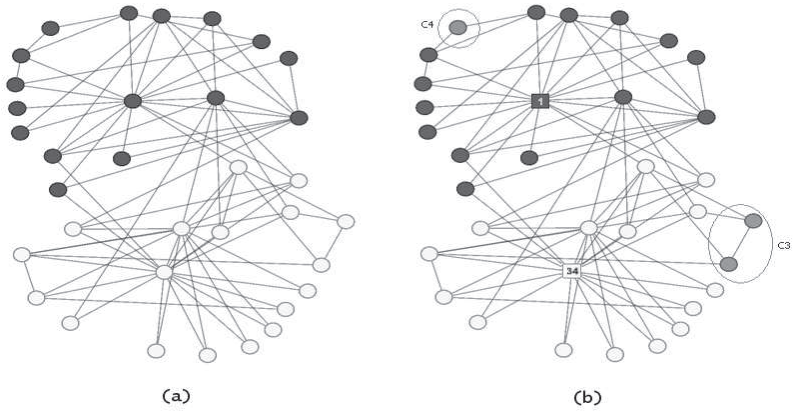


FIG. 1 – *Graphes de départ (a) et final (b) de Karaté*

Références

Chakrabarti, D., C. Faloutsos, et Y. Zhan (2007). Visualization of large networks with min-cut plots, a-plots and r-mat. *Int. J. Hum.-Comput. Stud. 65*(5), 434–445.

Donetti, L. et M. A. Munoz (2004). Detecting communities : a new systematic and efficient algorithm. *Journal of statical mechanics*.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM REVIEW 45*, 167–256.

Rodrigues Jr., J. F., A. J. M. Traina, C. Faloutsos, et C. Traina Jr. (2006). Supergraph visualization. In *ISM '06 : Proceedings of the Eighth IEEE International Symposium on Multimedia*, Washington, DC, USA, pp. 227–234. IEEE Computer Society.

Tian, Y., R. A. Hankins, et J. M. Patel (2008). Efficient aggregation for graph summarization. In *SIGMOD '08 : Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, New York, NY, USA, pp. 567–580. ACM.

Watts, D. J. et S. H. Strogatz (1998). Collective dynamics of 'small-world' networks. *Nature 393*(6684), 440–442.

Summary

Social networks can have a global vision of different actors and different interactions between them, thus facilitating the analysis and information retrieval. The network has in general a huge size which makes it difficult to analyze and visualize. An aggregation step is needed in order to have more understandable graphs. In this work, we propose an aggregation algorithm based on k-SNAP that produces a summary graph according to user-selected node attributes and relationships.

Détection de communautés: une approche par programmation DC

Brieuc CONAN-GUEZ*, Hoai An LE THI*, Manh Cuong NGUYEN*, Tao PHAM DINH**

*Laboratoire d'Informatique Théorique et Appliquée,
Université Paul Verlaine de Metz, Ile du Saulcy-Metz 57045, France
brieuc.conan-guez@univ-metz.fr,
**Laboratoire de Mathématiques de l'INSA de Rouen,
Avenue de l'Université, 76800 Saint-Etienne-du-Rouvray, France

Résumé. Afin de partitionner un réseau en communautés disjointes, un critère, appelé mesure de modularité, a été proposé en 2004. Dans cet article, nous proposons d'optimiser ce critère grâce à un nouvel algorithme: le MultistepDCA. Cet algorithme est basé sur la programmation DC (Différence de fonctions Convexes) et DCA (Algorithmes DC). Les expériences numériques montrent que le MultistepDCA est rapide et fournit des solutions de qualité.

1 Introduction

La détection de communautés dans les réseaux complexes est une problématique importante dans beaucoup de disciplines, dont les sciences de l'information (World Wide Web), la biologie (réseaux métaboliques), ou encore les sciences sociales (réseaux sociaux). Elle s'appuie sur le principe qu'un réseau peut être partitionné en sous-réseaux disjoints, appelés communautés ou modules. Les sommets composant une communauté doivent être densément connectés entre eux (liens intra), alors que le nombre de liens entre communautés doit être faible (liens inter).

En 2004, Newman et Girvan (Newman et Girvan, 2004) ont proposé une nouvelle mesure permettant d'évaluer la qualité du partitionnement d'un réseau en communautés : la mesure de modularité Q. Grâce à cette mesure, la détection de communautés peut être formulée comme un problème d'optimisation : en maximisant la fonction objectif Q, on obtient une décomposition du réseau en communautés disjointes de bonne qualité.

Dans ce travail nous proposons de maximiser la mesure de modularité grâce à la programmation DC (Différence de fonctions Convexes) et DCA (Algorithmes DC) (voir (Pham Dinh et Le Thi, 1997)). La programmation DC est une approche générale permettant de résoudre une classe très large de problèmes non convexes. Les algorithmes DC ont été appliqués avec succès à de nombreux problèmes appartenant à des domaines variés (voir `http://lita.sciences.univ-metz.fr/~lethi/DCA.html`).

2 Mesure de modularité

Comme expliqué dans l'introduction, la mesure de modularité Q est un critère quantitatif, qui évalue la qualité du partitionnement d'un réseau en communautés. La définition de cette mesure est la suivante : c'est la proportion de liens qui relient des sommets appartenant à une même communauté (liens intra) moins l'espérance de cette même quantité obtenue dans le cas d'un réseau où les liens sont générés aléatoirement mais en préservant le degré de chaque sommet (le nombre de voisins). La mesure de modularité est à valeurs dans l'intervalle $[-1, 1]$. Une valeur proche de 1 indique que la structure de communautés est très marquée.

Définissons à présent la mesure Q de manière plus formelle. On considère un graphe non orienté $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ avec n sommets ($\mathcal{S} = \{1, \ldots, n\}$), et m arêtes ($m = Card(\mathcal{A})$). La matrice d'adjacence est notée A. Le degré du sommet i, le nombre de voisins, est noté d_i ($d_i = \sum_j A_{ij}$). On note $\vec{d} = (d_1, \ldots, d_n)$. On définit la matrice de modularité : $B = A - \frac{1}{2m}\vec{d}\vec{d}^T$. B est une matrice constante qui ne dépend que du graphe $\mathcal{G}$.

On considère une partition $\mathcal{P}$ de $\mathcal{S}$ en c communautés. $\mathcal{P}$ peut être représentée par une matrice binaire d'affectation U de dimensions $n \times c$. La composante U_{ik} prend la valeur 1 si le sommet i appartient à la communauté k et 0 sinon. La mesure de modularité a alors l'expression suivante :

$$Q(U) = \frac{1}{2m}\sum_{i,j=1}^{n}\left(A_{ij} - \frac{d_i d_j}{2m}\right) < U_{i.}, U_{j.} >= \frac{1}{2m}Tr(U^T B U)$$

où $U_{i.}$ est la $i^{\text{ème}}$ ligne de U, $<,>$ est le produit scalaire de $\mathbb{R}^c$ et Tr est la trace d'une matrice.

3 Un algorithme DC pour la détection de communautés

3.1 Programmation DC

La programmation DC et les algorithmes DC (DCA) (Pham Dinh et Le Thi, 1997) permettent de minimiser une fonction objectif f qui peut s'écrire comme la différence de deux fonctions convexes : $f \equiv g - h$. g et h sont des fonctions convexes propres semi-continues inférieurement, et sont appelées des composantes DC. Un programme DC prend la forme : $\alpha = \inf\{f(x) := g(x) - h(x) : x \in \mathbb{R}^p\}$.

Une contrainte convexe $x \in \Delta$ peut être ajoutée au problème d'optimisation de la manière suivante : on considère la fonction indicatrice χ_Δ, qui est définie par $\chi_\Delta(x) = 0$ si $x \in \Delta$, et $+\infty$ sinon. En additionnant la fonction χ_Δ à la première composante DC g, on obtient un nouveau programme DC sans contraintes équivalent au programme contraint initial.

On définit la sous-différentielle de la fonction h en x_0, notée $\partial h(x_0)$, par $\partial h(x_0) \equiv \{y \in \mathbb{R}^p : h(x) \geq h(x_0) + \langle x - x_0, y\rangle, \forall x \in \mathbb{R}^p\}$. Si h est différentiable, on a $\partial h(x_0) = \{\nabla h(x_0)\}$.

L'idée principale de DCA est simple : chaque itération k de DCA approxime la seconde composante convexe h par sa minorante affine et résoud le programme convexe résultant. Si $y^k \in \partial h(x^k)$, l'équation de remise à jour de DCA est :

$$x^{k+1} \in \arg\min\{g(x) - (h(x^k) + < x - x^k, y^k >) : x \in \mathbb{R}^p\}.$$

On montre que la fonction objectif f décroit à chaque itération. Un des avantages de DCA est que cette méthode ne nécessite pas la minimisation de la fonction objectif sur une direction (*linesearch*), contrairement aux algorithmes classiques de descente de gradient.

3.2 Le MultistepDCA

Soit μ un scalaire, on a alors $Q(U) = \frac{1}{2m}Tr(U^T(B+\mu Id)U) - \frac{1}{2m}\mu n$. On pose $h(U) = \frac{1}{2}Tr(U^T(B+\mu Id)U)$. Maximiser Q est équivalent à maximiser h. Si l'on choisit $\mu > -\lambda_1(B)$, où $\lambda_1(B)$ est la plus petite valeur propre de B, $h(U)$ est une fonction convexe. Le problème d'optimisation combinatoire initial est alors équivalent au problème d'optimisation continue suivant : $\alpha = \inf\{\chi_{\Delta_c}(U) - h(U) : U \in \mathbb{R}^{n\times c}\}$, où $\Delta_c = \{U \in [0,1]^{n\times c} \mid \sum_{k=1}^{c} U_{ik} = 1\ \forall i\}$. On obtient donc une formulation DC du problème initial. Lors de l'itération k, la minorante affine de h en U^k est $l^k(U) = h(U^k) + Tr((U-U^k)^T(B+\mu Id)U^k)$. Chaque itération de DCA doit donc résoudre le problème sans contraintes : $U^{k+1} \in \arg\min\{\chi_{\Delta_c}(U) - l^k(U) : U \in \mathbb{R}^{n\times c}\}$. On a alors la formulation équivalente : $U^{k+1} \in \arg\max\{Tr(U^T(B+\mu Id)U^k) : U \in \Delta_c\}$. Ce problème est séparable en les lignes de U. On note $Y^k = (B+\mu Id)U^k$. La solution à l'itération k est alors la suivante : chaque ligne $U_{i.}^{k+1}$ prend la valeur 1 pour la composante de valeur maximale de la ligne $Y_{i.}^k$, et 0 sinon. Plus précisément, $U_{i.}^{k+1} = \vec{e}_{\arg\max_j Y_{ij}^k}$ où $\vec{e}_j$ est un vecteur de la base canonique. L'algorithme DC est alors le suivant :

Algorithm 1 L'algorithme DC pour la maximisation de la modularité

choisir une valeur initiale pour U^0
calculer $\mu = -\lambda_1(B) + \varepsilon$ {où ε est une petite valeur}
$k \leftarrow 0$
repeat
  calculer $Y^k = (B+\mu Id)U^k$
  calculer $U_{i.}^{k+1} = \vec{e}_{\arg\max_j Y_{ij}^k}, \forall i \in \{1,\ldots,n\}$
  $k \leftarrow k+1$
until convergence de U^k

Dans la pratique, ce premier schéma d'optimisation ne permet pas d'atteindre des résultats comparables à ceux fournis par les meilleures méthodes concurrentes. En effet, si le paramètre μ, qui assure la stricte convexité, prend une valeur trop importante, les changements d'affectation à chaque itération sont rares, l'algorithme s'arrête prématurément après un nombre restreint d'itérations.

Pour résoudre ce problème, nous proposons une amélioration de ce premier schéma d'optimisation : le MultistepDCA. Afin de comprendre son fonctionnement, il faut tout d'abord noter que la mesure de modularité ainsi que l'algorithme DC proposé ci-dessus peuvent être appliqués à des graphes valués. Il suffit pour cela de remplacer la matrice d'adjacence binaire par une matrice d'adjacence qui représente les valuations des arêtes. Le principe du MultistepDCA est alors simple : on considère une suite indicée par le scalaire t de matrices d'adjacence $A_t = A + tA^2$. On rappelle que la matrice A^2 contient le nombre de chemins de longueur 2 dans le graphe. On note B_t et μ_t les quantités associées à A_t.

MultistepDCA consiste à appliquer l'algorithme DC à A_t pour une valeur de t suffisamment grande ($t = 0.5$ par exemple). On réapplique alors l'algorithme DC pour une valeur de t plus petite ($t = 0.25$ par exemple). Cependant, la solution trouvée à l'étape précédente ($t = 0.5$) est utilisée comme point d'initialisation pour cette étape ($t = 0.25$). Ce shéma est réappliqué pour des valeurs de t de plus en plus petites. Dans la pratique, la séquence $(0.5, 0.25, 0)$ pour le paramètre t donne de bons résultats.

Ce nouveau schéma est efficace pour les raisons suivantes : premièrement l'ajout du terme tA^2 ne modifie pas trop la structure du problème d'optimisation (les communautés sont préservées). Deuxièmement, le paramètre t permet de contrôler μ_t : μ_t est une fonction décroissante de t. A $t = 0.5$, $\mu_{0.5}$ est proche de zéro, ce qui favorise les changements d'affectation.

4 Expériences

Nous comparons le MultistepDCA avec deux algorithmes de référence : une approche divisive *Bissection* (Newman, 2006), et une approche agglomérative *Fast-Clauset* (Clauset et al., 2004). Pour le MultistepDCA, les valeurs 5,10,20,40,80 sont testées pour le nombre de classes (avec un arrêt prématuré si le critère ne s'améliore pas). Le MultistepDCA est lancé 25 fois par valeur. Le tableau 1 indique les valeurs de la modularité pour 4 jeux de données de référence.

Réseaux	Sommets	Arêtes	MDCA	Bissection	Fast-Clauset
Karate	34	78	0.419 (0s)	0.393 (0s)	0.381 (0s)
Football	115	613	0.602 (0s)	0.513 (0s)	0.577 (0s)
Email	1 133	5 451	0.556 (1s)	0.498 (4s)	0.512 (5s)
PAP	10 617	127 564	0.417 (62s)	0.370 (264s)	0.386 (59s)

TAB. 1 – *Modularité et temps de calcul global (en secondes)*

En conclusion, on constate que le MultistepDCA obtient des solutions de qualité avec un temps de calcul réduit. Parmi les perspectives, nous souhaitons comparer le MultistepDCA à d'autres algorithmes, comme le recuit simulé ou l'algorithme *extremal optimization*.

Références

Clauset, A., M. E. J. Newman, et C. Moore (2004). Finding community structure in very large networks. *Phys. Rev. E 70*(6), 066111.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA 103*(23), 8577.

Newman, M. E. J. et M. Girvan (2004). Finding and evaluating community structure in networks. *Phys. Rev. E 69*(2), 026113.

Pham Dinh, T. et H. Le Thi (1997). Convex analysis approach to d.c. programming : Theory, algorithms and applications. In *Acta Mathematica Vietnamica*, Volume 22(1), pp. 289–355.

Summary

In order to cluster a network in separate communities, a criterion, called modularity measure, has been proposed in 2004. In this article, we propose to optimize this criterion thanks to a new algorithm: the MultistepDCA. This algorithm is based on DC (Difference of Convex function) programming and DCA (DC Algorithms). Numerical results show that the MultistepDCA is fast and has a high accuracy.

Systèmes de classes et graphes des attributs

François Brucker

Ecole Centrale Marseille,
Pôle de l'Etoile,
Technopôle de Château-Gombert,
38, rue Frédéric Joliot-Curie,
13451 MARSEILLE Cedex 20
francois.brucker@lif.univ-mrs.fr

Résumé. Nous montrons dans cette communication un exemple de représentation graphique des systèmes de classes parcimonieux via leur interprétation sous forme de table individus/attributs.

1 Introduction

Dans le domaine de la classification, les données sont souvent décrites soit par des attributs soit par une distance. Dans le premier cas, les correspondances de Galois (voir Birkhoff (1967) par exemple) permettent d'associer un treillis aux données ou aux attributs de celles-ci, ces deux treillis étant duaux. On parle alors de treillis de concepts (Ganter et Wille (1996)), chaque classe d'éléments étant une classe d'attributs (le concept) dans son treillis dual.

Dans le second cas, on a coutume d'associer un système de fermeture aux classes de la dissimilarité (ou d'une dissimilarité approchée). On retrouve ainsi un treillis (Brucker et Barthélemy (2007)) sur lequel on peut définir un ensemble d'attributs.

Nous montrons dans cette communication que lorsque le système de classes à représenter est parcimonieux (Brucker et Gély (2009), Brucker et Gély (2010)) ce qui inclut de nombreux systèmes classiques comme les hiérarchies, les hypergraphes d'intervalles ou les classes formées par un X-arbre, on peut le représenter sous sa forme de table tout en conservant la représentation de ses classes associées.

2 Systèmes de classes parcimonieux

Les systèmes de classes parcimonieux sont un cas particulier de hiérarchies faibles. Ils correspondent aux treillis sans couronne (Brucker et Gély (2010)), c'est à dire aux treillis n'admettant pas de "cycles" (la figure 1 montre une couronne pour les éléments x_1, …, x_n, y_1, …, y_n d'un treillis). Ils peuvent donc être vu, du point de vu des treillis, comme une structure minimale liant les éléments entre eux (d'ou le terme de parcimonieux) : ils sont aux treillis ce que les arbres sont aux graphes.

D'un point de vu classificatoire, ces structures peuvent donc être vues comme une façon d'ordonner les données en admettant l'empiétance des classes de façon minimale.

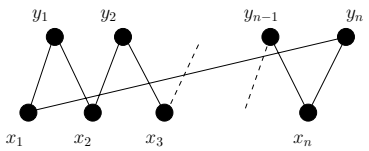


FIG. 1 – *Une couronne*

Pour obtenir la table individus/attributs à partir de données décrites par une distance (ou plus généralement une dissimilarité) on commence par approximer la distance originale par une distance dont les classes forment un treillis sans couronne (Brucker et Gély (2009)). On peut ensuite générer facilement le treillis car le nombre de classes d'une hiérarchie faible est borné par le carré du nombre d'individus.

Par exemple, les données de la table 1 sont décrites par une dissimilarité. Celle-ci est approximée par un système de de classes parcimonieux (Brucker et Gély (2009)) dont le treillis associé forme la table individus/attributs de la table 2.

	homme	bonobo	chimpanzé	gorille	orang-outang	gibbon
homme	0					
bonobo	0,19	0				
chimpanzé	0,18	0,07	0			
gorille	0,24	0,23	0,21	0		
orang-outang	0,36	0,37	0,37	0,38	0	
gibbon	0,52	0,56	0,51	0,54	0,51	0

TAB. 1 – *Distance d'évolution entre six primates (matrice triangulaire inférieure)*

	0	1	2	3	4	5	6	7	8	9	10
homme			X		X				X	X	X
bonobo			X	X				X		X	X
chimpanzé		X	X	X					X	X	X
gorille		X	X	X		X					X
orang-outang			X		X		X			X	
gibbon	X									X	

TAB. 2 – *Attributs/valeurs du système de classes parcimonieux associé à la table 1.*

3 Représentation graphique

Les systèmes de classes parcimionieux étant des hiérarchies faibles, l'intersection de trois classes est toujours l'intersection de deux d'entres elles (Bandelt et dress (1989)). Ceci se

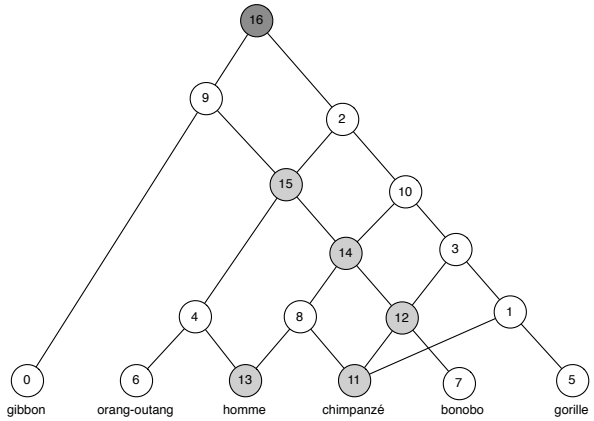


FIG. 2 – *Sup demi-treillis associé à la table 2.*

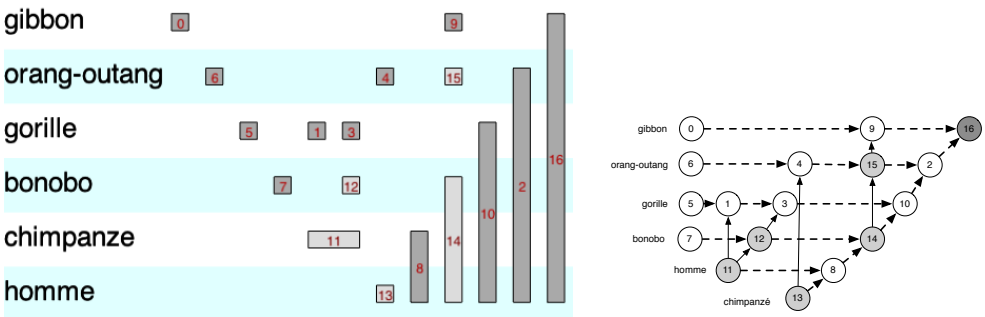


FIG. 3 – *Représentations graphiques la table 2.*

traduit sur la table individus/attributs par le fait que l'intersection de 3 colonnes sera toujours l'intersection de deux d'entres elles et donc que les éléments du treillis (*ie.* les classes) sont exactement formées des colonnes de la table et de leurs intersections deux à deux. La figure 2 montre le sup demi-treillis associé à la table 2. Les numéros de classes correspondent soit aux attributs (numéros de 0 à 10), soit à l'intersection de deux classes (de 11 à 15), soit à l'ensemble tout entier (16). Par exemple, la classe {bonobo, chimpanzé, home} (numéro 14) correspond à l'intersection des colonnes numéro 9 et 10.

Nous avons montré que l'on peut toujours réordonner des lignes et les colonnes pour que tous les éléments du treillis puissent être représenter comme dans la figure 3 (gauche). Cette figure correspond à un ordonnancement des lignes et des colonnes de la table 2 correspondant à la table 3 et chaque classe se lit de haut en bas. Ainsi la classe numéro 14 correspond à {bonobo, chimpanzé, home} et la classe 3 est, composée de toutes les classe plus basses (ici 1 et 12), correspond aux individus {gorille, bonobo, chimpanzé}.

	0	6	5	7	1	3	2	4	8	9	10
gibbon	X									X	
orang-outang		X					X	X		X	
gorille			X		X	X	X				X
bonobo				X		X	X			X	X
chimpanzé					X	X	X		X	X	X
homme							X	X	X	X	X

TAB. 3 – *Réordonancement de la table 1.*

Cette représentation permet de représenter en un unique graphique et la table individus/attributs et les classes du treillis correspondant. La représentation graphique des classes selon cet ordre est donnée en figure 3 (droite). Elle permet de plus de montrer que tout treillis sans couronne s'organise autour de deux "hiérarchies", l'une en lignes (les traits en tirets), l'autre en colonnes (les traits pleins).

Plus la table à représenter est grande, plus cette représentation est préférable à la représentation classique car il n'y a pas de problèmes de visualisations liés aux chevauchements des arêtes : elles sont soit verticales, soient horizontales.

Enfin, les structures sans couronne étant auto-duales, la transposée de la table 3 est exactement un réordonnancement de la transposée de la table 2 : que l'on s'intéresse aux individus ou aux attributs le même ordonancement est utilisé.

Références

Bandelt, H.-J. et W. M. dress (1989). Weak hierarchies associated with similarity measures - an additive clustering technique. *Bulletin of Mathematical Biology 51*, 133–166.

Birkhoff, G. (1967). *Lattice Theory (3th Edition)*. Providence, Rhode Island : American Mathematical Society.

Brucker, F. et J.-P. Barthélemy (2007). *Eléments de classification - aspects combinatoires et algorithmiques*. Hermes Science.

Brucker, F. et A. Gély (2009). Parsimonious cluster systems. *Advances in Data Analysis and Classification 3*, 189–204.

Brucker, F. et A. Gély (2010). Crown-free lattices and their related graphs. *Order, to appear*.

Ganter, B. et R. Wille (1996). *Formal Concept Analysis, Mathematical Foundations*. Springer-Verlag Berlin.

Summary

We show in this communication how to graphically represent parsimonious clustering systems through their attributes/values table.

Congruences de treillis et classifications

Vincent Duquenne*

*CNRS & Université Pierre et Marie Curie (Paris 6)
case 189 - Combinatoire et Optimisation
4 place Jussieu, 75252 Paris cedex 05
duquenne@math.jussieu.fr
http://www.ecp6.jussieu.fr/pageperso/duquenne/duquenne.html

Résumé. Les *treillis de Galois* ont été identifiés dans les années soixante comme structures pertinentes en analyse de données et classification, pour coder, manier, représenter … toute dualité par exemple les *extensions / intensions* de systèmes de concepts. Ils généralisent les *arbres*, généralisation qui se paye par la complexité de leur représentation graphique. L'objet de cette note est de montrer que l'on peut tirer profit de théorèmes classiques pour en clarifier la structure et en simplifier voire standardiser la représentation graphique.

1 Introduction

Les *treillis de Galois* ont été identifiés d'un point de vue mathémathique par Ore (1944), puis dans les années soixante par Barbut (1965), Barbut et Monjardet (1970) comme structures pertinentes en analyse de données et en classification. La raison principale de cet intérêt est qu'ils permettent de coder, manier, représenter … toute dualité *extensions / intensions* de systèmes de concepts, ce qui a été développé si ce n'est martelé depuis par Wille (1982), Ganter et Wille (1999). Ils ont reçu beaucoup d'attention sur le plan algorithmique dans le cadre de la *fouille de données*, notamment pour les bases de données de grandes tailles.

Les treillis généralisent les arbres (qui sont des *semi-treillis* particuliers), du fait aussi qu'un parcourt de la relation de *couverture* (voisins immédiats) d'un treillis suivant la relation d'*ordre* « déplie » un arbre (avec répétition des éléments rencontrés plusieurs fois…). Maintenant, cette généralité est souvent un handicap, car les treillis sont difficiles à représenter graphiquement, surtout de manière qui soit canonique et ne souffre pas d'arbitraire.

2 Un exemple de pédagogie mathématique

Les données viennent de la Thèse de Camilio Charon (1998), et concernent la pédagogie des mathématiques dans l'enseignement élémentaire (pour des élèves de moyenne / grande sections maternelles et cours préparatoire). Les enfants sont évalués pour leur maîtrise *de propriétés des nombres entiers* (*ordre*, *égalité*, *addition*, … voir Fig. 1). Elles ont alors été analysées par un treillis de Galois global, mélangeant les groupes, et en termes « intensionels » par la *base canonique d'implications* (Guigues et Duquenne 1986, Duquenne 1999) qui en résume de manière exhaustive et canonique les implications entre « propriétés » (voir aussi Kuznetsov et Obiedkov (2008) pour la complexité du calcul de cette base).

Plus récemment (Duquenne 2007) ces analyses ont été raffinées par groupe d'âge, et en terme de développement par l'introduction d'une *base relative d'implications* exprimant spécifiquement quelles sont les implications invalidées pour les trois groupes d'âges mélangés et qui étaient valides pour les deux groupes de moyenne et grande section maternelles.

Nous prendrons ici un point de vue « extensionel » -classification oblige…- en posant la question suivante : comment tirer profit du treillis de Galois et des propriétés classiques des treillis pour structurer / hiérarchiser les élèves par la maîtrise des « propriétés » concernées ?

Traiter pleinement cette question nécessite de réels progrès pour la représentation graphique des treillis de Galois, vu d'une part leur complexité et d'autre part leur explosion combinatoire. Des propositions existent dans la littérature, basées sur des propriétés structurelles (dessins respectant des *règles de parallélisme*, utilisation de *produit de semi-treillis*…), soit sur des algorithmes dynamiques mais non structuraux (« *spring embedder* » Freese (2004)).

Nous suivrons ici une voie plus structurelle encore basée sur les notions de *congruences / homomorphismes* et sur les *éléments inf-irréductibles* (générateurs) du treillis, et surtout sur la combinaison de deux théorèmes classiques et fondamentaux (cf. Birkhoff (1970)) :

- Le treillis des *filtres* (parties héréditaire supérieures) $F(P)$ d'un ensemble ordonné $(P, \leq)$ est *distributif*. Réciproquement, un treillis distributif $(D, \leq, \wedge, \vee)$ est dualement isomorphe au treillis des filtres de l'ensembles ordonné de ses *éléments inf-irreductibles* $(M(D), \leq)$.
- Une *congruence* sur un treillis est une *relation d'équivalence* qui respecte sa *relation d'ordre* $\leq$ et ses deux *opérations* $\wedge$ et $\vee$, et est la préimage d'un *homomorphisme* de treillis.

Pour tout treillis fini $(L, \leq, \wedge, \vee)$, le treillis de ses congruences Con(L) est distributif, et peut donc être résumé exhaustivement par ses congruences inf-irreductibles $(M(Con(L)), \leq)$.

L'idée simple (introduite et décrite plus en détail dans Duquenne (2010)) est de représenter le treillis L quotienté par sa congruence de *Frattini*, qui est l'intersection des congruences maximales de $(M(Con(L)), \leq)$ (ou plus généralement par une série de congruences emboitées correspondant à un « shelling » descendant -ou ascendant- de $(M(Con(L)), \leq)$, ce qui n'est pas sans rappeler les analyses hiérarchiques descendantes, ou ascendantes…).

3 Conclusion

Maintenant la question légitime en miroir, qui intéresse plus les spécialistes du domaine ou de l'analyse de donnée est bien sûr : qu'est-ce qu'on y gagne ? Quelles sont les interrogations et les réponses que peuvent porter un tel treillis de Galois étiqueté et ainsi quotienté ?

- Ici, pour cet exemple, la congruence de Frattini quotiente le treillis en un *treillis distributif de classes* (le cas n'est pas rare dans la pratique, voir Duquenne 2010), engendré par les « propriétés » A,D,E,F,J, et les deux relation d'ordre F,J<A, voir Fig. 1. Ces cinq propriétés sont chacune *complètement indépendantes entre elles* et *de toutes les autres* vis à vis de la structure globale du treillis / groupe d'élèves observés, mais indiquent que F:différence et J:commutativité supposent que l'élève a déjà acquis A:order.
- Par contre les propriétés B,C,G,I sont *complètement et symétriquement dépendantes* les unes des autres, et *dépendantes* des propriétés précédentes, alors que la propriété H:counting est *dépendante* de toutes les autres propriétés. Ce shelling descendant (ou montant, les deux ici coïncident) définit un *préordre* de dépendances sur M(L) : H < (B~C~G~I) < A,D,E,F,J, dont il faudrait tenir compte dans l'ordre ultérieur de présentation des exercices. Faire en sorte que tous les élèves dominent H:counting (ce qui identifiera H:counting et sa couverture supérieure) « bouscule » moins le treillis que çà n'est le cas pour les propriétés A,D,E…

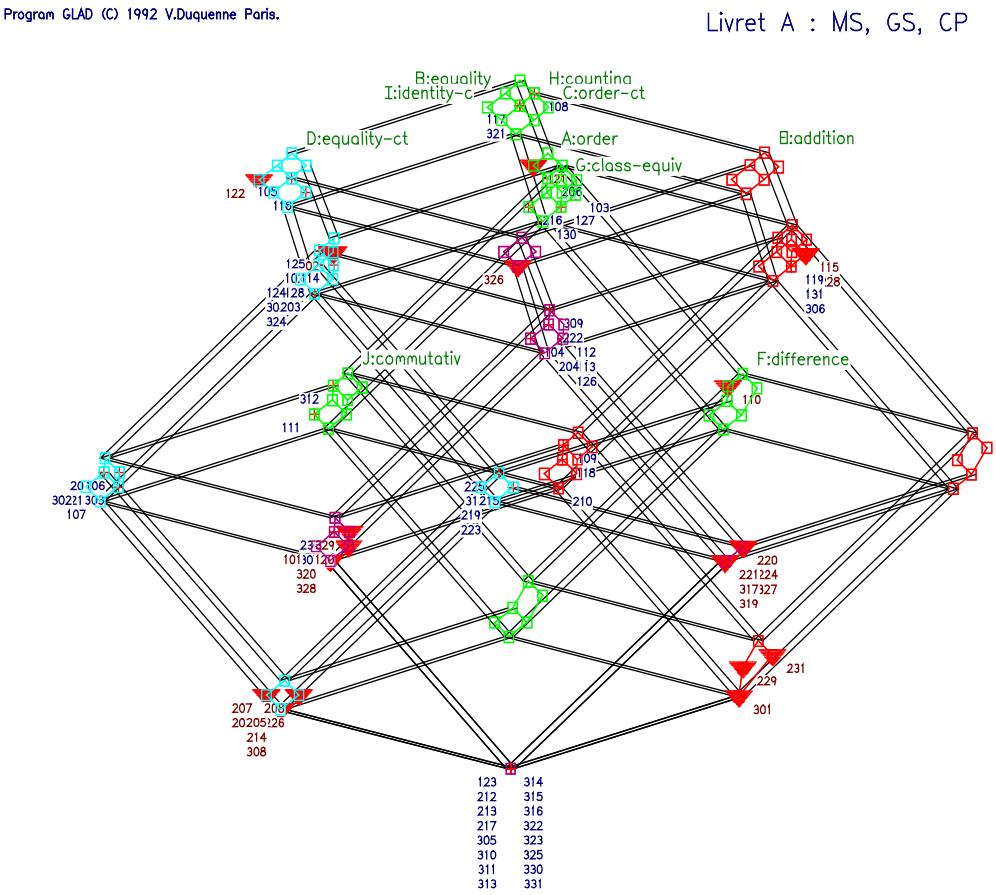


FIG. 1- Trois groupes d'élèves sont décrits par des variables sur la maîtrise de propriétés des nombres entiers. La congruence de Frattini utilisée pour quotienter le treillis de Galois définit un treillis de classes (reliées par des double traits) qui est ici *distributif*, ce qui révèle une hiérarchie simple entre les propriétés *indépendantes* vis–à-vis de la structure globale des données, et qui est « contrôlée » par l'implication : *Commutativité, Différence* $\rightarrow$ *Ordre*.

• Sur le plan du spécialiste du domaine et de la pédagogie, cela donne l'idée de constituer des groupes de travail –ordonnés hiérarchiquement- mélangeant les élèves des trois âges « tombant » dans la même classe, et où l'on ferait travailler d'abord *toutes* les propriétés (H, et B,C,G,I), puis les propriétés du premier groupe (A,D,E,F,J) –dans un ordre quelconque puisqu'elles sont globalement indépendantes les unes des autres-, de telle sorte que les élèves progressent de façon constante et certaine, le long de la relation de couverture du treillis de Galois, le but étant que tous arrivent, après apprentissage, « à tout savoir » au 0 du treillis...

• Le treillis de Galois dont la structure a ainsi été clarifiée par la congruence de Frattini acquiert ici un statut de « modèle de groupe » multidimensionnel, séparant nettement les propriétés *complètement indépendantes* et celles qui sont *complètement dépendantes* entre elles (et *dépendantes des précédentes*), même s'il est toujours possible sur le même treillis étiqueté d'y restaurer la performance individuelle des élèves, et de les situer dans le groupe.

4 Références

Barbut, M. Note sur l'algèbre des techniques d'analyse hiérarchique. Appendice de: *L'analyse hiérarchique* (B. Matalon). Gautiers-Villars, Paris 1965, 125-146.

Barbut, M. and B. Monjardet. *Ordre et classification. Algèbre et Combinatoire* (1970), 2 tomes. Paris, Hachette.

Birkhoff, G. *Lattice Theory* (1967). American Mathematical Society Colloq. Publ. Vol.25, third edition, Amer. Math. Soc., New York, N. Y. (first ed. 1940).

Charron, C. Ruptures et continuités dans la construction des nombres, *Phd Thesis* (1998), Université Paris V-René Descartes, Paris.

Duquenne, V. Latticial structures in Data Analysis, *ORDAL 96*: *Order and decision-making* (I. Rival ed.), Ottawa, www.csi.uottawa.ca, and *Theoretical Computer Science 217* (1999) 407-436.

Duquenne, V. What can lattices do for teaching math.?, in *CLA'07* (J. Diatta, P. Eklund, M. Liquière eds), (2007) 72-87, posted at: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-331/ .

Duquenne, V. Lattice Drawings and Morphisms. *ICFCA'2010* (L. Kwuida, B. Sertkaya eds), *LNAI 5986* (2010) 88-103, posted at : http://www.ecp6.jussieu.fr/pageperso/duquenne/duquenne.html .

Freese, R. Automated lattice drawing, in *ICFCA'2004* (P. Ecklund ed.), *LNAI 2961* (2004) 112-127, and posted at http://www.latdraw.org/.

Ganter B. and R. Wille. *Formal Concept Analysis, Mathematical Foundations*. Springer Verlag, Berlin, 1999.

Guigues J.L. and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathematiques & Sciences Humaines 95* (1986) 5-18 (*preprint* Groupe Mathématiques et Psychologie, Université Paris V-René Descartes, 1984).

Kuznetsov, S. O. and S. A. Obiedkov. Some decision and counting problems of the Duquenne-Guigues basis of implications. *Discrete. Applied Mathematics* 156 (11): 1994-2003 (2008).

Ore, O. Galois connexions, *Trans. Amer. Math. Soc. 55* (1944) 493-513.]

Wille, R. Restructuring lattice theory: an approach based on hierarchies of concepts. In *Ordered sets* (I. Rival ed), Reidel, Dordrecht-Boston (1982) 445-470.

Summary

By the end of the sixties *Galois lattices* have been identified as pertinent structures of data analysis and classification methods, due to their potentialities to code, manipulate, represent … any duality, such as the duality between *extensions / intensions* of concept systems. As *semi-lattices*, they generalize *trees*, which often involves a complexity of their graphic representation. The goal of this note is to illustrate that classical theorems can be put together to become instrumental in clarifying and standardizing their graphic representation.

Un nouvel algorithme pour la détection des transferts horizontaux de gènes partiels entre les espèces et pour la classification des transferts inférés

Alix Boc*, Alpha Boubacar Diallo*, Vladimir Makarenkov*

*Département d'informatique, Université du Québec à Montréal
Case postale 8888, Succursale Centre-ville, Montréal (Québec) H3C 3P8 Canada

Résumé. Nous présentons un nouvel algorithme pour la détection et la validation des transferts horizontaux de gènes (THG) partiels. L'algorithme proposé se base sur une procédure de fenêtre coulissante qui analyse les fragments d'un alignement de séquences. Une procédure de validation par bootstrap, permettant d'évaluer le support statistique des THG partiels obtenus, a été introduite. Le nouvel algorithme peut être utilisé pour confirmer ou rejeter les transferts complets détectés à l'aide de n'importe quel algorithme de détection des THG ainsi que pour classifier les transferts retrouvés (en tant que complets ou partiels).

1 Introduction

Les bactéries et les archées s'adaptent à différentes conditions environnementales via la formation de gènes mosaïques. Le terme "mosaïque" découle de la configuration des blocs entrecoupés de séquences ayant des histoires évolutionnaires différentes, mais se trouvant combinés dans le gène résultant suite à des évènements de recombinaison intragénique (Gogarten et al. (2002)). Les segments recombinés peuvent être dérivés d'autres souches de la même espèce ou des espèces différentes. Le modèle du transfert horizontal complet suppose que soit le gène transféré supplante le gène orthologue entier dans le génome receveur, soit, si le gène transféré est absent du génome receveur, il lui est ajouté (Boc et al. (2010)). Le second modèle, celui du transfert partiel, implique la formation des gènes mosaïques. Un gène mosaïque est formé à travers les mécanismes de la transformation et de la conjugaison qui permettent l'acquisition et l'intégration subséquente de fragments d'ADN provenant des organismes distincts. Alors que plusieurs méthodes ont été proposées pour l'identification et la validation des THG complets (Page (1994) ; Mirkin et al. (1995) ; Hallett et Lagergren (2001) ; Tsirigos et Rigoutsos (2005) ; Than et Nakhleh (2008) ; Boc et al. (2010)), seulement deux méthodes traitent le problème de l'inférence de THG partiels (Denamur et al. (2000) et Makarenkov et al. (2006)). Toutefois, les deux derniers travaux ne considèrent pas la problématique de la validation des transferts partiels obtenus et n'incluent pas de simulations Monte-Carlo. Dans les faits, aucune méthode fiable pour l'identification des gènes mosaïques et des transferts horizontaux de gènes partiels associés n'a été proposée jusqu'à maintenant.

2 Algorithme pour la détection des transferts partiels

Les principales étapes de l'algorithme, qui cherche à produire un scénario optimal de transferts partiels (i.e., comportant le nombre minimal de THG) d'un gène donné, sont résumées ci-dessous. Une validation par bootstrap est effectuée pour chaque transfert partiel obtenu et seuls les transferts significatifs (i.e., ayant le pourcentage de bootstrap supérieur à un seuil donné) sont inclus dans la solution finale. Une procédure de fenêtre coulissante est utilisée pour tester différents fragments de l'alignement de séquences multiples (ASM) donné.

Algorithm 1 Détection des transferts partiels

Require: X : un ensemble d'espèces étudiées.
ASM : un alignement de séquences multiples de taille l.
$S_{i,j}$: le fragment de l'ASM, étant analysé, situé entre les sites i et j, où $1 <= i < j <= l$.
w : la taille de la fenêtre coulissante ($w = j - i + 1$).
s : la taille du pas de progression.

1: Inférer l'arbre phylogénétique d'espèces T. L'arbre T doit être enraciné.
2: Fixer la taille de la fenêtre coulissante w et la taille du pas s.
3: **for** $k = 1 \rightarrow l - w$; $k \leftarrow k + s$ **do**
4: $i \leftarrow 1 + s(k - 1)$
5: $j \leftarrow i + w - 1$
6: Inférer avec PhyML (Guindon et Gascuel (2003)) un arbre de gène partiel T'caractérisant l'évolution du fragment de l'ASM localisé dans l'intervalle $[i, j]$.
7: Appliquer un algorithme de détection existant pour inférer un scénario de THG partiels associés à l'intervalle $[i, j]$. L'algorithme HGT-Detection (Boc et al. (2010)) a été utilisé ici pour inférer des transferts complets.
8: Exécuter la procédure pour évaluer la fiabilité des transferts partiels obtenus. Cette procédure, basée sur le principe de bootstrap, prend en compte l'incertitude des arbres de gène partiels, ainsi que le nombre de fois qu'un transfert donné apparait dans tous les scénarios de coût minimum (i.e., comportant le nombre minimum de THG nécessaires pour réconcilier les arbres T et T').
9: **end for**

La complexité de cet algorithme est comme suit : $O(r * (\frac{(l-w)}{s} * (C(PhIn) + \tau * n^4)))$, où w est la taille de la fenêtre coulissante, s est le pas de progression, $C(PhIn)$ est la complexité de la méthode d'inférence d'arbres phylogénétiques (e.g., PhyML) utilisée pour inférer les phylogénies à partir des fragments de séquences situés dans la fenêtre coulissante, r est le nombre de réplicats dans le bootstrap, n est le nombre d'espèces et τ est le nombre moyen de transferts horizontaux détectés pour un fragment de séquences de taille w.

Des simulations Monte-Carlo ont été effectuées pour tester l'efficacité du nouvel algorithme dans le contexte des THG partiels. La procédure de simulations incluait les étapes suivantes. Des arbres d'espèces binaires avec 8, 16, 32 et 64 feuilles ont été créés en utilisant la procédure de génération d'arbres aléatoires de Kuhner et Felsenstein (1994). Nous avons ensuite exécuté le programme SeqGen (Rambaut et Grassly (1997)) pour générer des ASM de protéines le long des arêtes des arbres d'espèces construits à la première étape. Des séquences de protéines avec 500 acides aminés ont été générées. Puis, pour chaque arbre d'espèces T,

nous avons généré des arbres de gène T', ayant le même nombre de feuilles, en effectuant des déplacements aléatoires de ses sous-arbres. Pour chaque arbre d'espèces, 1 à 5 déplacements aléatoires de sous-arbres ont été effectués et différents arbres de gène T', englobant entre 1 et 5 THG partiels, ont été générés. Nous avons fixé la taille de chaque séquence transférée à 200 acides aminés. Finalement, nous avons exécuté l'algorithme pour chaque arbre d'espèces généré et l'ASM associé qui était affecté par les THG partiels. La taille de la fenêtre coulissante a été fixée à 100, 200, 300, 400, puis 500 acides aminés ; 100 réplicats de chaque arbre de gène partiel T'ont été générés pour évaluer le support de bootstrap des arêtes de T'dans un premier temps, puis le support des THG partiels obtenus dans un deuxième temps. Parmi les THG obtenus, seuls les transferts avec un bootstrap supérieur à 90% ont été retenus. Finalement, nous avons estimé le taux de détection (les vrais positifs seulement) et le taux de faux positifs. Les performances moyennes obtenues par le nouvel algorithme sont illustrées sur la figure 1. Cette figure met en lumière les différences entre le taux de détection moyen et le taux de faux positifs moyen en fonction du nombre d'espèces. La moyenne ici était calculée à partir des résultats obtenus pour les arbres englobant 1 à 5 THG générés. Alors que le taux de détection moyen était toujours supérieur à 70% (79,6% en moyenne), le taux moyen de faux positifs était toujours inférieur à 40% (30,8% en moyenne).

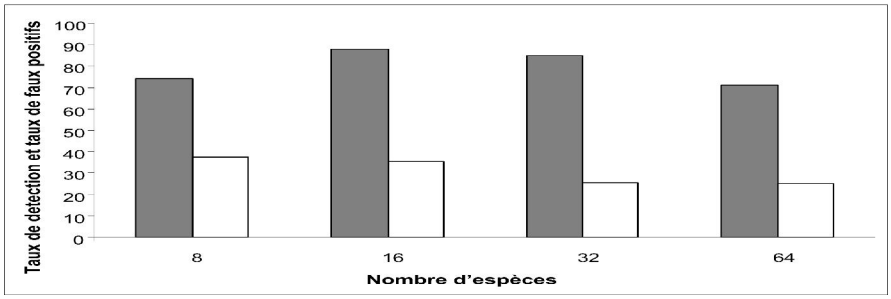


FIG. 1 – *Taux moyens de détection (vrais positifs - colonnes grises et faux positifs - colonnes blanches) des transferts partiels générés.*

3 Conclusion

Nous avons présenté un nouvel algorithme pour la prédiction des THG partiels et pour l'identification des gènes mosaïques. Au meilleur de notre connaissance, ce problème d'actualité n'a pas été convenablement traité dans la littérature. L'algorithme proposé se base sur une procédure de fenêtre coulissante pour analyser les fragments de l'ASM. La taille de la fenêtre coulissante doit être ajustée en fonction des informations existantes sur les gènes et les espèces étudiés. Une procédure de validation, permettant d'évaluer la robustesse des transferts partiels obtenus et prenant en compte l'incertitude des arbres de gène partiels, a aussi été développée. Le nouvel algorithme peut être utilisé pour confirmer ou exclure les transferts complets inférés avec n'importe quel algorithme de détection des THG ainsi que pour classifier les transferts retrouvés (en tant que complets ou partiels). Cet algorithme peut aussi être appliqué à une échelle génomique pour estimer la proportion de gènes mosaïques dans des génomes des espèces étudiées de même que pour déterminer le taux de transferts complets et partiels entre

ces espèces. L'algorithme proposé a été inclus dans le package T-REX (Makarenkov (2001)) disponible sur : www.trex.uqam.ca.

Références

Boc, A., H. Philippe, et V. Makarenkov (2010). Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Syst. Biol. 59*, 195–211.

Denamur, E., G. Lecointre, P. Darlu, O. Tenaillon, C. Acquaviva, C. Sayada, I. Sunjevaric, R. Rothstein, J. Elion, F. Taddei, M. Radman, et I. Matic (2000). Evolutionary implications of the frequent horizontal transfer of mismatch repair genes. *Cell 103*, 711–721.

Gogarten, J. P., W. F. Doolittle, et J. G. Lawrence (2002). Prokaryotic evolution in light of gene transfer. *Mol. Biol. Evol. 19*, 2226–2238.

Guindon, S. et O. Gascuel (2003). A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol. 52,5*, 696–704.

Hallett, M. et J. Lagergren (2001). Efficient algorithms for lateral gene transfer problems. *RECOMB 2001. ACM Press, New-York*, 149–156.

Kuhner, M. et J. Felsenstein (1994). A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol. 11*, 459–468.

Makarenkov, V. (2001). T-rex: reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics 17*, 664–668.

Makarenkov, V., A. Boc, C. F. Delwiche, A. B. Diallo, et H. Philippe (2006). New efficient algorithm for modeling partial and complete gene transfer scenarios. *IFCS-2006. Data Science and Classification. Springer*, 341–349.

Mirkin, B. G., I. Muchnik, et T. F. Smith (1995). A biologically consistent model for comparing molecular phylogenies. *J. Comp. Biol. 2*, 493–507.

Page, R. D. M. (1994). Maps between trees and cladistic analysis of historical associations among genes, organism and areas. *Syst. Biol. 43*, 58–77.

Rambaut, A. et N. C. Grassly (1997). Seq-gen: an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. *Comput. Appl. Biosci. 13*, 235–238.

Than, C. et L. Nakhleh (2008). Spr-based tree reconciliation: Non-binary trees and multiple solutions. *Proc. of the 6th Asia Pacific Bioinformatics Conf.. Kyoto, Japan 13*, 251–260.

Tsirigos, A. et I. Rigoutsos (2005). A new computational method for the detection of horizontal gene transfer events. *Nuc. Acids Res. 33*, 922–933.

Summary

In this article, we describe a new algorithm for detecting and validating partial horizontal gene transfers (HGT). The presented algorithm is based on a sliding window procedure which analyzes fragments of the given multiple sequence alignment. A bootstrap procedure incorporated in our method can be used to estimate the support of each inferred partial HGT. The new algorithm can be also applied to confirm or discard complete (i.e., traditional) horizontal gene transfers detected by any HGT inferring algorithm.

Protein sequence classification: a comparative study of HMM classifier

Wajdi Dhifli*, **, ***, Rabie Saidi*, **, ***, Jalel Akaichi***, Engelbert Mephu Nguifo*, **

*Clermont Université, Université Blaise Pascal, LIMOS, BP 10448, F-63000 Clermont-Ferrand, France
**CNRS, UMR 6158, LIMOS, F-63173 Aubière, France
{dhifli, saidi, mephu}@isima.fr
*** Université de Jendouba, Faculté des sciences juridiques, économiques et de gestion, Jendouba, Tunisie
jalel.akaichi@isg.rnu.tn

Abstract. Many algorithms and techniques have been proposed to address the problem of protein classification. In this context, HMM have been used for ages and are known to be good modeling tool. In this paper we detail the setup of an HMM based classifier. Then, we compare its results with those of other classifiers and techniques used in literature. Moreover, we evaluate the effect of data characteristics on HMM accuracy.

1 Introduction

Generally speaking, proteins do everything in the living cell. All functions of the living organisms are related to proteins. Hence, the task of protein classification is very important in bioinformatics since it reveals important information such as the function it plays in the living cell. This has been a main concern for many researchers, therefore, several techniques have been proposed in literature dealing with this task. In Saidi et al. (2010), authors were combining different encoding methods with some well-known machine learning classifiers then compared the results with those of the alignment based classification (supervised classification) using Blast (Altschul et al. 1990). They also used in their experiments five datasets of protein sequences with very dissimilar characteristics in term of size, number of classes, sequences identity, etc… Thus, these datasets can represent a good benchmark for our work. In this paper, we are trying to extend Saidi et al. (2010) by establishing a Hidden Markov Models (HMM) based classifier and using it to classify the same datasets. Then we will compare its results with those of other classification techniques already cited in the same paper. Furthermore, we are evaluating the effect of the dataset characteristics variation on HMM accuracy.

2 Protein classification techniques

Generally, the proposed protein classification techniques in literature can be divided in two main categories namely the alignment based approach and the machine learning based

approach. In the alignment based approach, we use a group of proteins called the reference sequences which we already know their classes to predict the class of an unknown protein. In other words, a query protein sequence takes the same class of the reference sequence having the best hit score. The machine learning based approach is based on the use of panoply of well-known classifiers (Cornuéjols et al. 2010) for instance naïve bayes (NB), support vector machine (SVM), decision trees (DT) and nearest neighbour (NN) which have proven their efficiency in many fields (such as economy). In fact, it is not evident to directly use these classifiers with protein sequences, since they operate on inputs in relational format however the latter are represented by strings of characters (Saidi et al. 2010). For that reason, it is required to perform a preprocessing step to enable the use of these classifiers in protein classification.

3 Experimentations

3.1 Datasets

As mentioned before, we used five datasets (DS) in our experiments. Each of these datasets represents a different challenge since they have dissimilar characteristics. DS1 comprises three distinct and distant protein families (classes), whereas DS2 contains two big size families that make part of the Rhodopsin Like/Peptide family. DS3 is supposed to be the most challenging dataset since it is composed of seven unbalanced classes having low sequence identity and regrouped based on their quaternary structure, so here we are trying to recognize the quaternary protein structure from its primary structure. DS4 contains two families namely the human Toll-like Receptors (TLR) protein sequences and the non-human ones. The challenge here is due to the structural and functional similarity of the two groups. DS5 consists of 277 domains. This dataset was mostly used for structural class prediction. For a full description of the datasets, refer to Saidi et al. (2010).

Moreover, we measured some features of each dataset, for instance intra-class and intra-class identity, and tried to investigate the relation between their variation and the variation of HMM results. Table 1 illustrates the measured values of datasets features.

Dataset	Total size	Class size standard deviation	Average intra-class identity	Average inter-class identity
DS1	60	1	41,57	33,98
DS2	510	0	48,28	36,36
DS3	717	124,73	25,02	26,74
DS4	40	8,48	28,23	28,27
DS5	277	8,65	88,31	36,29

TAB. 1 – *Dataset characteristics*

3.2 Experimental setup

In (Saidi et al 2010), each classifier was executed multiple runs then compared them later with Blast's results. As for our experiments, we programed an HMM classifier based on HMMER (John-son et al. 2010). The key idea very similar to the alignment based approach

and the machine learning based approach. In the alignment based approach, we first create an HMM-profile for each protein group (class), then we score the query sequence against all the created pro-files, and the query protein sequence takes the class of the HMM-profile having the best hit score. We performed multiple runs varying its parameters such as the weighting algorithm and the effective sequence number. Experiments were run under Linux using 2 GB Intel core 2 duo processor and 4 GB RAM DDR3. We used the leave one out (LOO) technique for evaluation as in (Saidi et al 2010). Blast and Weka (Bouckaert et al. 2010) classifiers were used with default values. Figure 1 illustrates the process for our HMM based protein classifi-cation. For the whole, we only considered the highest obtained accuracies.

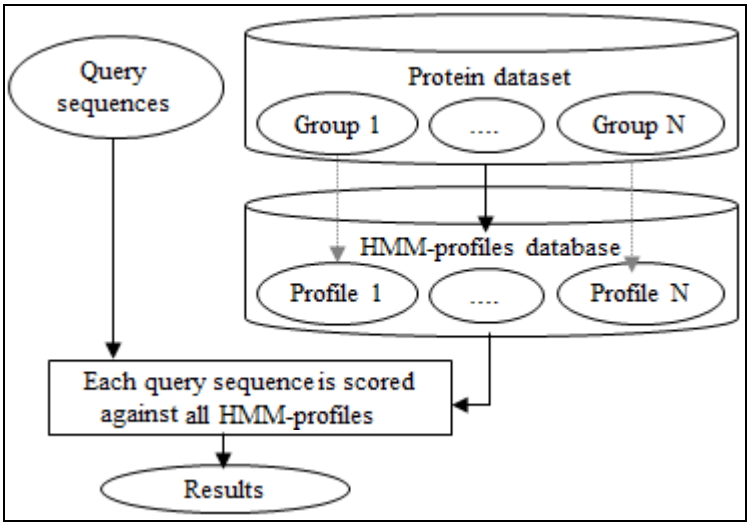


FIG. 1 – Experimental process

4 Results and discussion

Dataset	HMM	Blast	C4.5	SVM	NB	NN
DS1	**100**	**100**	96,7	96,7	90	78,3
DS2	99,21	**100**	99,8	**100**	**100**	**100**
DS3	28,73	69.60	**79,2**	78,94	59,4	77
DS4	70	78.57	82,5	87,5	**95**	80
DS5	82,67	78.3	75,5	84,1	**85,9**	80,5

TAB. 2 –*Classification accuracies*

The obtained results are illustrated in table 2. In DS1, DS2 and DS5 all classifiers including HMM scored very high since these datasets presented good inter-class and intra-class identity and close class sizes. DS3 was a real challenge since it comprised seven unbalanced classes with low intra-class identity. Therefore, HMM did not score as well as in the other datasets and its accuracy decreased significantly in DS3, since the generated profiles were

poor with very low discrimination power. As well, DS4 classification was not an easy task, the intra-class identity was low too but class sizes were close, though no full accuracy was reached however HMM and all the other classifiers performed well.

5 Conclusion

To conclude, this study shows that using HMM as classifier represents a competitive approach since it even rich full accuracy in some cases. However, it may fail especially with datasets having unbalanced classes, low intra-class, and/or high inter-class identity. This yields generating poor profiles and decreases the discrimination power of the classifier. Therefore, studying the data characteristics before choosing the classifier would be a wise decision.

References

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215:403-413.

Bouckaert, R. R., E. Frank, M. A. Hall, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten (2010). WEKA-experiences with a java open-source project. *Journal of Machine Learning Research*, 11:2533-2541.

Cornuéjols A., L. Miclet, et Y. Kodratoff (2010). *Apprentissage artificiel, 2ème Ed*. Eyrolles.

Johnson, L. S., S. R. Eddy, and E. Portugaly (2010). Hidden Markov model speeds heuristic and iterative HMM search procedure. *BMC Bioinformatics*, 11: 431-438.

Saidi, R., M. Maddouri, and E. Mephu Nguifo (2010). Protein sequences classification by means of feature extraction with substitution matrices. *BMC Bioinformatics*, 11:175-187.

Acknowledgments

This work has been partially supported by a scholarship awarded by the Tunisian government to the first author.

Résumé

Plusieurs algorithmes et techniques ont été proposés pour traiter le problème de classification des séquences protéiques. Dans ce contexte, les HMM sont connus en tant qu'un bon outil de modélisation. Dans cet article, nous présentons la mise en œuvre d'un classifieur HMM et ses résultats sur la classification des séquences protéiques, et nous comparons avec celles d'autres méthodes de la littérature. En plus, nous évaluons l'effet des caractéristiques des données sur le taux de classification du classifieur HMM.

Segmentation de séries temporelles avec prise en compte a priori de composantes de variance

Christian Derquenne

EDF R&D – 1, avenue du Général de Gaulle – 92141 Clamart Cedex
christian.derquenne@edf.fr

Résumé. La méthode proposée permet de segmenter une série temporelle en deux phases. La première consiste à exhiber la variabilité de la série à l'aide d'une première segmentation sur un signal adéquat, puis de tenir compte de la structure de dispersion dans la seconde phase qui offrira une segmentation à l'aide d'un modèle linéaire gaussien hétéroscédastique.

1 Problématique

Les séries temporelles se décomposent généralement en plusieurs types d'évolution : tendance, saisonnalité, volatilité et bruit. Elles peuvent être plus ou moins régulières selon le domaine d'application. Les changements de comportements qui caractérisent principalement ces séries sont de plusieurs types : pics, sauts en niveau, en tendance, en variabilité. La modélisation de ces séries est donc très délicate et demande beaucoup d'expérience. Il peut alors être intéressant de détecter des ruptures de comportement pour la construction de sous-modèles, la stationnarisation de la série, la construction de courbes symboliques pour la classification de courbes. De nombreuses méthodes de segmentation dans Guédon (2008), Lavielle (2009), Perron et al (2006) ont été et sont développées pour répondre à ces différentes problématiques en économie, en finance, en séquençage humain, en météorologie, en management de l'énergie, etc. La plupart de ces méthodes reposent sur l'utilisation de la programmation dynamique pour diminuer drastiquement le nombre de segmentations possible. Ces méthodes de détection de points de rupture ont pour vocation de résoudre trois problèmes, voir Lavielle (2009) : (i) la détection de changement de la moyenne, avec une variance constante, (ii) la détection de changement de variance avec une moyenne constante et (iii) la détection de changements dans l'ensemble de la distribution du phénomène étudié. La méthode introduite par Derquenne (2011A) permet non seulement de réduire la complexité par rapport à d'autres méthodes, mais surtout de proposer des solutions de segmentation de la série contenant des segments croissants, décroissants, constants et des dispersions différents. Notre méthode est originale dans son approche car elle propose, par étapes successives, une aide à la décision pour la segmentation des données. Cependant, il s'avère, pour l'ensemble des méthodes issues d'approches par programmation dynamique ou exploratoire comme la nôtre, que la qualité de la segmentation peut faire défaut lors de la détection de segments contigus quand les niveaux (constants ou pentes) sont proches statistiquement mais ont des variances différentes. Dans ce cas un seul segment sera détecté, alors qu'il y en a deux structurellement. Par conséquent, nous proposons une nouvelle méthode améliorant la précédente. Cette nouvelle approche permet d'estimer préalablement la dispersion par segmentation, puis l'intègre dans une seconde phase de segmentation pour obtenir le résultat final. Afin de tester notre approche, nous avons alors réalisé une étude comparative avec des algorithmes de programmation dynamique proposés dans Lavielle (2009).

2 Proposition de la nouvelle approche

Soit une série temporelle $(Y_t)_{t=1,T}$, nous supposons qu'elle se décompose selon le modèle linéaire gaussien hétéroscédastique (MLGH) en S segments tel que :

$$Y_t = \sum_{s=1}^{S} (\beta_0^{(s)} + \beta_1^{(s)} t + \sigma_s \varepsilon_t) 1_{[t \in \tau_s]} \quad (1)$$

où $\beta_0^{(s)}$, $\beta_1^{(s)}$ et σ_s>0 sont respectivement les paramètres de niveau, de pente et de dispersion pour le segment τ_s, et $\varepsilon_t \sim> \mathcal{N}(0,1)$. Il y a $3S$ paramètres à estimer et S est inconnu. Pour estimer le MLGH, nous avons utilisé le maximum de vraisemblance restreint (REML).

L'approche introduite par Derquenne (2011A) contient deux phases : préparation des données, puis modélisations successives fondées sur le modèle (1). La première phase se décompose en trois étapes pour obtenir une première segmentation : lissage, différenciation et comptage. Celle-ci peut contenir beaucoup trop de segments, alors pour la résumer au mieux nous utilisons (1) à l'aide de la phase de modélisation.

La nouvelle approche contient trois phases. La première consiste à établir une transformation adéquate des données afin d'obtenir une nouvelle série caractérisant l'évolution temporelle de la dispersion des observations, la deuxième revient à segmenter cette nouvelle série avec le même principe que notre précédente méthode pour obtenir des segments de dispersion, enfin la troisième applique à nouveau notre précédente méthode en tenant compte de la distribution des segments de dispersion, notamment lors de la construction du MLGH.

Phases 1 et 2 : Transformation des données et première segmentation. Nous construisons une nouvelle série temporelle exhibant la volatilité des observations de Y_t dont on suppose qu'elles sont régies par le modèle (1). La transformation la plus naturelle est telle que : $Z_t = (1\text{-B})^2 Y_t$, où Y_t est la série temporelle originale. Nous appliquons sur Z_t, l'opérateur : $U_t = |Z_t|$. Le théorème suivant permet alors d'exhiber la dispersion σ associée.

Théorème 1 : Soit Y_t un processus gaussien i.i.d. indicé dans le temps de moyenne $\beta_0+\beta_1 t$ et de variance σ^2, tel que $Y_t = \beta_0+\beta_1 t+ \sigma\varepsilon_t$, où ε_t, est la loi Normale standard, alors $\sigma = \left(\sqrt{\pi}/(2\sqrt{3})\right)\mathbf{E}\left(\left|Y_t - 2Y_{t-1} + Y_{t-2}\right|\right)$. Ce théorème[1] fournit un estimateur de σ tel que : $\hat{\sigma}_U = \left(\sqrt{\pi}/(2\sqrt{3}(T-2))\right)\sum_{t=3}^{T}\left|y_t - 2y_{t-1} + y_{t-2}\right|$. Les résultats obtenus à l'aide de ce théorème est essentiel pour la phase 2 car ils permettent de faire apparaitre dans la série observée u_t, les niveaux de dispersion des segments candidats. En effet, la démarche de segmentation (Derquenne, 2011A) est alors appliquée sur u_t. A la fin du processus, la segmentation sélectionnée offrira un ensemble de segments caractérisés par le modèle (1). Soient $\tau_1^\sigma,...,\tau_s^\sigma,...,\tau_{S_1}^\sigma$, les S_1 segments de dispersion obtenus précédemment sur la série u_t. Alors le segment τ_s^σ fournira une estimation des T_s valeurs u_t, telle que : $\hat{u}_t = \hat{\alpha}_0 + \hat{\alpha}_1 t$ pour $t \in \tau_s^\sigma$

Phase 3 : Seconde segmentation en tenant compte de la dispersion. Cette phase a pour objectif de fournir une segmentation finale de Y_t en tenant compte de la dispersion a priori des données estimée lors de la phase 2. Les $\hat{u}_t$ sont intégrés dans la matrice de dispersion du MLGH (1). En d'autres termes, cela revient à utiliser une matrice diagonale de poids lors de l'estimation des paramètres du modèle à l'aide de l'estimateur REML.

[1] La démonstration de ce théorème est donnée dans (Derquenne, 2011B)

3 Application

Nous avons appliqué la nouvelle méthode proposée sur le même jeu de données simulées que nous avions utilisé dans (Derquenne, 2011A) afin de comparer les résultats. Nous avons choisi 10 segments, selon le modèle (1). Pour chacun des 10 segments, le nombre d'observations, les valeurs des coefficients β_0 et β_1, et la dispersion σ associés sont générés aléatoirement. La figure 1a affiche simultanément la série simulée y_t (en bleu) et la série u_t associée (en rouge) issue de la phase 1. La figure 1b fournit la segmentation des u_t en rouge (phase 2) sur laquelle 12 segments de dispersion ont été détectés. Nous pouvons constater que tous les segments ne sont pas constants, en effet les deuxième et neuvième segments sont croissants. La figure 1d donne la segmentation finale (phase 3) qui comporte 18 segments sur laquelle l'ensemble des ruptures semble être détecté. Il en été de même pour l'ancienne méthode de segmentation dans laquelle 12 segments avait été identifiés (fig. 1c). Visuellement les qualités respectives des deux segmentations sont comparables. Cependant, bien que cela soit peu visible, les segments supplémentaires de la nouvelle approche permettent de mieux découper la variation. Par exemple, le quatrième segment de l'ancienne méthode qui est découpé en deux sous-segments grâce à la nouvelle approche, faisait apparaitre effectivement une variation hétérogène de la série (plus de variabilité dans le premier sous-segment que dans le second). Par ailleurs, nous avons comparé nos résultats à ceux obtenus à l'aide d'algorithmes de programmation dynamique développés par Lavielle (2009). Ces dernières manquent des ruptures (fig. 2a, b, c et d) alors que ce n'est pas le cas pour nos deux méthodes. Nous avons alors évaluer la qualité de reconstitution du signal brut et l'adéquation des segmentations estimées par les six méthodes à la segmentation générée. Par exemple, les MAPE de nos ancienne et nouvelle méthodes sont très proches de celle de la segmentation générée (11,1% et 10,2% *vs* 9,9%), alors que le MAPE varie entre 12,4% et 75,8% pour les quatre méthodes de programmation dynamique. De plus, nous avons calculé le pourcentage d'erreurs inférieures à 10%, sur laquelle notre nouvelle méthode apparait à nouveau la plus performante (87% contre 70% en moyenne pour les quatre méthodes et 86,2% pour notre ancienne méthode). Le nombre de segments identifiés au même endroit que ceux de la segmentation générée montrent également que nos deux approches sont plus performantes sur ce jeu de données que les quatre autres, avec 6 segments retrouvés, contre 0, 2 ou 3. Ces résultats sont complétés par l'erreur d'éloignement des segments estimés par rapport aux segments générés. La méthode proposée obtient un pourcentage d'erreur relativement faible (16,3%) comparé à l'ancienne (24,3%) et surtout par rapport à ceux des quatre autres (35% à 52%).

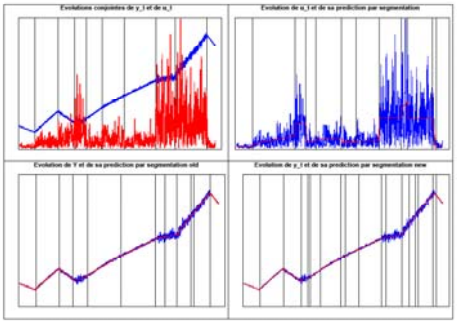


FIG. 1 a,b,c,d : *Nouvelle et ancienne approches*

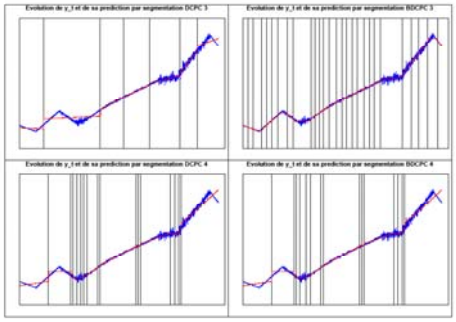


FIG. 2 a,b,c,d : *Programmation dynamique*

4 Apports, critiques, applications et voies futures

La méthode proposée ici, qui permet de segmenter une série temporelle, a pour objectif d'améliorer une démarche que nous avions introduite (Derquenne, 2011A). Celle-ci avait obtenu des résultats encourageants sur des données simulées et sur des données réelles. Elle concurrence fortement les approches fondées sur la programmation dynamique. L'amélioration proposée consiste à exhiber un signal à partir des données brutes représentant leur dispersion grâce à un théorème permettant d'exhiber l'écart-type d'un signal gaussien, puis de réaliser une première segmentation de celui-ci afin de tirer des segments de dispersion, et enfin d'inclure ces derniers comme des poids dans une seconde segmentation lors de la phase de modélisations successives. Sur des exemples simulés, cette nouvelle approche permet à la fois d'améliorer l'ancienne méthode, mais aussi de montrer qu'elle est plus performante que des approches par programmation dynamique. Cette méthode est notamment très intéressante pour des applications dans lesquelles les signaux changent de processus (non stationnarité). En perspective, nous comparerons notre méthode à celle développée dans Arlot et al. (2010) qui utilise une approche par validation croisée. Enfin, le théorème introduit pour un signal gaussien sera généralisé à d'autres lois dans nos futures recherches.

Références

Arlot, S. and Celisse, A. (2010), Segmentation of the mean of heteroscedastic data via cross-validation, *Statistics and Computing*, pp. 1-20.

Derquenne, C. (2011A), An Explanatory Segmentation Method for Time Series, *in Proceedings of Compstat 2010*, Y. Lechevallier and G. Saporta (eds.), 1st Edition, pp. 935-942.

Derquenne, C. (2011B), Segmentation of Time Series with Heteroskedastic Components, *in Proceedings of the 58th Congress of ISI*, Dublin, Ireland.

Guédon, Y. (2008), Exploring the segmentation space for the assessment of multiple change-point models. Institut National de Recherche en Informatique et en Automatique, *Cahier de recherche 6619*.

Lavielle, M. (2009), Detection of Changes using a Penalized Contrast (the DCPC algorithm), http:\\www.math.u-psud.fr\~lavielle\programmes_lavielle.html.

Perron, P. and Kejriwal, M. (2006), Testing for Multiple Structural Changes in Cointegrated Regression Models. Boston University, *C22*.

Summary

The proposed method allows to segment a time series into two phases. The first is to exhibit the variability of the series with an initial segmentation on a proper signal, then consider the structure of dispersion in the second phase which will provide a segmentation using an heteroskedastic linear model.

Dynamic clustering algorithm for geostatistical functional data

Antonio Balzanella*, Elvira Romano*
Rosanna Verde*

*Second University of Naples, Via del Setificio 15, 81100 Caserta
antonio.balzanella@gmail.com, elvira.romano@unina2.it, rosanna.verde@unina2.it

Résumé. Dans cet article nous proposons une stratégie de classification dynamique pour la partition d'un ensemble de données géostatistiques fonctionnelles. L'idée est de découvrir une différent structures de variabilité spatiales entre les classes. Nous utilisons comment représentants des classes des fonctions variogrammes. L'approche proposé est validé sur données réelles.

1 Introduction

Spatial interdependence of phenomena is a common future of many environmental applications such as oceanography, geochemistry, geometallurgy, geography, forestry, environmental control, landscape ecology, soil science, and agriculture. We can think, for instance, of daily patterns of geophysical and environmental phenomena where data (from temperature to sound) are instantaneously recorded over large areas by sensor networks. In these applications, explanatory variables are functions of the time essentially continuous but that are observed and recorded discretely and that have a certain degree of spatial correlation each other.

In the last years the analysis of such data is performed by Spatial Functional Data Analysis (SFDA) (Delicado et al., 2010) which is a branch of Functional data analysis (Ramsay et al., 2005).

In this paper we focus on a clustering strategy based on the Dynamic Clustering Algorithm (DCA) in (Celeux et al., 1988). Differently from other approaches for clustering spatial functional data (Giraldo et al., 2010), the proposed method allows to discover a partition of the curves into clusters and a representation of the spatial variability structure of each cluster.

The Dynamic Clustering Algorithm is a general clustering approach which optimizes a criterion of fitting between the partition of a set of objects and the representative elements of the clusters. According to our objective, the representative element of each cluster is a variogram function for functional data and the clusters are groups of functions similar to each other in terms of spatial functional variability.

2 Variogram based Dynamic Clustering approach for spatially dependent functional data

Spatially dependent functional data may be defined as realizations of a continuous Spatial Functional process $\{\chi_s(t) : t \in T, s \in D \subseteq R^d\}$, where s is a generic data location in the $d-$dimensional Euclidean space.

We assume to observe a sample of curves $(\chi_{s_1}(t), \ldots, \chi_{s_i}(t), \ldots, \chi_{s_n}(t))$ for $t \in T$ where s_i is a generic data location in D.

For each t we have a second order stationary and isotropic random process, with mean and variance functions constant and covariance depending only on the distance between sampling sites : $E(\chi_{\mathbf{s}}(t)) = m(t)$, for all $t \in T,\ s \in D$, $V(\chi_{\mathbf{s}}(t)) = \sigma^2(t)$, for all $t \in T,\ s \in D$, and $Cov(\chi_{s_i}(t), \chi_{s_j}(t)) = C(h,t)$ where $h = \|s_i - s_j\|$ and all $s_i, s_j \in D$

This implies that exists a variogram function for functional data $\gamma(h,t)$, also called trace-variogram function (Delicado et al., 2010), such that :

$$\gamma(h,t) = \gamma_{s_i s_j}(t) = \frac{1}{2}V(\chi_{s_i}(t) - \chi_{s_j}(t)) = \frac{1}{2}E\left[(\chi_{s_i}(t) - \chi_{s_j}(t))^2\right] \tag{1}$$

where $h = \|s_i - s_j\|$ and all $s_i, s_j \in D$.

By using Fubini's theorem, the previous becomes $\gamma(h) = \int_T \gamma_{s_i s_j}(t)dt$ for $\|s_i - s_j\| = h$. This variogram function can be estimated by the classical methods of the moments by means of :

$$\hat{\gamma}(h) = \frac{1}{2\,|N(h)|} \sum_{i,j \in N(h)} \int_T \left(\chi_{s_i}(t) - \chi_{s_j}(t)\right)^2 dt \tag{2}$$

where $N(h) = \{(s_i; s_j) : \|s_i - s_j\| = h\}$ for regular spaced data and $|N(h)|$ is the number of distinct elements in $N(h)$.

The empirical variograms cannot be computed at every lag distance h and due to variation in the estimation it is not ensured that it is a valid variogram.

Such as in applied geostatistics the empirical variograms are thus approximated (by ordinary least squares (OLS) or weighted least squares (WLS)) by model functions ensuring validity. Some widely used models are : Spherical, Gaussian, exponential or Mathern.

The variogram, as defined before, is used to describe the spatial variability among functional data across a spatial domain.

In order to describe the spatial variability substructures we introduce the concept of the spatial variability components regard to a specific location, defining the centered variogram function.

Coherently with the definition above, given a curve $\chi_{s_i}(t)$, the centered variogram can be expressed by

$$\gamma^{s_i}(h,t) = \frac{1}{2}E[(\chi_{s_i}(t) - \chi_{s_j}(t))^2] \tag{3}$$

for each $s_j \neq s_i \in D$.

Similarly to the variogram function, the centered variogram of the curve $\chi_{s_i}(t)$, as a function of the lag h, can be estimated through the method of moments :

$$\hat{\gamma}^{s_i}(h) = \frac{1}{2\,|N^{s_i}(h)|} \sum_{j \in N^{s_i}(h)} \int_T \left(\chi_{s_i}(t) - \chi_{s_j}(t)\right)^2 dt \tag{4}$$

where $N^{s_i}(h) \subset N(h) = \{(s_i; s_j) : \|s_i - s_j\| = h\}$ and it is such that $|N(h)| = \sum_i |N^{s_i}(h)|$.

Consistently with the DCA schema, we propose to optimize a fitting criterion between the centered variogram function $\gamma_k^{s_i}(h)$ and a theoretical variogram function $\gamma_k^*(h)$ for each cluster C_k (with k= 1,. . .,K) as follows :

$$\Delta = \sum_{k=1}^{K} \sum_{\chi_{s_i} \in C_k} (\gamma_k^{s_i}(h) - \gamma_k^*(h))^2 \tag{5}$$

where $\gamma_k^{s_i}$ is the centered variogram which describes the spatial dependence between a curve χ_{s_i} at the site s_i and all the other curves χ_{s_j} at different spatial lag h.

In order to optimize the criterion Δ, starting from a random partitioning of the curves, the algorithm alternates a *representation* and an *allocation* step until the convergence to a stationary value of the criterion.

In the *representation* step the theoretical variogram $\gamma_k^*(h)$ of the set of curves $\chi_{s_i} \in C_k$, for each cluster C_k is estimated. This involves the computation of the empirical vatiogram , and its model fitting by the Ordinary Least Square method.

In the *allocation* step, the function $\gamma_k^{s_i}$ is computed for each curve χ_{s_i}. Then a curve χ_{s_i} is allocated to a cluster C_k by evaluating its matching with the spatial variability structure of the clusters.

3 Main results

In order to evaluate the performance of the proposed strategy on real data, we use a dataset provided by the Institute for Mathematics Applied to Geosciences [1].

The dataset reports the average monthly temperatures recorded by approximately 8000 stations located in US, in the period 1895 to 1997.

Our tests have been performed taking the data in the period $1993 - 1997$, thus for each station we have a time series made by at most 60 observations.

In order to run the clustering algorithm we need to select the number of clusters K and the theoretical variogram model to fit the empirical variogram estimated for each cluster. Since we do not have any information on the true number of spatial variability structures, we apply the algorithm for $K = 2, \ldots, 6$ and then we select K according to the maximum decreasing of the value of the optimized criterion Δ. For the tested dataset the best choice is $K = 3$.

The choice of the the theoretical variogram model is performed evaluating the value of the criterion Δ for several well known parametric models : Esponential, Spherical, Gaussian. According to this test, the best model for the dataset is the exponential variogram.

Starting from the chosen input parameters, the algorithm run on the dataset, detects the spatial regions available in Fig. 1. The value of the optimized criterion is $\Delta = 2.9e^{+4}$, the number of iterations until convergence has been 9.

It is possible to note that the three discovered clusters split the studied area into three spatial regions having three different spatial variability structures. The regions include : most of the East end West coasts, North, South.

1. http ://www.image.ucar.edu/Data/US.monthly.met/

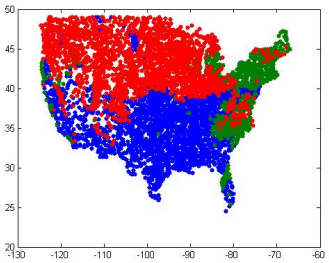


FIG. 1 – *Clusters plotted on the geographic map.*

4 Conclusions

In this paper we have introduced a clustering strategy which partitions the set of spatially located curves into groups which are homogeneous in terms of spatial variability. The spatial variability of each cluster is defined by a variogram function for functional data which represents the prototype of the cluster. The method has been tested on real data in order to evaluate its capability to discover spatial regions characterized by different spatial variability structures.

Références

Celeux, G., E. Diday, G. Govaert, Y. Lechevallier, et H. Ralambondrainy (1988). Classiffication automatique des donnees : Environnement statistique et informatique.

Delicado, P., R. Giraldo, C. Comas, et J. Mateu (2010). Statistics for spatial functional data: some recent contributions. *Environmetrics 21*, 224–239.

Giraldo, R., P. Delicado, et J. Mateu (2010). Hierarchical clustering of spatially correlated functional data. Technical report, http://www.ciencias.unal.edu.co/unciencias/datafile/estadistica/RepInv12.pdf.

Ramsay, J., E., et B. W. Silverman (2005). *Functional Data Analysis (Second ed.)*. Springer Verlag.

Summary

In this paper we propose a dynamic clustering algorithm for partitioning a set of geostatistical functional data. The aim is to detect clusters of curves that are homogeneous in terms of the spatial functional variability. Each cluster is represented by a variogram function for functional data which allows to summarize its spatial variability structure. The effectiveness of the proposed method is evaluated on real data.

Une extension de l'indice de Rand par une mesure de similarité entre matrices de partitions non strictes

Carl Frélicot*, Romain Quéré*

*Univ La Rochelle,
Laboratoire Mathématiques, Image et Applications
{carl.frelicot,romain.quere}@univ-lr.fr,

Résumé. Un nouvel indice de comparaison de deux partitions est proposé, étendant aux partitions non strictes l'indice de Rand. Il repose à la fois sur des outils de la théorie des ensembles flous pour la construction des matrices de coïncidence de chaque partition et sur une nouvelle mesure de similarité entre les colonnes des deux matrices de partition.

1 Introduction

Une partition d'un ensemble $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ de n objets en c groupes peut être représentée par une matrice de partition U de taille $(c \times n)$ dont le terme général u_{ik} indique le degré d'appartenance du $k^{\text{ème}}$ objet au $i^{\text{ème}}$ groupe. Ces matrices sont de nature différente selon que les groupes sont mutuellement exclusifs ou non et selon que les objets appartiennent totalement ou partiellement aux groupes. On distingue en général les partitions :

- *possibilistes* à valeurs dans $\mathcal{M}_{pcn} = \{U \in \mathbb{R}^{c\,n} : u_{ik} \in [0,1]\}$,
- *floues/probabilistes* à valeurs dans $\mathcal{M}_{fcn} = \{U \in \mathcal{M}_{pcn} : \sum_{i=1}^{c} u_{ik} = 1\}$, et
- *strictes* à valeurs dans $\mathcal{M}_{hcn} = \{U \in \mathcal{M}_{fcn} : u_{ik} \in \{0,1\}\}$.

Anderson et al. (2010) y ont ajouté l'ensemble des partitions *douces* $\mathcal{M}_{scn} = \mathcal{M}_{pcn} \backslash \mathcal{M}_{hcn}$. Les algorithmes de partitonnement étant nombreux et leur paramétrage quasi-infini, il est essentiel de pouvoir comparer les partitions qu'ils fournissent à l'aide de mesures de concordance ou *indices de comparaison* $I(U,V)$ qui mesurent l'accord (généralement entre 0 et 1) entre deux partitions U et V (Borgelt, 2005). La littérature récente montre un regain d'intérêt pour la définition de nouveaux indices dédiés aux partitions non strictes, soit par approche directe (Di Nuovo et Catania, 2007), soit par extension des indices dits *stricts* dans le sens où l'on retrouve les indices stricts si U et V sont dans $\mathcal{M}_{hcn}$. On trouve dans cette catégorie les indices reposant sur l'extension de la *matrice de contingence* croisant U et V (Ceccarelli et Maratea, 2008; Anderson et al., 2010) et ceux reposant sur l'aggrégation des deux *matrices de coïncidence* croisant chaque partition avec elle-même (Borgelt, 2005; Quéré et al., 2010). Les deux constructions ne sont en général équivalentes que dans le cas strict. Nous nous intéressons dans cet article aux indices fondés sur les matrices de coïncidence qui présentent les avantages suivants : aucune mise en correspondance des groupes de U et V n'est nécessaire, les nombres de groupes de U et V peuvent être différents, et ils étendent les indices bien connus de Rand, Jaccard, Fowlkes-Mallow (Albatineh et al., 2006). La plupart de ces indices non stricts utilisent des outils de la théorie des ensembles flous (essentiellement des normes triangulaires (Klement

et al., 2000)) qui étendent ceux de la théorie classique des ensembles (Borgelt, 2005; Brouwer, 2009; Quéré et al., 2010). Une alternative possible, introduite dans Hüllermeier et Rifqi (2009) pour le cas des partitions de $\mathcal{M}_{fcn}$, consiste à utiliser une mesure de similarité entre les colonnes de U et V, vues comme des vecteurs d'appartenance non stricte $\mathbf{u}_k$ et $\mathbf{v}_k$ des objets $\mathbf{x}_k$ aux partitions. Nous avons récemment proposé un cadre unifiant la plupart des indices $I(U,V)$ des deux approches dans Quéré et Frélicot (2011) sous la forme d'un triplet (f,g,N) :

$$I(U,V) = \frac{1}{N}\sum_{k=2}^{n}\sum_{l=1}^{k-1} g(f(\mathbf{u}_k,\mathbf{u}_l), f(\mathbf{v}_k,\mathbf{v}_l)) \tag{1}$$

où f est la fonction génératrice des matrices de coïncidence non strictes, g celle de la mesure de concordance et N un facteur de normalisation permettant de retrouver les indices stricts. Ce cadre permet d'une part de montrer les conditions aux bornes et les propriétés d'unité, de symétrie et de maximalité que f et g vérifient ou non selon les indices, d'autre part de bien mettre en évidence le partage du même f, g ou N par certains indices. Enfin, il met en perspective la possibilité de définir de nouveaux indices de comparaison de partitions non strictes hybrides, tirant profit des avantages des deux alternatives, définis par un triplet où f reposerait sur la théorie des ensembles flous et g serait une mesure de similarité entre les termes des matrices de coïncidence non strictes, f et g vérifiant toutes les conditions et propriétés répertoriées. Nous présentons dans cet article une telle proposition pour l'indice de Rand.

2 Un indice de comparaison étendant l'indice de Rand

Pour étendre l'indice de Rand aux partitions de $\mathcal{M}_{fcn}$, il est usuel d'utiliser, comme fonction génératrice du terme général de la matrice de coïncidence d'une partition U, la fonction $f_R^{\top}(\mathbf{u}_k,\mathbf{u}_l) = \sum_{i=1}^{c} \top(u_{ik},u_{il})$ où $\top$ est une t-norme [1]. Il a été montré que pour les partitions de $\mathcal{M}_{scn}$, il faut procéder à une normalisation de sorte que f reste à valeurs dans $[0,1]$ et satisfasse la propriété d'unité : $f(\mathbf{u}_k,\mathbf{u}_k) = 1$. Quéré et al. (2010) ont proposé une solution générique à l'aide de fonctions [2] $K_{\top}$ telles que $\top\big(K_{\top}(a),K_{\top}(a)\big) = a$. Nous suivons cette idée et utilisons pour f, la fonction $\mathbb{R}_+ \times \mathbb{R}_+ \to [0,1]$:

$$f_{K,R}^{\top}(\mathbf{u}_k,\mathbf{u}_l) = \frac{\sum_{i=1}^{c} \top(u_{ik},u_{il})}{\top\left(K_{\top}\left(\sum_{i=1}^{c}\top(u_{ik},u_{ik})\right), K_{\top}\left(\sum_{i=1}^{c}\top(u_{il},u_{il})\right)\right)}. \tag{2}$$

La fonction g que nous proposons dans cet article, dont le but est d'aggréger les termes des matrices de coïncidence de U et V, repose sur :

1. une transformation géométrique par rapport à un point $O' = (o,o)$ de la première bissectrice $\Delta : (x,y) \mapsto (x',y')$, comme montré à la figure 1-(*gauche*)
2. une fonction de poids w à valeurs dans $[0,1]$, paire, telle que $w(0) = 1$, qui fait diminuer $g(x,y)$ lorsque (x',y') s'écarte de Δ ; des exemples classiques sont donnés à la figure 1
3. une fonction-profil p à valeurs dans $]0,1]$ qui modifie $g(x,y)$ selon que (x',y') est proche ou non de O' ; parmi les profils possibles, nous proposons la fonction : $p(x') = t + \rho\, x'^2$, où $t \in]0,\sqrt{2}]$ et $\rho \in \mathbb{R}_+$ sont des paramètres utilisateur [3].

1. exemples de t-normes basiques : le minimum $\top_M(a,b) = \min(a,b)$, le produit $\top_P(a,b) = ab$ et la t-norme de Łukasiewicz $\top_L(a,b) = \max(a+b-1,0)$

2. les fonctions de normalisation de la coïncidence sont données dans l'article cité pour la plupart des t-normes

3. il est facile de voir que O' n'a pas d'influence si $\rho = 0$, de sorte que o est inutile

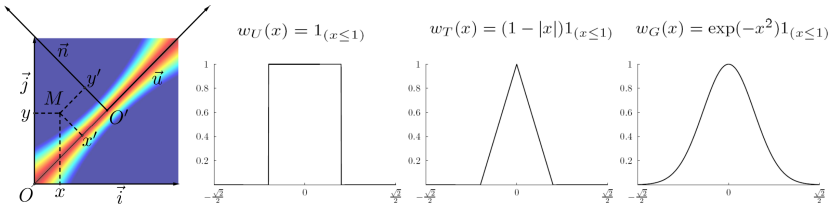
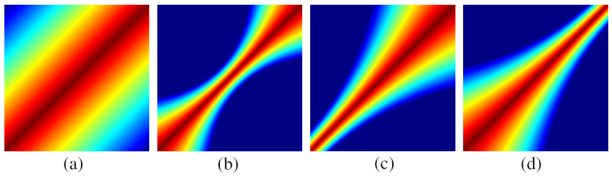


FIG. 1 – *Transformation géométrique et exemples de fonctions* w.

Étant donné l'ensemble de paramètres $Q = \{w, t, c, o\}$, nous proposons de prendre pour g, la fonction $[0,1]^2 \to [0,1]$:

$$g_Q(x, y) = w\left(\frac{y'}{p(x')}\right). \tag{3}$$

Il est aisé de montrer qu'ainsi définie, g_Q satisfait les conditions et propriétés requises détaillées dans (Quéré et Frélicot, 2011), et qu'associée à $f^{\top}_{K,R}$ et au nombre de paires différentes d'objets $N = \frac{n(n-1)}{2}$ dans (1), elle permet de retrouver l'indice de Rand original si les partitions sont strictes, quels que soient les paramètres $Q = \{w, t, \rho, o\}$ de g et la t-norme $\top$ de f. De plus, cet indice fondé sur le triplet $(f^{\top}_{K,R}, g_Q, N)$, que nous appellerons *Normalized Soft Window-based similarity Rand Index* ($NSWRI$), atteint sa valeur maximum (un) lorsque $U \equiv V$, quelle que soit leur nature (strictes, floues ou possibilistes). La figure 2 illustre, par quelques exemples, l'influence des paramètres Q sur la fonction de similarité g_Q.



Q	w	t	c	o
(a)	w_T	$\frac{\sqrt{2}}{2}$	0	–
(b)	w_T	0.1	0.8	0.5
(c)	w_T	0.1	0.3	0
(d)	w_T	0.1	0.3	1

FIG. 2 – *Isosurfaces de* g_Q *sur* $[0,1]^2$ *pour divers ensembles de paramètres* Q.

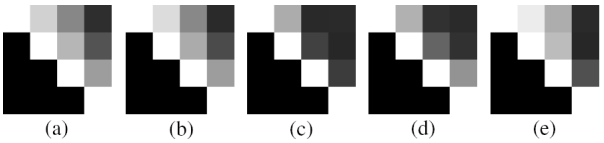
3 Résultats et conclusion

Des résultats sur la comparaison de partitions floues et possibilistes (nombre de groupes différents) fournies par des algorithmes de partitionnement (*Fuzzy/Possibilistic C-Means*) à la partition de référence de nombreux benchmarks seront présentés lors de la conférence [4]. Ils montrent l'intérêt de l'approche par rapport aux indices étendus de la littérature cités ici, en particulier parce qu'elle permet de comparer des partitions de nature différente. Illustrons brièvement [5] la flexibilité que le paramétrage confère à l'indice proposé sur quatre partitions : $U_1 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$, $U_2 = \begin{pmatrix} 0.1 & 0.1 & 0.8 & 0.9 \\ 0.0 & 0.9 & 0.2 & 0.0 \\ 0.9 & 0.0 & 0.0 & 0.1 \end{pmatrix}$, $U_3 = \begin{pmatrix} 0.2 & 0.3 & 0.6 & 0.7 \\ 0.1 & 0.6 & 0.3 & 0.1 \\ 0.7 & 0.1 & 0.1 & 0.2 \end{pmatrix}$ et $U_4 = \begin{pmatrix} 0.33 & 0.33 & 0.34 & 0.34 \\ 0.33 & 0.34 & 0.33 & 0.33 \\ 0.34 & 0.33 & 0.33 & 0.33 \end{pmatrix}$, où $U_1 \in \mathcal{M}_{h34}$, $U_{\alpha+1} \in \mathcal{M}_{f34}$, $\alpha \in \{1, 2, 3\}$ est plus floue que U_α (au sens de l'entropie de partition), et telles que les partitions floues se réduisent à U_1 par binarisation. Intuitivement, ces partitions s'ordonnent par similarité décroissante. La figure 3 montre les dissimilarités de toutes les paires $(U_\alpha, U_{\alpha'})$ obtenues avec les indices $SERI$ (*Soft Equivalence Rand Index*,

4. le format de soumission est trop court
5. la combinaison des différents paramètres étant infinie, l'incidence de $\top$ et w est omise ici

(Hüllermeier et Rifqi, 2009)) et $NSWRI$ pour les jeux de paramètres indiqués. Plus la cellule est claire, plus la valeur de l'indice de Rand étendu est élevée et par conséquent la concordance. On voit clairement, parmi d'autres choses, que :

- $NSWRI$ peut être paramétré afin de se comporter comme $SERI$ – (b) vs (a),
- diminuer t permet de rendre $NSWRI$ plus drastique – (b) vs (c)(d)(e),
- o peut être choisi afin de favoriser – (c) vs (d) – ou défavoriser – (c) vs (e) – plus ou moins la concordance entre les partitions les plus floues, selon la valeur de ρ.



$(\top, Q)$	$\top$	w	t	ρ	o
(b)	$\top_P$	w_T	$\frac{\sqrt{2}}{2}$	0	–
(c)	$\top_P$	w_T	0.3	0	–
(d)	$\top_P$	w_T	0.3	0.3	0
(e)	$\top_P$	w_T	0.3	0.8	1

FIG. 3 – *Dissimilarités entre partitions avec $SERI$ (a) et $NSWRI$ pour divers paramètres.*

Références

Albatineh, A., M. Niewiadomska-Bugaj, et D. Mihalko (2006). On similarity indices and correction for chance agreement. *Journal of Classification 23*, 301–313.

Anderson, D., J. Bezdek, M. Popescu, et J. Keller (2010). Comparing fuzzy, probabilistic, and possibilistic partitions. *IEEE Transactions on Fuzzy Systems 18*(5), 906–918.

Borgelt, C. (2005). Prototype-based classification and clustering. Habilitation Thesis, Fakultat fur Informatik der Otto von Guericke, Universitat Magdeburg.

Brouwer, R. (2009). Extending the rand, adjusted rand and jaccard indices to fuzzy partitions. *Journal of Intelligent Information Systems 32*(3), 213–235.

Ceccarelli, M. et A. Maratea (2008). A fuzzy extension of some classical concordance measures and an efficient algorithm for their computation. In *12th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems*, pp. 755–763. Springer-Verlag.

Di Nuovo, A. G. et V. Catania (2007). On external measures for validation of fuzzy partitions. In *Lecture Notes in Computer Science 4529*, pp. 491–501.

Hüllermeier, E. et M. Rifqi (2009). A fuzzy variant of the rand index for comparing clustering structures. In *13th International Fuzzy Systems Association World Congress*, pp. 1294–1298.

Klement, E., R. Mesiar, et E. Pap (2000). *Triangular Norms*. Kluwer Academic.

Quéré, R., H. L. Capitaine, N. Fraisseix, et C. Frélicot (2010). On normalizing fuzzy coincidence matrices to compare fuzzy and/or possibilistic partitions with the rand index. In *10th. IEEE International Conference on Data Mining*, pp. 977–982.

Quéré, R. et C. Frélicot (2011). A general framework for a class of comparison indices of soft partitions. In *14th Int. Fuzzy Systems Association World Congress (to appear in june)*.

Summary

We propose a new index to compare two partitions which extends the Rand index to soft partitions. It relies on fuzzy set theoretetic tools to construct the coincidence matrix of each partition, and on a new similarity measure between columns of both partition matrices.

Comparaison des partitions par la distance de transfert pour la coloration de graphes

Daniel Cosmin Porumbel*, Jin-Kao Hao**, Pascale Kuntz***

*Univ Lille Nord de France, F-59000 Lille, France, UArtois, LGI2A, F-62400, Béthune, France
daniel.porumbel@univ-artois.fr,
http://www.lgi2a.univ-artois.fr/ porumbel/
** Université d'Angers, LERIA, 2 Bd Lavoisier, 49045 Angers, France
hao@info.univ-angers.fr
http://info.univ-angers.fr/ hao/
** Université de Nantes, LINA, Polytech'Nantes, BP 50609, 44306 Nantes, France
pascale.kuntz@univ-nantes.fr
http://www.univ-nantes.fr/kuntz-cosperec-p/

Résumé. La distance de transfert est utilisée classiquement en classification pour comparer des partitions. Nous montrons ici comment elle peut servir à mesurer la proximité entre des solutions issues de méta-heuristiques pour le problème de coloration de graphes pour améliorer les stratégies de recherche.

1 Introduction

La distance de transfert, étudiée initialement par S. Régnier (Régnier (1965)) est bien connue en classification pour comparer des partitions (Day (1981); Denœud et Guénoche (2006)). Rappelons brièvement, qu'étant données deux partitions P_1 et P_2 d'un ensemble S, la distance de transfert $d(P_1, P_2)$ est définie comme le nombre minimal d'éléments qui doivent être transférés entre les classes de P_1 pour obtenir une partition égale à P_2. En classification, elle est souvent utilisée pour valider un algorithme en mesurant l'écart entre le résultat de l'algorithme et une solution connue, ou pour comparer les résultats de différentes approches. Dans cette communication, nous l'utilisons dans un contexte différent : celui de l'analyse d'espaces de recherche ("fitness landscapes") pour un problème de k-coloration de graphes.

D'une façon générale, il est bien connu en optimisation combinatoire que les performances des méta-heuristiques dépendent du choix de plusieurs paramètres qui sont souvent laborieusement déterminés de façon ad hoc. De plus, ces paramètres sont la plupart du temps réglés initialement au début de l'algorithme alors que le comportement de ce dernier est étroitement lié aux propriétés de l'espace de recherche en cours d'exploration. Une meilleure compréhension des comportements des méta-heuristiques et des structures des espaces de recherche associés reste nécessaire pour rendre leurs stratégies "mieux informées".

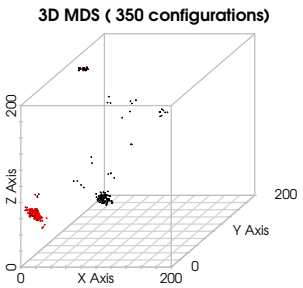
Dans cet objectif, l'intérêt de l'apprentissage et de la fouille de données en optimisation combinatoire est en plein essor comme l'attestent outre des publications (e.g. Boyan et al.

(2000); Battiti et al. (2008)) la récente conférence LION (Learning and Intelligent Optimization). Ces approches d'apprentissage en optimisation peuvent être classées en deux catégories : les méthodes "hors ligne" ou "en ligne". Les approches "hors ligne" s'appuient sur une analyse préalable des espaces de recherche dont les résultats sont exploités dans la phase d'optimisation. Et, les approches "en ligne" recueillent de l'information tout au long du processus d'optimisation afin de prendre des décisions en temps réel telles que notamment un réglage réactif des paramètres.

Dans cette communication, nous nous intéressons à ces deux approches. Nous montrons en particulier comment une analyse de l'espace de recherche basée sur la comparaison de solutions via la distance de transfert nous a permis d'améliorer une recherche Tabou et un algorithme évolutionnaire pour la k-coloration de graphes.

2 Classification des solutions

Afin de mieux cerner la structure d'un espace de recherche associée à une méta-heuristique -ici une méthode Tabou-, nous avons calculé la distance entre les meilleures solutions candidates trouvées (k colorations ayant le plus petit nombre d'arêtes avec les deux extrémités de la même couleur). A titre illustratif, la figure ci-contre montre le résultat d'une approche de type *Multidimensional Scaling* pour une instance du problème de coloration avec 250 sommets. Ici, 350 solutions ont été retenues. On observe que les minima locaux se regroupent en quelques classes qui correspondent à des bassins d'attractions dans l'espace de recherche.



Ce type de configuration a été retrouvé dans une série de tests expérimentaux à grande échelle. En particulier, en considérant les meilleures solutions candidates (ici 40000) visitées par une recherche locale en quelques heures, l'histogramme des distances indique une distribution bimodale qui confirme la présence de classes (avec des valeurs petites -resp. grandes- correspondant aux distances intra-classes -resp. inter-classes). Et, nous avons pu même en déduire une valeur approximative du rayon des classes : $0.1 \times |V|$, où $|V|$ est le cardinal de l'ensemble des sommets du graphe. Nous avons pu ainsi établir une hypothèse de classification selon laquelle les meilleures solutions potentielles eu égard à l'heuristique utilisée sont classifiables dans des classes sphériques dont le rayon peut être estimé expérimentalement assez précisément : la sphère $S(C)$ d'une coloration C est l'ensemble des solutions situées à une distance inférieure à $0.1 \times |C|$.

3 Application : amélioration de méta-heuristiques

3.1 Principe

En se basant sur l'hypothèse de classification des meilleurs solutions (optima locaux) dans l'espace de recherche, nous avons proposé une amélioration de la recherche Tabou, appelée TS–Div (Tabu Search Diversified). Elle introduit une fonction de mémorisation à "gros grain" : le processus mémorise un ensemble restreint de sphères couvrant la trajectoire de la recherche

dans l'espace de recherche. Cette mémorisation permet à la recherche de ne pas re-visiter inutilement un même sous-espace, et de mieux gérer ainsi les sorties de bassins d'attraction d'optima locaux. Plus précisément, TS–Div est basé sur un processus de recherche Tabou auquel est ajouté un processus d'apprentissage avec deux objectifs : *(i)* enregistrer toutes les sphères $S(C_1), S(C_2), ...$ visitées (où $C_i \notin S(C_j)$), et *(ii)* si la coloration à l'itération courante fait partie d'une sphere $S(C_j)$ déjà visitée, déclencher une phase de diversification qui a pour objectif de dévier la recherche hors $S(C_j)$ (Porumbel et al. (2010)).

Nous avons également récemment intégré des informations du même ordre dans un algorithme évolutionnaire de coloration de graphes. Ce travail a été d'ailleurs généralisé pour proposer un algorithme évolutionnaire à base de distances applicable à une classe plus large de problèmes d'optimisation (Porumbel et al. (2011b)).

3.2 Mise en oeuvre : calcul de la distance de transfert

La mise en oeuvre opérationnelle de ces approches nécessite des calculs intensifs de la distance de transfert entre des colorations de graphes qui sont des partitions de l'ensemble des sommets. Or, la méthodologie de calcul communément utilisée repose sur une modélisation du calcul par un problème d'affectation linéaire (Day (1981)) dont la résolution se fait par une méthode hongroise (Kühn (1955)) de complexité en $O\left(k^3\right)$ où k est le nombre de classes (couleurs dans notre cas). Or, cette complexité est trop élevée pour notre contexte applicatif qui nécessite des millions de calculs. Afin de surmonter ces limites, nous avons récemment proposé un nouvel algorithme de calcul de la distance de transfert en complexité linéaire en nombre de sommets dans des cas bien adaptés à l'analyse de l'espace de recherche du problème de k-coloration. Nous présentons ci-dessous un exemple de propriété que nous avons démontré.

Théorème (Porumbel et al. (2011a)). Soient P_1 et P_2 deux partitions en k classes d'un ensemble S. Si pour tout $i \in \{1, 2, ...k\}$, il existe $j \in \{1, 2, ...k\}$ tel que $\left|P_1^i \cap P_2^j\right| \geq \frac{\left|P_1^i \cup P_2^j\right|}{2}$, alors la distance de transferts peut être calculée en $O(|S|)$ étapes.

En pratique, si ce type de condition de proximité entre partitions est vérifiée, alors le calcul est en complexité linéaire ; sinon, on exécute un algorithme hongrois classique. Pour vérifier expérimentalement que cette condition est effectivement bien souvent vérifiée dans notre contexte, nous avons calculé un million de distances entre des colorations trouvées par l'heuristique ci-dessus sur deux graphes (1000 sommets avec respectivement $k = 20$ et $k = 86$) extraits de la base DIMACS qui sert de référence pour la coloration. Nous avons observé que la condition de proximité requise par la propriété ci-dessus est vérifiée dans $99,99\%$ des cas.

3.3 Résultats et perspectives

D'un point de vue expérimental, nous avons comparé TS–Div avec d'autres heuristiques locales de la littérature sur 12 instances variées de la base DIMACS connues pour être difficiles et où le k optimal n'est pas toujours connu. Nous avons montré que dans tous les cas, TS-Div est plus efficace qu'un algorithme Tabou classique, et qu'en moyenne il obtenait des résultats compétitifs avec les meilleurs algorithmes connus de la littérature. Nous avons même trouvé une meilleure borne -depuis 1991- pour une des instances. Pour l'algorithme évolutionnaire, nous avons pu à nouveau trouver de nouvelles bornes inférieures ; ainsi, nous avons ensuite gé-

néralisé notre algorithme à d'autres problèmes d'optimisation combinatoires (Porumbel et al. (2011b)).

Si ces résultats confirment que notre approche est prometteuse, différentes pistes complémentaires sont à explorer. Pour la distance de transfert, une analyse fine des distributions avait montré que les valeurs n'étaient véritablement discriminantes que dans un intervalle restreint (Denœud et Guénoche (2006)) ; nous devons ainsi approfondir notre première analyse des distributions évoquées dans la Section 2. De plus, nous nous sommes focalisés ici sur la distance de transfert mais d'autres distances de comparaison de partitions pourraient être utilisées ; la seule contrainte d'application étant leur complexité de calcul. Enfin, d'un point de vue méthodologique en optimisation, notre approche pourrait être étendue à d'autres problèmes d'optimisation pour lesquels il est possible de calculer efficacement une dissimilarité entre solutions.

Références

Battiti, R., R. Brunato, et F. Mascia (2008). *Reactive Search and Intelligent Optimization*. Springer.

Boyan, J., W. Buntine, et A. Jagota (2000). Statistical machine learning for large-scale optimization. *Neural Computing Surveys 3*(1), 1–58.

Day, W. (1981). The complexity of computing metric distances between partitions. *Mathematical Social Sciences 1*, 269–287.

Denœud, L. et A. Guénoche (2006). Comparison of distance indices between partitions. In V. Batagelj et al. (Eds.), *Data Science and Classification*, pp. 21–28. Berlin, Germany : Springer.

Kühn, H. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 83–97.

Porumbel, C., J. Hao, et P. Kuntz (2010). A search space "cartography" for guiding graph coloring heuristics. *Computers & Operations Research 37*, 769–778.

Porumbel, D. C., J.-K. Hao, et P. Kuntz (2011a). An efficient algorithm for computing the distance between close partitions. *Discrete Applied Mathematics 159*(1), 53–59.

Porumbel, D. C., J.-K. Hao, et P. Kuntz (2011b). Spacing memetic algorithms. In *Genetic and Evolutionary Computation Conference (GA Track)*, pp. 1061–1068. ACM.

Régnier, S. (1983 et 1965). Sur quelques aspects mathématiques des problèmes de classification automatique. *Mathématiques et Sciences Humaines 82*, 20. (reprint of *ICC Bulletin*, 4, 175- 191, Rome, 1965).

Summary

The transfer distance is typically used in classification to compare partitions. This paper shows how it can be used to measure the proximity between the best candidate solutions generated by graph coloring meta-heuristics. The objective is to discover certain structures in the spatial distribution of these best candidate solutions, so as to improve the search algorithms.

Sur le degré d'éparpillement d'un sous-ensemble dans une partition

Jean Diatta

Université de La Réunion, EA2525-LIM, Saint-Denis de la Réunion, F-97490, France
jean.diatta@univ-reunion.fr,
http://personnel.univ-reunion.fr/jdiatta

Résumé. Nous proposons un indice permettant de mesurer le degré auquel un sous-ensemble d'un ensemble donné est éparpillé dans une partition de cet ensemble. Nous donnons alors une condition nécessaire pour que pour un sous-ensemble et une partition donnés, il existe une classe de la partition, qui contienne au moins une proportion donnée des éléments du sous-ensemble. De plus, nous montrons comment le degré d'éparpillement peut être utilisé pour construire des indices de comparaison de partitions.

1 Introduction

Le nombre élevé d'indices de comparaison de partitions, introduits dans littérature, témoigne, s'il en est besoin, de l'intérêt et de l'importance de comparer des partitions. A titre indicatif, nous mentionnons, par exemple, le populaire indice de Rand (Rand, 1971), ses versions ajustée proposée par Hubert et Arabie (1971) et asymétrique proposée par Chavent et al. (2001), l'indice de Mirkin (Mirkin, 1996) qui est une autre version corrigée de l'indice de Rand, l'indice de Jaccard (Jaccard, 1908), l'indice de Fowlkes-Mallows (Fowlkes et Mallows, 1983), l'indice de Meilă-Heckerman (Meilă et Heckerman, 2001), la distance de transferts de Régnier (Charon et al., 2006).

Dans cette note, nous proposons un indice permettant de mesurer le degré auquel un sous-ensemble d'un ensemble donné est éparpillé dans une partition de cet ensemble. Nous donnons alors une condition nécessaire pour que pour un sous-ensemble et une partition donnés, il existe une classe de la partition, qui contienne au moins une proportion donnée des éléments du sous-ensemble. De plus, nous montrons comment le degré d'éparpillement peut être utilisé pour construire des indices de comparaison de partitions.

Dans tout ce qui suit, E désigne un ensemble fini ayant au moins 2 éléments. Par ailleurs, les sous-ensembles de E que nous considérons sont tous non vides.

2 Degré d'emboîtement de deux sous-ensembles

Soient S_1, S_2 deux sous-ensembles non vides de E. Nous définissons le degré auquel S_1 est contenu dans S_2 par :

$$\eta(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1|},$$

où $|X|$ désigne le nombre d'éléments de X. L'indice η est clairement non symétrique. Plus précisément, $\eta(S_1, S_2)$ est la proportion d'éléments de S_1 qui appartiennent à S_2. Il atteint son minimum, à savoir 0, lorsque S_1 et S_2 sont disjoints et, son maximum, à savoir 1, est atteint lorsque S_1 est contenu dans S_2. De même, le degré auquel S_1 est en dehors de S_2 peut être défini par :

$$\delta(S_1, S_2) = \frac{|S_1 \setminus S_2|}{|S_1|}.$$

Le minimum de $\delta(S_1, S_2)$, à savoir 0, est atteint lorsque S_1 est contenu dans S_2, alors que son maximum, à savoir 1, est atteint lorsque S_1 et S_2 sont disjoints ou, de manière équivalente, lorsqu'aucun élément de S_1 n'appartient à S_2.

3 Degré d'éparpillement d'un sous-ensemble dans une partition

Une *partition* de E est une collection $\mathcal{C} = \{C_1, \ldots, C_K\}$ de sous-ensembles non vides de E tels que :

- pour $i \neq j$, $C_i \cap C_j = \varnothing$;
- $\cup_{k=1}^{K} C_k = E$.

Soient $\mathcal{C}$ une partition de E et $S \subseteq E$ un sous-ensemble de E. Supposons que S a au moins 2 éléments. Alors chaque élément de S appartient à exactement une classe de $\mathcal{C}$. Ainsi, nous définissons le degré $\zeta(S, \mathcal{C})$ auquel S est éparpillé dans $\mathcal{C}$ par :

$$\zeta(S, \mathcal{C}) = \frac{|\{X \in \mathcal{C} : S \cap X \neq \varnothing\}| - 1}{|S| - 1}.$$

En d'autres termes, $\zeta(S, \mathcal{C})$ est le nombre, normalisé, de classes de $\mathcal{C}$ qui recoupent S. L'intuition de la formule de normalisation est fondée sur le fait que chaque sous-ensemble non vide S recoupe au moins une (et au plus $|S|$) classe(s) d'une partition $\mathcal{C}$. On notera que $\zeta(S, \mathcal{C})$ est minimum (*i.e.* égal à 0) lorsque tous les éléments de S appartiennent à une seule et même classe de $\mathcal{C}$. Cela correspond au cas où S n'est pas éparpillé dans $\mathcal{C}$. Par ailleurs, $\zeta(S, \mathcal{C})$ est maximum (*i.e.* égal à 1) lorsque chaque classe de $\mathcal{C}$ contient au plus un élément de S. Cela correspond au cas où S est totalement éparpillé dans $\mathcal{C}$.

Exemple 1 *Désignons par $\mathcal{C}_D(E)$ la partition discrète de E, consistant en l'ensemble des singletons de E, et par $\mathcal{C}_G(E)$ la partition grossière $\{E\}$ de E.*

1. *Pour tout sous-ensemble S de E, $\zeta(S, \mathcal{C}_D(E)) = 1$ et $\zeta(S, \mathcal{C}_G(E)) = 0$.*
2. *Soit $E = \{1, 2, \ldots, 7\}$, $\mathcal{C} = \{\{1\}, \{2, 3\}, \{4, 5, 6, 7\}\}$, et soient $S_1 = \{1, 2\}$, $S_2 = \{1, 2, 3\}$ et $S_3 = \{1, 2, 3, 4\}$. Alors*
   (a) $\zeta(S_1, \mathcal{C}) = 1$,
   (b) $\zeta(S_2, \mathcal{C}) = \frac{1}{2}$,
   (c) $\zeta(S_3, \mathcal{C}) = \frac{2}{3}$.

Le résultat suivant donne une condition nécessaire pour que pour un sous-ensemble et une partition donnés, il existe une classe de la partition, qui contienne au moins une proportion donnée des éléments du sous-ensemble.

Proposition 1 *Soient $\mathcal{C}$ une partition de E et S un sous-ensemble de E ayant au moins deux éléments. Soient q et r deux entiers tels que $1 \leq q \leq r$. Alors $\zeta(S, \mathcal{C}) \leq \frac{|S|(r-q)}{r(|S|-1)}$ dès lors qu'il existe une classe de $\mathcal{C}$ contenant au moins la proportion $\frac{q}{r}$ des éléments de S.*

En conséquence, le résultat suivant donne une condition nécessaire pour que pour un sous-ensemble et une partition donnés, il existe une classe de la partition, qui contienne plus de la moitié des éléments du sous-ensemble.

Proposition 2 *Soient $\mathcal{C}$ une partition de E et S un sous-ensemble de E ayant au moins deux éléments. Alors $\zeta(S, \mathcal{C}) \leq \frac{1}{2}$ s'il existe une classe de $\mathcal{C}$ contenant plus de la moitié des éléments de S.*

La condition de la proposition 1 n'est pas suffisante, sauf lorsque $|S| \leq \frac{r}{q}$. En effet, on vérifie le résultat suivant sans difficulté.

Proposition 3 *Soient $\mathcal{C}$ une partition de E et S un sous-ensemble de E ayant au moins deux éléments. Soient q et r deux entiers tels que $1 \leq q \leq r$. Alors il existe une classe de $\mathcal{C}$ contenant au moins la proportion $\frac{q}{r}$ des éléments de S si $\zeta(S, \mathcal{C}) \leq \frac{r-q}{q(|S|-1)}$.*

4 Comparaison de deux partitions

Dans cette section, nous montrons comment le degré d'éparpillement peut être utilisé pour comparer deux partitions. Soit à cet effet deux partitions $\mathcal{C}$ et $\mathcal{C}'$ avec $\mathcal{C} = \{C_k\}_{1 \leq k \leq K}$. Désignons par $S_{\mathcal{C},\mathcal{C}'} = (\zeta(C_k, \mathcal{C}'))_{1 \leq k \leq K}$ le vecteur dont le $k$ème terme est le degré d'éparpillement de C_k dans $\mathcal{C}'$. Le vecteur $S_{\mathcal{C}',\mathcal{C}}$ serait défini de manière similaire.

Un indice pour comparer $\mathcal{C}$ et $\mathcal{C}'$ peut avoir une expression de la forme $g(f(S_{\mathcal{C},\mathcal{C}'}), f(S_{\mathcal{C}',\mathcal{C}}))$, où f et g sont des fonctions. Des exemples de valeur de $f(S_{\mathcal{C},\mathcal{C}'})$ pourraient être :

– une transformation de Minskowski pondérée

$$(\sum_{k=1}^{K} w_k(\zeta(C_k, \mathcal{C}'))^p)^{\frac{1}{p}},$$

où $p > 0$ est un réel strictement positif et les w_k des poids ;

– la proportion de termes de $S_{\mathcal{C}',\mathcal{C}}$ situés en dessous ou au dessus d'un certains seuil.

Des exemples de valeur de $g(f(S_{\mathcal{C},\mathcal{C}'}), f(S_{\mathcal{C}',\mathcal{C}}))$ pourraient être :

– une combinaison convexe $\beta f(S_{\mathcal{C},\mathcal{C}'}) + \gamma f(S_{\mathcal{C}',\mathcal{C}})$;

– la moyenne géométrique $\sqrt{f(S_{\mathcal{C},\mathcal{C}'})\, f(S_{\mathcal{C}',\mathcal{C}})}$.

Ces indices pourront faire l'objet d'une étude plus détaillée et être comparés aux indices connus.

Soient $\mathcal{C}$ et $\mathcal{C}'$ deux partitions de E et $\alpha \in [0, 1[$. Nous considérons la proportion $\nu_\alpha(\mathcal{C}, \mathcal{C}')$ des classes de $\mathcal{C}$, de cardinalité au moins 2, éparpillées dans $\mathcal{C}'$ à un degré inférieur ou égal à α. Formellement :

$$\nu_\alpha(\mathcal{C}, \mathcal{C}') = \begin{cases} 1 \text{ si } \mathcal{C}_{\geq 2} = \varnothing \\ \frac{|\{C \in \mathcal{C}_{\geq 2} : \zeta(C, \mathcal{C}') \leq \alpha\}|}{|\mathcal{C}_{\geq 2}|} \text{ sinon} \end{cases}$$

où, pour une partition $\mathcal{C}$, $\mathcal{C}_{\geq 2}$ désigne l'ensemble des classes de $\mathcal{C}$ ayant au moins 2 éléments. Pour α proche de 0, plus $\nu_\alpha(\mathcal{C}, \mathcal{C}')$ est proche de 1, plus $\mathcal{C}$ est en accord avec $\mathcal{C}'$. En fait, $\nu_0(\mathcal{C}, \mathcal{C}') = 1$ si et seulement si $\mathcal{C}$ est plus fine que $\mathcal{C}'$, *i.e.*, toute classe de $\mathcal{C}$ est contenue dans une classe de $\mathcal{C}'$. Ainsi, $\nu_0(\mathcal{C}, \mathcal{C}')$ peut être interprété comme le degré de finesse de $\mathcal{C}$ par rapport à $\mathcal{C}'$.

Références

Charon, I., L. Denoeud, A. Guénoche, et O. Hudry (2006). The maximum transfer distance between partitions. *Journal of Classification 23*, 103–121.

Chavent, M., C. Lacomblez, et B. Patouille (2001). Critère de Rand asymétrique. In *8èmes Rencontres de la Société Francophone de Classification*, Pointe à Pitre, France, pp. 82–88.

Fowlkes, C. et C. Mallows (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association 78*(383), 553–569.

Hubert, L. et P. Arabie (1971). Comparing partitions. *Journal of Classification 2*, 193–198.

Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. *Bulletin de la Société Vaudoise des Sciences Naturelles 44*, 223–270.

Meilă, M. et D. Heckerman (2001). An experimental comparison of model-based clustering methods. *Machine Learning 42*, 9–29.

Mirkin, B. (1996). *Mathematical Classification and Clustering*. Dordrecht : Kluwer Academic Press.

Rand, W. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66*, 846–850.

Summary

We define an index capturing the degree to which a subset of a given set is scattered in a partition of this set. Then, we provide a necessary condition for a given proprortion of elements of a subset to be contained in some cluster of a partition. Moreover, we show how the scattering degree can be used to design indices for comparing partitions.

Analyse de la stabilité d'une partition par décomposition de l'indice de Rand

Lassad El Moubarki*, Ghazi Bel Mufti**, Patrice Bertrand***

*Time Université, Tunis, Tunisie - elmoubarki.lassad@yahoo.fr,
**ESSECT, Université de Tunis, Tunisie - belmufti@yahoo.com
***Université Paris-Dauphine, Ceremade, Paris - bertrand@ceremade.dauphine.fr

Résumé. Pour valider un partitionnement, une approche empirique fréquemment appliquée consiste à évaluer la stabilité des classes obtenues. Partant du principe qu'un manque de stabilité est dû à un défaut de cohésion et/ou d'isolation des classes, nous proposons d'interpréter l'indice de Rand par les degrés d'isolation et de cohésion de la partition. Nous illustrons notre approche sur des jeux de données réels et simulés, et nous la comparons à d'autres méthodes de validation pour la détermination du bon nombre de classes.

1 Introduction

La stabilité est une propriété naturelle et nécessaire pour valider une classification. Pour évaluer la stabilité de leurs résultats, les utilisateurs de méthodes de classification ont souvent recours au rééchantillonnage des données. Cependant, Ben-David et von Luxburg (2008) ont montré l'existence de cas où la meilleure valeur de stabilité, mesurée après rééchantillonnage, n'est pas suffisante pour garantir la validité d'une partition et du nombre de classes associé. Il est ainsi préférable non seulement de vérifier la stabilité d'une partition après rééchantillonnage des données, mais aussi de calculer différentes mesures de validité (cf. Guénoche et Grandcolas (2002)). Dans ce qui suit, nous proposons de décomposer l'indice de Rand qui est une mesure de stabilité fréquemment utilisée. Plus précisément, nous montrons que la valeur de l'indice de Rand est une moyenne pondérée de contributions mesurant différents critères de classification, *i. e.* isolation, cohésion et stabilité (globale) de chaque classe de la partition examinée. Cette décomposition qui, dans son principe général, est analogue à la décomposition introduite par Bertrand et Bel Mufti (2006) à l'aide de l'indice de Loevinger, permet d'interpréter les valeurs de l'indice de Rand avec plus de fiabilité. Dans la dernière section, nous évaluons notre approche de validation par des expérimentations sur des jeux de données réels et simulés.

2 Décomposition de l'indice de Rand

Considérons un ensemble X de n objets à classer. Si P est une partition de X et S un sous-ensemble de X, nous notons P_S la restriction de P à S. De plus, pour tous $x, y \in X$, notons $P(x, y) = 1$ si x et y sont classés ensemble selon P, et $P(x, y) = 0$ sinon. Rappelons que si P et Q sont deux partitions de X, alors il est usuel de mesurer l'écart entre P et Q à l'aide de

l'indice de Rand, noté $\mathcal{R}(P,Q)$ qui, avec nos notations, s'écrit :

$$\mathcal{R}(P,Q) = \frac{1}{n\,(n-1)} \sum_{x,y\in X,\, x\neq y} \big[P(x,y)Q(x,y) + (1-P(x,y))(1-Q(x,y))\big].$$

Notons $P = \mathcal{A}(X)$ la partition de X obtenue par une méthode de partionnement arbitraire, notée $\mathcal{A}$. Pour évaluer la stabilité de P, on mesure les écarts entre P et chaque partition $\mathcal{A}(S_j)$, où $S_1, \ldots, S_N$ sont des ensembles X perturbés : pour simplifier notre présentation, les ensembles $S_1, \ldots, S_N$, sont des échantillons i.i.d. de X qui, de plus, sont stratifiés selon la partition P (cf. Bertrand et Bel Mufti (2006) pour plus de détails). Une estimation de la stabilité de la partition P est alors donnée par la moyenne $\mathrm{Ra}(P) = \frac{1}{N}\sum_{j=1}^{N} \mathcal{R}(P_{S_j}, \mathcal{A}(S_j))$, appelée par la suite *indice de Rand de* P. Nous proposons d'évaluer la contribution de chacune des classes $C \in P$ à la mesure de stabilité $\mathrm{Ra}(P)$ de P, selon les deux critères suivants :

Critère d'isolation : *Si* $x \in C$ *et si* $P(x,y) = 0$*, alors* $\mathcal{A}(S_j)(x,y) = 0$.

Critère de cohésion : *Si* $x \in C$ *et si* $P(x,y) = 1$*, alors* $\mathcal{A}(S_j)(x,y) = 1$.

Le critère d'isolation s'interprète de la façon suivante : "Si l'on considère deux objets, un seul appartenant à C, alors les deux objets ne sont pas classés ensemble par la partition $\mathcal{A}(S)$ obtenue sur les données perturbées". Le critère de cohésion s'interprète de manière similaire. Chaque critère se présente donc sous la forme d'une règle de la forme $\mathcal{C}_1 \rightarrow \mathcal{C}_2$, où $\mathcal{C}_1$ et $\mathcal{C}_2$ sont des conditions portant sur les couples d'objets de S_j. On peut donc mesurer le degré de satisfaction de chaque critère à l'aide de l'indice de confiance $\mathbb{P}(\mathcal{C}_1|\mathcal{C}_2)$. On estime ainsi le degré d'isolation de la classe C à l'aide de l'indice $R_i(C;S)$ défini par :

$$R_i(C;S) = \frac{1}{n_{CS}\,(n' - n_{CS})} \sum_{x\in C\cap S,\, y\in S\setminus C} \big[1 - \mathcal{A}(S)(x,y)\big], \text{ où } n_{CS} = |C\cap S| \text{ et } n' = |S|.$$

De même, le degré de cohésion de C est estimé par l'indice $R_c(C;S)$ défini par :

$$R_c(C;S) = \frac{1}{n_{CS}\,(n_{CS}-1)} \sum_{x,y\in C\cap S,\, x\neq y} \mathcal{A}(S)(x,y).$$ Pour une estimation fiable, on utilise la moyenne de ces indicateurs sur un grand nombre, noté N, d'échantillons $S_1, \ldots, S_N$ de X. Nous choisissons N, à l'aide du théorème Central Limite, en fonction de la précision souhaitée sur la valeur de l'indice : en général $N = 100$ est suffisant. Ainsi les estimations des degrés d'isolation et de cohésion, notées $\mathrm{Ra}_i(C)$ et $\mathrm{Ra}_c(C)$, sont respectivement :

$$\mathrm{Ra}_i(C) = \frac{1}{N}\sum_{j=1}^{N} R_i(C;S_j) \quad \text{et} \quad \mathrm{Ra}_c(C) = \frac{1}{N}\sum_{j=1}^{N} R_c(C;S_j).$$

En définissant *l'indice de Rand de la classe* C, noté $\mathrm{Ra}(C)$, comme étant le pourcentage de paires $\{x,y\}$ vérifiant à la fois les critères d'isolation et de cohésion, on montre que :

$$\mathrm{Ra}(C) = \frac{n_{CS}-1}{n'-1}\,\mathrm{Ra}_c(C) + \frac{n'-n_{CS}}{n'-1}\,\mathrm{Ra}_i(C) \quad \text{et} \quad \mathrm{Ra}(P) = \sum_{C\in P} \frac{n_{CS}}{n'}\,\mathrm{Ra}(C).$$

Ces deux dernières relations indiquent une double décomposition de l'indice $\mathrm{Ra}(P)$: selon les classes, puis selon l'isolation et la cohésion de chaque classe. Remarquons que l'on peut, de façon similaire, décomposer $\mathrm{Ra}(P)$, additivement, selon deux indices globaux l'un portant sur l'isolation de P et l'autre sur la cohésion de P.

3 Expérimentations

Considérons d'abord le jeu de données artificiel, présenté dans la figure 1, qui est structuré en 5 classes "naturelles" non convexes de taille 50 chacune. La méthode de partitionnement *pam* fournit 5 classes (cf. FIG 1). Les mesures de stabilité (cf. TAB 1), définies en section 2, montrent que la classe centrale, notée C_4, manque à la fois de cohésion (0.603) et d'isolation (0.909), ces valeurs étant les plus faibles du tableau. L'indice de Rand de C_4 est faible aussi (0.883). L'indice d'isolation de C_3 (0.971) montre une faible isolation et l'indice de cohésion de C_1 (0.892) confirme le manque de cohésion de cette classe.

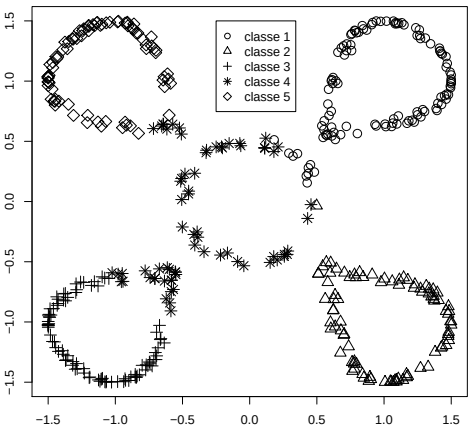


FIG. 1 – *Jeu de données artificiel*

	C_1	C_2	C_3	C_4	C_5	Partition
Ra_c	0.892	0.990	0.922	0.603	0.937	0.891
Ra_i	0.982	0.985	0.971	0.909	0.987	0.969
Ra	0.969	0.986	0.966	0.883	0.982	0.953

TAB. 1 – *Mesures de stabilité des 5 classes.*

Par ailleurs, nous avons testé notre approche sur 7 bases de données réelles issues du UCI Machine Learning Repository et nous l'avons comparée aux 8 méthodes de validation suivantes : CH (Calinski et Harabasz, 1974), KL (Krzanowski et Lai, 1985), Gap statistique (Tibshirani et al., 2001), Silhouette (Rousseeuw, 1987), $\mathrm{In}-\mathrm{Group}\,\mathrm{Proportion}$ (Kapp et Tibshirani, 2007), Jump (Sugar et James, 2003), $\mathrm{Prediction}\,\mathrm{strength}$ (Tibshirani et Walther, 2005) et Clest (Dudoit et Fridlyand, 2002). Pour déterminer la meilleure partition d'un ensemble de données selon notre approche nous avons procédé de la manière suivante :

a) nous avons identifié l'ensemble des partitions valides en sélectionnant celles telles que le minimum des indices d'isolation et de cohésion de leurs classes est supérieur à 0.95. Si aucune partition ne vérifie cette condition, nous avons considéré qu'il y a absence de structure (*i. e.* le meilleure nombre de classe est égal à 1) ;

b) parmi les partitions de cet ensemble, nous avons choisi celle ayant le plus grand nombre de classes.

Les résultats montrent que notre approche est avec la méthode Silhouette, la meilleure méthode : elle identifie le bon nombre de classes de 4 bases de données parmi les 7 étudiées. Par ailleurs, et même si la base de données *Glass* compte réellement 6 classes, la partition en 2 classes retenue par notre méthode ainsi que par les méthodes CH et $\mathrm{Gap}\,\mathrm{Statistique}$ a un sens dans la mesure où cette base de données est constituée de 2 types de verres : ceux qui sont utilisés pour les fenêtres et ceux qui ne le sont pas.

Nous avons aussi comparé notre approche aux 8 méthodes de validation citées plus haut sur 1000 jeux de données simulées selon 10 modèles dans lesquels nous avons tenu compte de

plusieurs facteurs (*i. e.* nombre de classes générées, taille et forme des classes, méthode de classification utilisée, dimension des données). Les résultats montrent que notre approche, avec un pourcentage de succès global supérieur à 0.8, a été la méthode la plus performante (cf. El Moubarki (2009) pour une présentation des résultats plus détaillée).

Références

Ben-David, S. et U. von Luxburg (2008). Relating clustering stability to properties of cluster boundaries. In R. Servedio et T. Zhang (Eds.), *Proceedings of the 21st Annual Conference on Learning Theory*, Berlin, pp. 379–390. Springer.

Bertrand, P. et G. Bel Mufti (2006). Loevinger's measures of rule quality for assessing cluster stability. *Computational Statistics & Data Analysis 50*, 992–1015.

Caliǹski, R. B. et J. Harabasz (1974). A dendrite method for cluster analysis. *Communications in Statistics 3*, 1–27.

Dudoit, S. et J. Fridlyand (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biol 3*(7).

El Moubarki, L. (2009). *Décomposition et évaluation des mesures de stabilité d'un partitionnement*. Thèse de doctorat, Université Paris Dauphine.

Guénoche, A. et S. Grandcolas (2002). Representation and evaluation of partitions. In *Classification, clustering, and data analysis, IFCS 2002*, Stud. Classification Data Anal. Knowledge Organ., pp. 131–138. Springer, Berlin.

Kapp, A. V. et R. Tibshirani (2007). Using the in-group proportion to estimate the number of clusters in a dataset. *preprint submitted to Anals of Applied Statistics*.

Krzanowski, W. J. et Y. T. Lai (1985). A criterion for determing the number of groups in data set using sum of squares clustering. *Biometrics 44*, 23–34.

Rousseeuw, P. J. (1987). Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*, 53–65.

Sugar, C. A. et G. M. James (2003). Finding the number of clusters in a data set : An information theoretic approach. *Journal of the American Statistical Association 98*(463), 750–763.

Tibshirani, R. et G. Walther (2005). Cluster validation by prediction strength. *Journal of Computational & Graphical Statistics 14*(3), 511–528.

Tibshirani, R., G. Walther, et T. Hastie (2001). Estimating the number of clusters in a dataset via the gap statistic. *Journal of Royal Statistical Society 32*(2), 411–423.

Summary

Several stability measures have been proposed with the aim to assess partitions obtained from clustering algorithms. We consider the well known Rand index as a measure of stability, and observe that this index can be expressed as a weighted mean of two indices that estimate, respectively, the isolation and the cohesion of the clusters. We compare our approach with the most successful methods for predicting the number of clusters reported in recent surveys.

Classification de données concernant l'Érika

Marc Le Pouliquen* Marc Csernel**

*Telecom Bretagne, Labsticc UMR 3192, BP 832, 29285 Brest Cedex - France
marc.lepouliquen@telecom-bretagne.eu,
**Inria-Rocqencourt, BP-105- 78180 Le Chesnay - France
marc.csernel@inria.fr

Résumé. Cet article étudie la possibilité d'obtenir une classifications des données issues de la marée noire de l'Érika en mettant en valeur l'impact de celle-ci. Un intérêt particulier est porté au classement des espèces collectées selon leur abondance dans les zones sinistrées ou non.

1 Introduction

En décembre 1999, se déroulait dans le Finistère le naufrage du pétrolier Érika provoquant une marée noire qui a touché le littoral français du Sud Finistère à la Vendée. Avant cet accident, il n'y a précédemment aucune étude de référence concernant le milieu marin de ces zones qui permettrait de faire le bilan de l'impact de la catastrophe. Afin de tenter d'évaluer malgré tout les effets de la marée noire, pendant trois ans de 2000 à 2002, un inventaire des espèces a été effectué sur 10 sites (dont 5 impactés par la marée noire cf. la figure 1). Les premiers résultats obtenus par Jacques Grall (cf. Chauvaud et al. (2004)) ne permettent pas de conclure à un impact fort de la marée noire sur l'ensemble des peuplements étudiés.

Il nous a paru intéressant de reprendre ces données et de procéder à différentes classifications afin de vérifier les résultats statistiques. Nous avons ensuite entrepris d'ordonner les variables explicatives afin de visualiser les espèces susceptibles d'avoir été affectées par la catastrophe.

2 Présentation des données

Les données récupérées par Jacques Grall se présentent sous la forme d'une base de données que l'on peut transformer en une simple table individu-variable (cf. Tab 1) où les individus sont les 10 sites étudiés sur les trois années c'est-à-dire 30 sites-annuels et les variables correspondent aux espèces collectées (environ 400). Il faut savoir qu'un certain nombre de paramètres influent de façon importante sur le peuplement des estrans [1] :

- les différents types de substrat (Roches, sédiments,...) influent énormément sur la composition de la faune rencontrée ;
- les échantillonnages des espèces sont pratiqués dans diverses zones de l'estran du fait de sa stratification en ceintures algales (cf. Le Hir (2002)) ;

1. L'estran est la partie du littoral située entre les limites hautes et basses des marées

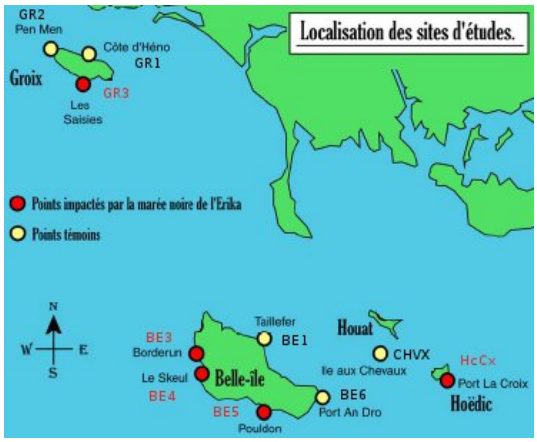


FIG. 1 – *Localisation géographique des 10 sites étudiés avec leur code*

– l'influence du temps d'immersion provoque lui aussi un étagement vertical de la faune ;
– la diversité entre les sites est aussi un facteur sensible, non seulement la diversité inter-îles, mais aussi la diversité intra-îles (Ex : la différence de l'habitat entre Belle-Île et les îles plus proches du continent) ;
– la variabilité du peuplement entre les différentes années est importante.

Enfin, les espèces n'étant pas représentées de façon égale (les études distinguent souvent la richesse spécifique, l'abondance, les espèces rares et les espèces uniques[2]) , nous avons « normalisé » leur quantité afin que les distances ne se focalisent pas uniquement sur les espèces les plus nombreuses.

	BE1-0	BE1-1	BE1-2	BE3-0	BE3-1	BE3-2	···
Modiolus adriaticus	0	0	0	0	0	0	
Triphora adversa	2	0	2	0	0	0	
Abra alba	0	0	0	0	2	0	
Melanella alba	0	0	0	0	0	0	
Ophiura albida	255	64	69	0	1	1	
Jaera albifrons	0	0	0	0	0	0	
Natica alderi	0	0	0	0	0	2	
Sabellaria alveolata	0	0	0	0	0	0	
Syllis amica	14	0	39	3	1	2	
Balanus amphitrite	0	0	0	0	0	0	
⋮							

TAB. 1 – *Extrait de la table* ***site-annuel*** *croisée par les* ***espèces inventoriées***

2. Voici quelques définitions :
– Richesse spécifiques : nombre total d'espèces.
– Abondances : nombres d'individus.
– Espèces rares : ce sont des espèces qui sont trouvées dans un seul échantillon.
– Espèces uniques : ce sont des espèces qui sont trouvées dans moins de 5 échantillons.

3 Ensemble des classifications réalisées

L'Analyse Factorielle des Correspondances de Benzécri (1973) est l'une des méthodes d'analyse factorielle les plus utilisées lorsque l'on dispose d'un tableau de données quantitatives dans lequel les observations (ici, les sites-annuels) sont décrits par un assez grand nombre de variables (ici, les espèces). L'AFC (cf. Fig 2) fait ressortir sur son premier axe (18% de l'inertie) la différence entre l'habitat de Belle-Île et celui des îles plus proches du continent. L'axe deux (16% d'inertie) discrimine quant à lui plutôt les années (il oppose l'année 2000 aux années 2001 et 2002). On peut constater que les sites impactés (en rouge) et non impactés (en vert) sont répartis uniformément sur le plan. L'analyse des autres axes représentant également chacun plus de 10% de l'inertie ainsi que celles réalisées sur des données réduites (espèces rares, espèces nombreuses,...) ne font pas non plus apparaître de discrimination entre les sites impactés et les autres.

La classification hiérarchique de la Figure 3 est réalisée sur les données complètes normalisées avec la distance de Ward (1963). Elle semble opposer les années 2000 aux autres ainsi que les sites ayant un grand nombre d'espèces rares aux autres. Les nombreuses C.A.H. obtenues ne permettent pas de discriminer les sites touchés par la marée noire des autres.

Pour terminer, nous avons utilisé l'algorithme des K-means de MacQueen (1967). Cet algorithme a été exécuté sur les données complètes en construisant deux classes avec la distance euclidienne. Sur la Figure 4, les deux classes obtenues ne discriminent pas les zones impactées. On n'obtient pas non plus de séparation entre les sites impactés en utilisant plus de classes ou des données réduites.

Finalement, l'impact de la marée noire ne semble pas suffisamment important pour que les différentes classifications obtenues le mettent en valeur. Les phénomènes listés dans le paragraphe précédent, comme la diversité entre les sites, ou la variabilité temporelle, semblent suffisamment prégnants pour apparaître dans les classifications et masquer l'impact de la marée noire. Il semble néanmoins intéressant de supposer que cet impact existe, c'est-à-dire de considérer une classification pour laquelle les données se divisent en deux groupes, celles concernant les sites impactés et celles concernant les sites non-impactés. Nous pouvons alors tenter d'orienter les variables (ici, les espèces) pour voir celles qui semblent le plus touchées par la catastrophe.

4 Classement des espèces

Pour réaliser le classement des espèces dont les effectifs semblent avoir été les plus affectés par la marée noire, nous avons établi une méthode intuitive sachant que le nombre de sites considérés est trop petit pour pouvoir utiliser l'arsenal des indices statistiques classiques. La méthode proposée consiste à comparer les effectifs des 15 sites-annuels impactés avec les 15 non-impactés. On obtient alors 225 différences (effectifs des sites-annuels non-impactés - effectifs des sites-annuels impactés) dont le signe permet de savoir pour chaque espèce si l'on a une diminution ou une augmentation de l'effectif. On peut alors supposer que les espèces dont le nombre de signes positifs est nettement plus important que celui des négatifs ont subi les effets toxiques de la catastrophe. Á l'opposé, on trouve probablement des groupes d'espèces opportunistes et tolérantes qui profitent du manque de prédateurs ou d'autres effets induits. Sur le Tableau 2, les 5 premières espèces listées ont plus de 86% des effectifs des sites-annuels

qui augmentent dans les zones impactées, probablement des espèces résistantes à la présence d'hydrocarbures qui se substituent à la faune normale. Au contraire, les 4 dernières espèces listées ont plus de 78% des effectifs des sites-annuels qui diminuent dans les zones impactées, celles-ci semblant être des espèces victimes de la catastrophe.

Espèce	Nb Signe +	Nb Signe -	Nb Zéro	diff Nb Signe
Lysidice ninetta	6	200	20	-194
Hyale nilssoni	23	202	1	-179
Hiatella arctica	38	184	4	-146
Idotea granulosa	38	183	5	-145
Patella ullysiponensis	28	173	25	-145
Nucella lapillus	41	181	4	-140
⋮				
Trivia arctica	27	27	172	0
⋮				
Aonides oxycephala	144	65	17	79
Amphitrite gracilis	133	31	62	102
Aora gracilis	153	42	31	111
Gibbula pennanti	166	52	8	114
Scolelepis fuliginosa	175	23	28	152

TAB. 2 – *Extrait du classement des espèces selon l'importance de la variation entre les zones impactées ou non-impactées*

5 Conclusion

Les classifications obtenues ne permettent pas de visualiser les perturbations provoquées par la catastrophe de l'Érika à partir du tableau des espèces collectées sur l'Estran. La méthode utilisée pour classer les espèces selon l'impact qu'elles ont subies semble intéressante et permet d'obtenir une liste d'espèces à étudier plus en détail.

Références

Benzécri, J.-P. (1973). *L'analyse des données (2 tomes)*. Dunod, Paris.

Chauvaud, S., J. Grall, et G. Gélinaud (2004). Mise en place d'un réseau d'observation des habitats insulaires marins du morbihan. Rapport, Direction Régionale de l'Environnement.

Le Hir, M. (2002). *Les champs de blocs intertidaux à la pointe de Bretagne : Biodiversité, structure et dynamique de la macrofaune*. Ph. D. thesis, Thèse de l'Université de Bretagne Occidentale, Brest, 263pp.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281Ű297.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association 58(301)*, 236–244.

6 Annexe, Figures réalisées avec le logiciel R

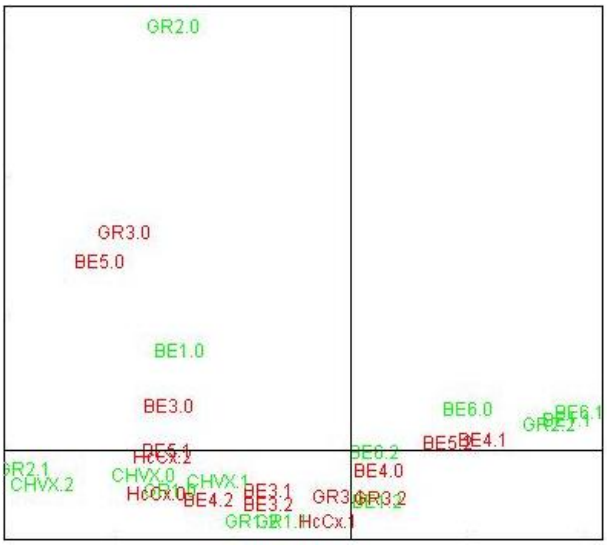


FIGURE 2 – *AFC réalisée sur les données complètes normalisées*

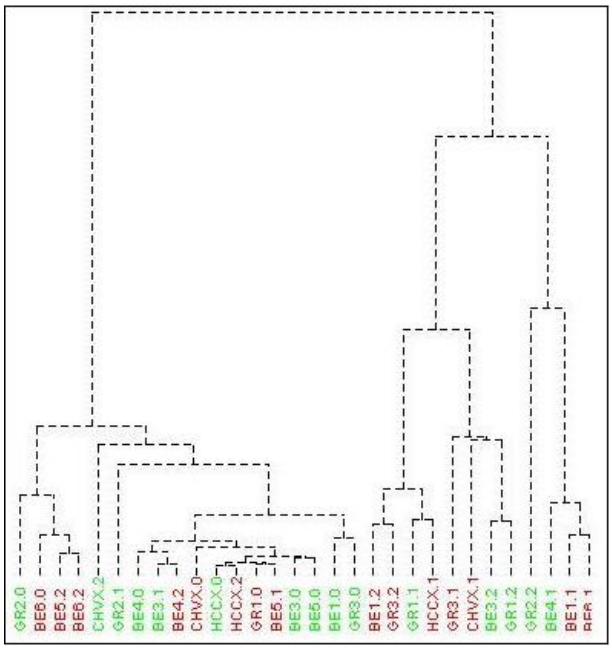


FIGURE 3 – *CAH sur les données complètes normalisées avec la distance de Ward*

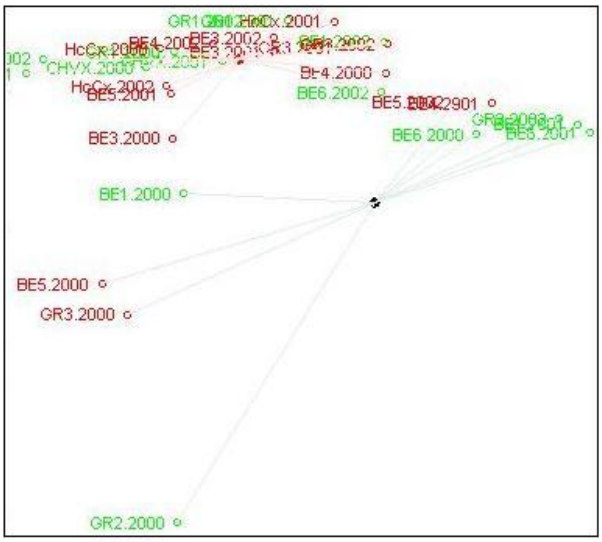


FIGURE 4 – *K-means sur données complètes normalisées avec la distance euclienne*

Summary

This article studies the clustering of the data obtained from the Érika oil spill. A particular interest is carried to the species for which the abundance greatly varies according to the pollution degree of the collecting area.

Typologie des usages d'une plate-forme de Travail Collaboratif Assisté par Ordinateur (TCAO) dans le cadre de la formation d'enseignants

H. Ralambondrainy *, J. Simon *

* LIM, Université de la Réunion, {ralambon,jsimon}@univ-reunion.fr

Résumé. Nous confrontons une typologie des usages d'une plate-forme de TCAO proposée par un expert à celle obtenue par les méthodes d'Analyse des Données.

1. Introduction. Depuis 2005, les professeurs stagiaires (PE2) de l'IUFM de La Réunion utilisent une plate-forme de travail collaboratif assisté par ordinateur (TCAO) pour se former. Stagiaires (et formateurs) y créent et partagent des dossiers où ils déposent et puisent diverses ressources qui doivent les aider à faire classe. Comprendre ce qui se passe sur la plate-forme doit permettre d'améliorer leur formation et pour cela nous analysons les traces qu'ils ont laissées sur celle-ci. Parmi ces traces, on distingue celles d'objets (dossier, document, utilisateur,...) et d'événements relatifs à ces objets (création, modification, lecture,...). Nous avons relié chaque trace à l'individu qui l'a laissée et au groupe auquel il appartenait par le biais des Dossiers Partagés de Plus Haut Niveau (dpphn) (Simon (2009)). Un dpphn regroupe donc toutes les traces laissées par un groupe d'utilisateurs. En 2006-2007, 1050 utilisateurs ont généré 509819 traces. Parmi ces utilisateurs, les 277 PE2 ont partagé 77 dpphn avec une vingtaine de formateurs. Dans (Simon (2009)), nous avons proposé une première catégorisation de ces dossiers (figure 1). Celle-ci s'est faite de manière empirique au travers d'analyses successives des données recueillies :1ère étape volume des échanges, 2ème étape nombre de producteurs, 3ème étape type de producteur (formateur ou stagiaire), 4ème étape analyse des titres des dossiers. Le côté empirique de cette approche nous pousse maintenant à développer une démarche plus formelle par l'utilisation de méthodes multidimensionnelles de classification. La stratégie suivie est celle d'une Analyse Factorielle suivie d'une Classification Ascendante Hiérarchique de Ward sur les dix premières coordonnées factorielles. Une partition est ensuite déterminée par coupure optimale de la hiérarchie, elle est consolidée par la méthode de classification "k-means" (logiciel SPAD).

2. Classification sur les données quantitatives. Pour décrire chaque dossier, nous avions 14 variables : nombre de membres, de stagiaires (PE2), de formateurs, total de producteurs, de producteurs formateurs, de producteurs PE2, de lecteurs formateurs, de lecteurs PE2, total de documents, de documents produits par les formateurs, de documents produits par les PE2, total de lectures, de lectures faites par les formateurs, de lectures faites par les PE2. Comme on l'a dit, le nombre de dossiers était de 77. Une première analyse quantitative des données a fait apparaître une partition à 6 classes. Sur les 6 classes, 5 sont pertinentes du point de vue TCAO. De plus, on repère immédiatement un certain recouvrement entre certaines classes de la partition et certaines catégories de la figure 1. Mais cette partition est aussi composée de deux autres classes que notre catégorisation n'avait pas fait apparaître et qui pourtant sont tout

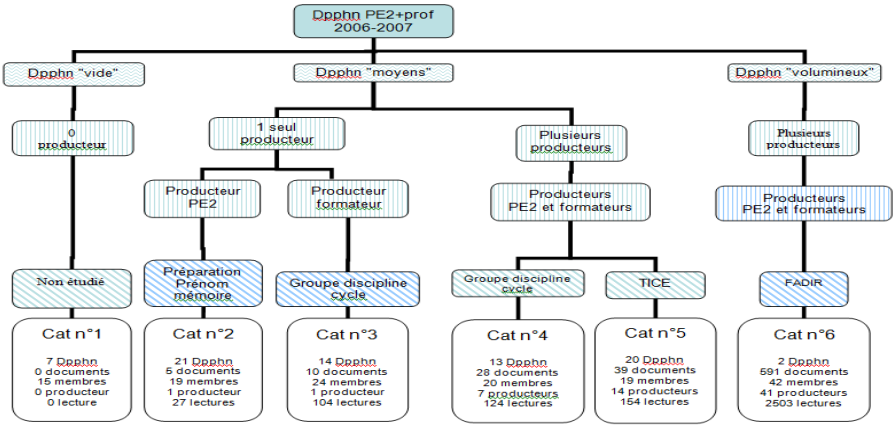


FIG. 1 – *Catégorisation des usages d'une plate-forme de TCAO dans le cadre de la formation*

à fait justifiées du point de vue de la formation car elles correspondent à un travail sur plusieurs groupes de la promotion. Leur objectif était soit de la diffusion de l'information par un seul formateur, soit une mutualisation des ressources. Cependant, ces 5 classes ne représentaient que 28 dossiers, la classe 1 de 49 éléments, trop disparate, non interprétable, a été l'objet d'une nouvelle classification. Cette deuxième analyse quantitative sur les 49 dossiers a permis de faire émerger de nouveaux profils et de mettre l'accent sur d'autres caractéristiques : un seul formateur producteur, un seul formateur lecteur, un nombre important de documents déposés par formateur, ou des dossiers très peu actifs. Cette première étude a ainsi permis de faire émerger les valeurs saillantes pour chaque variable. Certaines avaient déjà été repérées par l'analyse empirique mais d'autres non, par exemple, la taille des groupes. Elle a permis des codages des variables qualitatives dont l'analyse a mis en évidence des liaisons non linéaires. Pour chacune des 14 variables, nous avons défini les modalités en nous basant sur les critères qui étaient apparus grâce aux classifications continues précédentes. Chaque variable s'est vue attribuer entre 3 et 5 modalités. Pour toutes les variables, la modalité 1 était le nombre d'éléments minimum possible. Puis, pour les modalités suivantes, nous nous sommes basés sur les moyennes proposées dans les classifications précédentes en prenant comme intervalle cette moyenne plus ou moins l'écart type indiqué.

3. Classification sur les données qualitatives. La classification a été refaite sur ces nouvelles données qualitatives. La partition obtenue comporte 7 classes, dans ce qui suit, nous les comparons aux 6 catégories de la figure 1. La **classe 1** (7 éléments) est celle des **"dossiers vides"** qui se caractérise par une valeur nulle sur toutes les variables hors membres. Cette classe recouvre exactement la catégorie 1 de la figure 1. Elle correspond à des erreurs ou des essais : des dossiers qui sont créés mais non utilisés. La **classe 7** (2 éléments) **"Accompagnement en stage"** recouvre exactement la catégorie 6 de la figure 1. L'objectif des dossiers est l'accompagnement des PE2 lorsqu'ils sont en stage et qu'ils ont une classe en charge. Ces dossiers indiquent une très grande activité : plus de 1800 lectures, plus de 200 documents déposés. Une

autre des caractéristiques est que le nombre de formateurs impliqués est de 8 ou 9 alors qu'il ne dépasse pas 2 dans les autres dossiers. Enfin ce sont les seuls dossiers où tout le monde produit et tout le monde lit. Les classes 2,3,4,5 et 6 recouvrent donc exactement les catégories 2,3, 4 et 5. La **classe 2** (11 éléments) **"Accompagnement individualisé"** se caractérise par le fait qu'il n'y a que deux membres par dossier, un formateur et un stagiaire, et un seul producteur, le PE2. Ces dossiers correspondent à un travail demandé par le formateur au stagiaire ou une demande d'aide du stagiaire vers le formateur. Ceci est confirmé par le titre des dossiers qui indiquent que ce sont soit des mémoires soit des séances ou des séquences de classe. Cette classe est totalement incluse dans la catégorie 2 mais ne la recouvre pas. C'est en fait une sous catégorie qui indique un travail d'accompagnement personnalisé d'un PE2 par un formateur. A l'opposé, la **classe 5** (4 éléments) **"Diffusion à plusieurs groupes"** contient les dossiers qui concernent une partie de la promotion de PE2. Les dossiers se caractérisent par un nombre élevé de membres et de PE2. Pour 3 dossiers, il n'y a qu'un seul producteur. Il s'agit donc de dossiers destinés à diffuser de l'information à plusieurs groupes de la promotion. Ces trois dossiers relèvent des catégories 2 ou 3 de la figure 1 *"un seul producteur"* mais sont cependant totalement différents en termes d'objectifs de ceux de la classe 2. La distinction entre classe 2 et 5 est donc pertinente (*"accompagnement d'un seul"* **vs** *"diffusion à plusieurs groupes"*). Cependant, le quatrième dossier de la classe 4 est un individu atypique vu le nombre très grand de producteurs PE2 (72). La **classe 4** (22 éléments) **"Diffusion à un groupe"** comprend en moyenne 1 producteur (seuls 4 dossiers ont deux producteurs). La plupart du temps ce producteur est un formateur (16 éléments). Cette classe recouvre donc une bonne partie de la catégorie 3 et une partie de la catégorie 2 (figure 1). Les dossiers de la classe 4 servent à diffuser de l'information aux autres stagiaires du groupe. Cependant le nombre moyen de documents étant de 5, le nombre moyen de PE2 étant de 18 et le nombre de moyen de lectures étant de 18, on peut s'interroger sur l'efficacité de cette diffusion. La **classe 6** (19 éléments) **"Coopération forte"** est composée essentiellement de dossiers TICE (15 sur 19), elle recouvre donc une partie des caractéristiques de la catégorie 5 (15 dossiers sur 20). Ces dossiers doivent permettre de valider le c2i2e du PE2 ce qui implique un certain nombre d'échanges et notamment une certaine homogénéité dans le nombre de lectures par les formateurs, dans le nombre de producteurs PE2 et dans le nombre documents que ceux-ci déposent. Il s'agit d'une classe où il y a réellement production et lecture et donc coopération forte mais de façon cependant nettement moins intensive que dans la classe 7. **La classe 3** (12 éléments) **"Coopération faible"** est plus disparate. Elle se caractérise uniquement par un nombre homogène et assez faible de documents déposés dans les dossiers. Comme elle comporte majoritairement plusieurs producteurs (10 dossiers sur 12), les dossiers ne relèvent pas de la diffusion mais si on parle de coopération il convient de préciser qu'il s'agit de coopération faible. La figure 2 présente la projection des classes dans le premier plan factoriel, les modalités actives n'ont pas été représentées pour des raisons de clarté.

4. En conclusion, notre démarche a consisté à utiliser des méthodes d'analyse de données dans un premier temps pour repérer des caractéristiques qui ont pu échapper à une analyse davantage empirique et dans un deuxième temps pour utiliser ces caractéristiques afin de définir des variables qualitatives et obtenir une classification moins dispersée. Cela a permis de mettre en évidence des profils intéressants comme la classe 2 "Accompagnement individualisé" et la classe 5 "diffusion à plusieurs groupes" absentes de la catégorisation de la figure 1. Cela a aussi montré que certains dossiers qui auraient dû être en "coopération forte" se retrouvent en

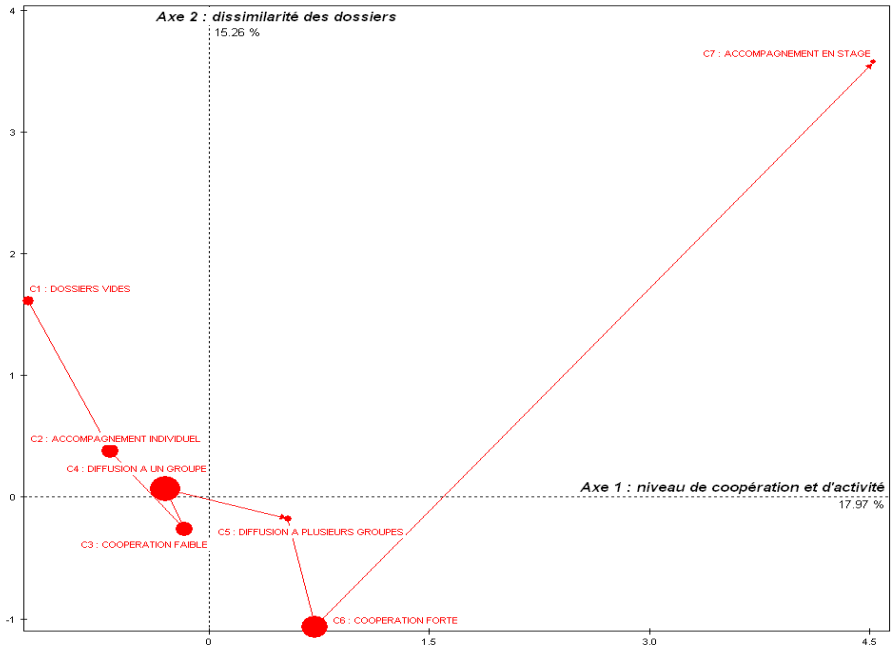


FIG. 2 – *Représentation des classes de la partition dans le premier plan factoriel*

"coopération faible"et ne remplissent donc pas le contrat attendu. Ainsi, sans s'appuyer sur l'étude des titres, les méthodes d'analyse de données ont permis d'obtenir une classification aussi pertinente du point de vue de la formation des stagiaires que l'analyse empirique.

Références

Simon, J. (2009). Three years of use of a cscw platform by the preservice teachers and the trainers of the reunion island teacher training school. In *ICALT 09, Proceedings of the 2009 Ninth IEEE International Conference on Advanced Learning Technologies*, Riga, pp. 637–641.

Summary

We compare a typology of uses of a CSCW platform proposed by an expert with one obtained by Data Analysis methods.

Approche interactive pour la classification non supervisée

Lydia Boudjeloud, Kamel Chelghoum

Laboratoire d'Informatique Théorique et Appliquée (EA 3097)
Ile du Saulcy, 57045 Metz Cedex 1
lydia.boudjeloud|kamel.chelghoum@univ-metz.fr

Résumé. Nous nous intéressons dans cet article à la classification non supervisée interactive. Nous proposons une approche permettant la recherche d'un résultat d'une classification non supervisée de façon interactive, sur des données où la perception visuelle est, bien souvent, plus efficace que les outils classiques de classification non supervisée. L'approche proposée, développée sous `R`, a été testée et comparée sur plusieurs jeux de données artificiels (données qui se chevauchent, les anneaux, ...).

1 Introduction

Dans le processus d'extraction de connaissances à partir de données, il y a au moins deux moyens de faire collaborer les méthodes automatiques avec des méthodes visuelles interactives. Il est possible d'utiliser les méthodes de visualisation en prétraitement de l'algorithme automatique ou en post-traitement de ce même algorithme. En prétraitement de données, on s'aperçoit que, bien souvent, une intuition initiale des concepts cachés peut être acquise de façon visuelle dans les très grandes quantités d'information. On peut trouver des tendances ou corrélations dans ces données grâce à la visualisation des données initiales. Cette étape peut également guider l'utilisateur dans le choix des algorithmes de fouille les plus pertinents ou de leurs paramètres. En post-traitement des connaissances, les méthodes de visualisation sont plutôt utilisées pour interpréter et évaluer des résultats en se basant sur des représentations graphiques plus accessibles que des colonnes de chiffres ou un ensemble de règles. Ces différentes interactions illustrent l'intérêt de faire coopérer des méthodes automatiques et des méthodes visuelles interactives. La compréhension des résultats est ainsi accrue et la précision des algorithmes automatiques peut être facilement améliorée. Une des possibilités pour augmenter la part de la visualisation dans les algorithmes de fouille de données est de remplacer l'algorithme automatique de fouille de données par un algorithme visuel interactif. On parle alors de fouille visuelle de données qui se distingue de la visualisation d'information et consiste donc, en l'utilisation de la visualisation comme outil pour la fouille. L'utilisateur d'un tel système peut être le spécialiste des données (pas forcément un expert en fouille ou analyse de données). Les utilisateurs peuvent, par exemple, construire de façon interactive des arbres de décision en effectuant des coupes successives sur les projections de données (Poulet (2004)). L'approche proposée dans cet article s'inscrit dans cette direction, nous nous basons sur les projections des données en petites dimensions pour sélectionner des groupes d'individus homogènes. L'ap-

proche est efficace sur des ensembles de données (anneaux, demi cercles, ...) où la perception visuelle est, bien souvent, plus efficace que les méthodes classiques.

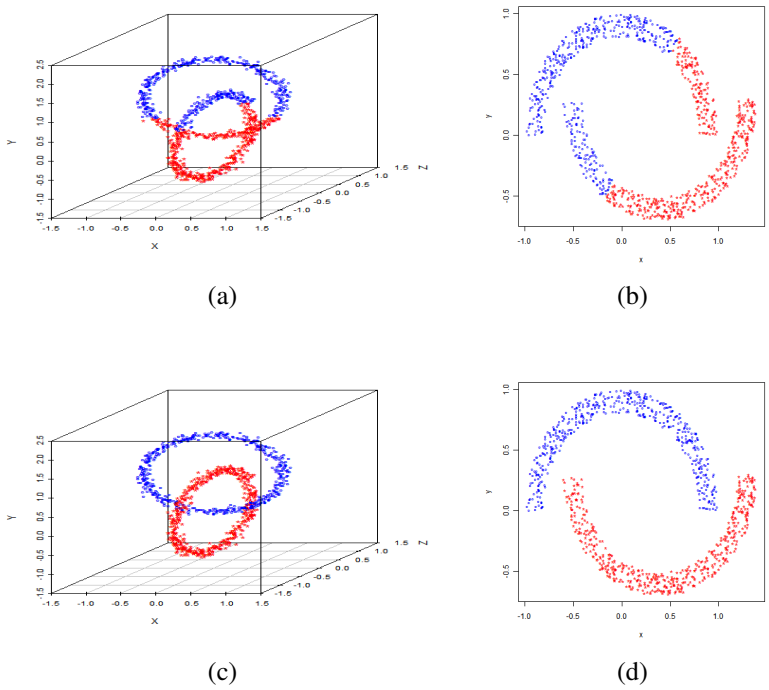
2 Approche interactive

Par son caractère perceptif extrêmement puissant, la vision de l'analyste, ou du spécialiste des données, est un composant essentiel dans le processus d'identification des groupes. Il s'agit dans notre cas de construire une classification non supervisée de façon interactive. L'objectif est d'utiliser la perception visuelle pour identifier les groupes homogènes pouvant former un cluster à partir de plusieurs visualisations. Ces visualisations représentent des projections des données sur un espace réduit (2D ou 3D), le spécialiste des données, ou l'utilisateur de l'outil, peut ainsi sélectionner, ou désélectionner, avec la souris les points individuellement, un à un, ou par petits groupes, à partir d'une ou plusieurs visualisations. L'identification d'un groupe (cluster) peut nécessiter plusieurs opérations de sélection (ou désélection) sur une ou plusieurs projections (visualisations). Les différents groupes de points sélectionnés sont ensuite combinés par des opérations logiques (xor, nand, ...) ou ensemblistes (union, intersection, différence, ...) pour former un ensemble de données représentant le cluster. L'outil permet également, des manipulations sur ces groupes de points sélectionnés, par exemple, changement de cluster pour une donnée en particulier, identification des données atypiques et étiquetage des clusters. Dans le cas de données multidimensionnelles (de dimension supérieure ou égale à 3), la classification non supervisée interactive nécessite une étape préalable de réduction de dimension (de type ACP (aPearson (1901))), de sélection de dimensions (Boudjeloud et Poulet (2005)) ou de projections 2D multiple (Scatterplot (Carr et al. (1987))), les différents groupes de points sélectionnés sur les projections sont ensuite combinés, de la même façon par des opérations ensemblistes, pour former le cluster.

3 Expérimentation et visualisations

L'approche proposée utilise la librairie `iPlots` (Urbanek et Theus (2003)) de `R` (http ://www.r-project.org). Cette librairie développée en Java, permet de visualiser et d'interagir avec les graphiques en utilisant des méthodes classiques de visualisation (nuage de points, histogramme, coordonnées parallèles (Inselberg (1985)), ...). Elle permet de sélectionner un ensemble de données à partir de plusieurs graphiques représentants les mêmes données tout en répercutant la sélection sur les différentes vues. Nous nous basons sur cette interaction pour sélectionner les données qui semblent appartenir à un groupe homogène (formant un cluster) en rajoutant des fonctionalités classiques de classification non supervisée en particulier et de fouille de données en général (comparaison avec des algorithmes classiques (K-means (McQueen (1967))), régression (figure 2), modélisation des données). Nous présentons notre approche sur 2 ensembles de données artificiels tels que les anneaux (Ring, figure 1-(a)) et les demi-cercles (figure 1-(b)), présentant la particularité de chevauchement. Nous projetons également les résultats obtenus (figures 1-(a) et 1-(b)) en utilisant l'algorithme des K-means (McQueen (1967)). On voit clairement sur les figures 1-(a) et 1-(b), résultats des K-means, que certaines données sont mal classées, le clsuter 1 représenté en bleu par des petits cercles re-

prend tous les points sur la partie supérieure de la projection et le clsuter 2 en rouge représenté par des étoiles, reprend tous les autres points.



(a) (b)

(c) (d)

FIG. 1 – *Visualisation des résultats des K-means sur l'ensemble de données Ring (a) et sur l'ensemble de données demi-cercle (b). Les figures (c et d) représentent les visualisations des résultats de l'approche interactive de classification non supervisée sur l'ensemble de données Ring et demi-cercle respectivement.*

4 Conclusion et travaux futurs

Nous avons présenté dans cet article une approche interactive de classification non supervisée. L'approche est basée sur des projections des données en petites dimensions pour sélectionner les groupes d'individus homogènes. L'approche est efficace sur des ensembles de données (anneaux, demi cercles, ...) où la perception visuelle est, bien souvent, plus efficace que les outils classiques de classification. L'approche proposée a été testée et comparée sur plusieurs jeux de données adaptés (données qui se chevauchent, les anneaux, ...) ; les visualisations montrent l'efficacité de l'approche et de l'outil proposé. Ces expérimentations préliminaires sont encourageantes et plusieurs perspectives peuvent se découler de ce travail. Notre objectif, à moyen terme, est de proposer une combinaison avec des méthodes de sélection ou de réduction de dimensions pour pouvoir traiter les données multidimensionnelles et rendre ainsi l'interprétation

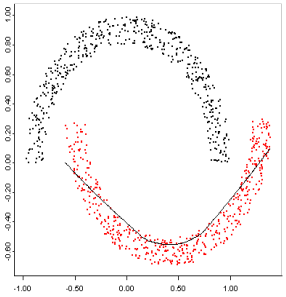


FIG. 2 – *Modélisation par régréssion d'un cluster.*

plus facile par l'utilisateur ou le spécialiste des données. Nous pensons, également, proposer une étape interactive, coopérative, avec les méthodes classiques, afin d'assister les procédures de classification non supervisée dans les étapes d'initialisation par exemple.

Références

aPearson, K. (1901). On lines and planes of closest fit to systems of points in space. In *Philosophical Magazine*, Volume 2, pp. 559–572.

Boudjeloud, L. et F. Poulet (2005). Visual interactive evolutionary algorithm for high dimensional data clustering and outlier detection. In *proceedings of The Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining*. PAKDD'05.

Carr, D. B., R. J. Littlefield, et W. L. Nicholson (1987). Scatterplot matrix techniques for large n. *Journal of the American Statistical Association 82*(398), 424–436.

Inselberg, A. (1985). The plane with parallel coordinates. In *Special Issue on Computational Geometry*, Volume 1, pp. 69–97.

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Poulet, F. (2004). Svm and graphical algorithms : A cooperative approach. In *IEEE ICDM'04, the 4th International Conference on Data Mining*, pp. 499–502.

Urbanek, S. et M. Theus (2003). `iPlots`, high interaction graphics for `R`. In K. Hornik, F. Leisch, et A. Zeileis (Eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, Number ISSN 1609-395X, pp. 1–11.

Summary

We are interested in this paper by interactive clustering. We propose an approach allow us clustering data interactively where the visual perception is accurately more than classical clustering methods. Experiments and comparisons on artificial data sets show the effectiveness of the proposed approach.

Premiers résultats pour un assistant utilisateur en fouille visuelle de données

Abdelheq Et-tahir Guettala[*], Fatma Bouali[***,*], Christiane Guinot[**,*], Gilles Venturini[*]

[*]Université François Rabelais Tours, Laboratoire d'Informatique
64 avenue Jean Portalis, 37200 Tours, France
{abdelheq.guettala,venturini}@univ-tours.fr
[**]CERIES, 20 rue Victor Noir, 92521 Neuilly-sur-Seine Cedex
christiane.guinot@ceries-lab.com
[***]Université de Lille2, IUT, Dpt STID
25-27 Rue du Maréchal Foch, 59100 Roubaix, France
Fatma.Bouali@univ-lille2.fr

Résumé. Nous nous intéressons dans cet article au problème d'automatisation du processus de choix et de paramétrage des visualisations en fouille de données. Pour résoudre ce problème, nous avons développé un assistant utilisateur qui effectue 2 étapes : le système commence par proposer aux utilisateurs différents appariements entre la base de données à visualiser et les visualisations qu'il gère. Ces appariements sont générés par une heuristique utilisant une base de connaissances sur les visualisation et la perception visuelle. Ensuite, afin d'affiner les différents paramétrages suggérés par le système, nous utilisons un algorithme génétique interactif qui permet aux utilisateurs d'évaluer et d'ajuster visuellement ces paramétrages.

1 Introduction

Les assistants utilisateur en fouille visuelle de données ont pour but d'aider l'utilisateur à trouver une représentation visuelle de ses données qui soit la plus adaptée et la plus informative possible. Si l'on se limite strictement au domaine de la fouille visuelle de données, il existe peu d'assistants utilisateur cités dans la littérature qui utilisent un processus automatisé pour aider les utilisateurs dans le choix et le paramétrage des visualisations. Mackinlay (1986) a développé un outil de présentation graphique (APT) s'appuyant sur un système à base de règles (règles d'expressivité et d'efficacité) pour automatiser le processus de visualisation. Cependant, APT est limité dans le nombre d'appariements qu'il suggère entre les attributs de données et les attributs visuels, et il n'utilise pas d'étape interactive pour permettre à l'utilisateur de modifier les paramétrages proposés. ViA est un autre assistant visuel (Healey et al., 1999) plus récent qui utilise un ensemble de moteurs d'évaluation dont chacun permet d'évaluer un attribut visuel dans les appariements générés. Bien qu'il se base sur les caractéristiques des attributs de données à visualiser et un ensemble de règles de perception visuelles (Healey et al., 2008), ViA est limité à une seule visualisation (une carte bidimensionnelle).

Les motivations qui nous ont poussés à développer notre assistant portent donc (1) sur la capacité à gérer des visualisations différentes ainsi que de nouvelles visualisations, (2) sur le fait de proposer une étape visuelle et interactive pour permettre à l'utilisateur d'optimiser le paramétrage mais aussi d'explorer ses données. Nous présentons dans la section 2 notre assistant utilisateur qui permet d'une part d'aiguiller des utilisateurs novices pour le choix et le paramétrage automatique de visualisations, et d'autre part d'ajuster interactivement le paramétrage initialement suggéré avec un algorithme génétique interactif. La section 3 conclut par les premiers résultats que nous avons obtenus et les perspectives possibles faisant suite à ce travail.

2 Modèle proposé

2.1 Modèle des données, des visualisations et des objectifs utilisateur

Nous avons défini un modèle pour représenter formellement les caractéristiques des données utilisateur. Notons $D = \{d_1, ..., d_n\}$ la base de n données à visualiser. Chaque donnée d_i est définie par k attributs de données A_1, .., A_k dont chacun est caractérisé par un type t_i et une importance u_i. Notre système gère différents types de données (numérique/quantitative, symbolique/ordinal ou nominal, temporel, image et son, lien Web). La valeur de u_i est définie dans l'intervalle $[0, 100]$. Elle représente l'intérêt que porte l'utilisateur à l'attribut A_i, et peut être déterminée manuellement par l'utilisateur en fonction de ses connaissances a priori ou automatiquement via des méthodes de sélection de variables.

En s'inspirant principalement des travaux de Bertin (1983) et Card et al. (1999), nous avons développé une base de connaissances qui nous permet de modéliser les différentes visualisations ainsi que les règles de perception visuelle. Nous définissons une visualisation V_i par ses éléments graphiques (points, lignes, formes 2D ou 3D). Chaque élément graphique de V_i est caractérisé par ses attributs visuels A_{i1}, .., A_{im}. A chaque attribut visuel est associé un type visuel t_{ij} (position, taille, couleur, etc.), un type d'attribut de données dont la valeur sera utilisée pour renseigner l'attribut visuel, et un degré d'importance v_{ij}. Les valeurs v_{ij} sont déterminées selon une matrice d'importance "type d'attribut visuel $\times$ type d'attribut de données" dont les valeurs sont déterminées par des études comme (Mackinlay, 1986). De plus, pour chacune des visualisations de la base de connaissances, nous décrivons quels sont les objectifs utilisateurs qu'elle pourra atteindre (découvrir des classes, avoir une vue d'ensemble, etc.), sa dimension visuelle (1D, 2D, 3D) et aussi sa catégorie visuelle (temporelle, relationnelle, etc.).

2.2 Algorithme d'appariement et choix d'une visualisation

La phase d'appariement entre les visualisations et les données commence par la sélection des visualisations qui sont compatibles avec les objectifs de l'utilisateur. Ensuite, pour chaque visualisation compatible, un appariement est tenté avec les données. Pour une visualisation V_i, cela consiste à sélectionner, pour chaque attribut visuel A_{ij}, un attribut de données A_i. Pour cela, l'assistant utilise une heuristique qui trie les attributs de données par type puis par ordre décroissant de leur importance. Les attributs visuels sont triés de la même manière. Ensuite, l'appariement consiste à établir des correspondances dans l'ordre indiqué par les deux tris. A la fin de cette phase, plusieurs visualisations avec chacune un paramétrage sont proposées à

l'utilisateur. Pour l'aider à choisir une visualisation, l'assistant prévisualise chacune d'elles avec les données D en ordonnant les visualisations de manière décroissante selon un score d'appariement (produit scalaire entre les importances).

2.3 Algorithme génétique interactif

Une fois que l'utilisateur a choisi une visualisation, il peut améliorer encore celle-ci grâce à une étape interactive. Cette étape a deux objectifs : tout d'abord, le paramétrage d'une visualisation ayant une dimension subjective importante, il peut être intéressant pour l'utilisateur d'améliorer encore la visualisation de manière à ce qu'elle soit encore plus informative à ses yeux. Ensuite, l'utilisateur peut ne pas connaître l'importance réelle des attributs de données, cette importance pouvant se réveler au cours d'une exploration interactive des données. Pour cette étape, nous avons défini un algorithme génétique interactif (AGI) (Dawkins, 1986). Dans cet AGI, chaque individu $I_{i=1..8}$ de la population P représente un nouveau vecteur de poids d'attributs de données, ce vecteur venant influencer directement l'appariement avec les attributs visuels. L'utilisateur peut sélectionner les paramétrages qui lui semblent être les meilleurs sur la base d'une représentation visuelle de ses derniers, appliquée sur ses données D (voir figure 1). Les individus sélectionnés servent alors à générer une nouvelle population de paramétrages par le biais d'opérateurs de croisement et de mutation s'appliquant sur des vecteurs de poids. Cet AGI permet aussi à l'utilisateur d'explorer les données en combinant de manière différentes les attributs de données.

3 Résultats et conclusions

Toutes les étapes décrites précédemment ont été implémentées dans notre plateforme de visualisation VRMiner2. A titre d'exemple, avec la base de données Wine (13 attributs numérique et 1 attribut classe) à laquelle nous avons ajouté 1 attribut image (une image associée à chaque classe), le système suggère intialement 9 visualisations avec comme qualité d'appariement (64%, 60%, 61%, 56%, 56%, 51%, 56%, 51%, 51%). En choisissant la première visualisation, l'AGI génère 8 paramétrages potentiels qui sont alors visualisés (voir figure 1) et évalués par l'utilisateur en sélectionnant ceux qui traduisent le mieux ses objectifs. L'utilisateur peut affiner encore les paramétrages ou les enregistrer pour une utilisation ultérieure.

Nous avons présenté les premiers résultats d'un assistant utilisateur qui intègre une base de connaissances et utilise une méthode d'évaluation qui favorise la participation des utilisateurs dans le processus de choix et de paramétrage de leurs visualisations. Nous allons ensuite compléter ce travail en augmentant le nombre de visualisations représentées dans la base, en réalisant une évaluation utilisateur et en effectuant un retour de l'utilisateur vers la base de connaissances (mise à jour des poids en fonction des expériences).

Références

Bertin, J. (1983). *Semiology of graphics*. Berlin: University of Wisconsin Press.

Card, S. K., J. D. Mackinlay, et B. Shneiderman (1999). *Readings in Information Visualization: Using Vision to Think (Interactive Technologies)*. Morgan Kaufmann.

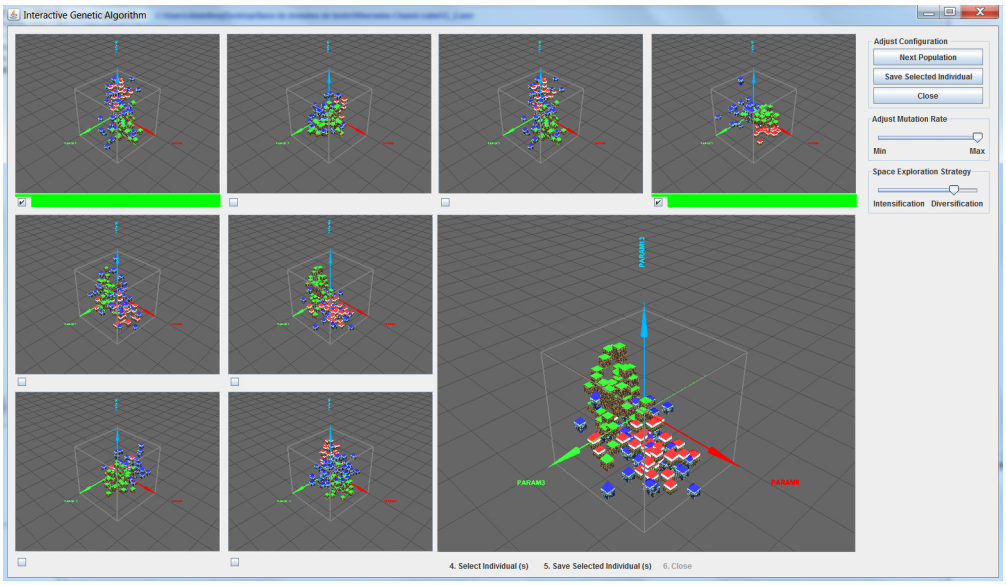


FIG. 1 – *Interface d'optimisation génétique du paramétrage.*

Dawkins, R. (1986). *The Blind Watchmaker*. San Mateo: Norton.

Healey, C., S. Kocherlakota, V. Rao, R. Mehta, et R. St Amant (2008). Visual perception and mixed-initiative interaction for assisted visualization design. *IEEE transactions on visualization and computer graphics 14*, 396–411.

Healey, C. G., R. S. Amant, et M. S. ElHaddad (1999). Via : A perceptual visualization assistant. *SPIE proceedings series 5*, 2–11.

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics 5*, 110–141.

Summary

We deal in this paper with the problem of automating the process of choosing a visualization and its parameters in data mining. To solve this problem, we have developed a user assistant that performs 2 steps: the system starts by suggesting to users different matchings between their database and the possible visualizations. These matchings are generated by using a knowledge-based heuristic. Then, in order to refine the different parameter settings suggested by the system, we use an interactive genetic algorithm which allows users to visually evaluate and adjust these parameters.

Classification spectrale : interprétation et résultats

Sandrine Mouysset*, Joseph Noailles*, Daniel Ruiz*

*Institut de Recherche en Informatique de Toulouse (IRIT-ENSEEIHT),
Université de Toulouse, 2 rue Camichel, 31000 Toulouse,
{sandrine.mouysset,joseph.noailles,daniel.ruiz}@enseeiht.fr

Résumé. La classification spectrale consiste à créer à partir des éléments spectraux de la matrice affinité gaussienne, un espace de dimension réduite dans lequel les données seront classées. Cette méthode non supervisée est principalement basée sur la mesure d'affinité gaussienne, son paramètre et ses élements spectraux. Cependant, les questions sur la séparabilité des classes dans l'espace de projection spectrale et sur le choix de ce paramètre restent ouvertes. Une nouvelle interprétation sur le fonctionnement de la classification spectrale pour un ensemble fini de données est proposée via les Equations aux Dérivées Partielles et les Eléments Finis.

1 Introduction

La classification spectrale, introduite par Ng et al. (2002), consiste à créer à partir de la similarité entre les points un espace de dimension réduite dans lequel les données sont regroupées en classes. Cet espace est défini en sélectionnant les plus grands vecteurs propres issus d'une matrice gaussienne normalisée dépendante d'un paramètre σ. Les problèmes inhérents à cette méthode consistent à expliquer comment la classification dans l'espace de projection spectrale caractérise la classification dans l'espace d'origine et à étudier le rôle de σ dans le partitionnement. Initialement formulée comme un problème de maximisation de trace via le critère de coupe de graphe normalisée Shi et Malik (2000), plusieurs travaux ont été, par ailleurs, menés pour expliquer le fonctionnement de la classification spectrale. Diverses définitions pour σ issues d'interprétations physiques Von Luxburg (2007) ont été suggérées. Comme le but est de regrouper des points par le biais de leur proximité, Belkin et Niyogi (2002) ont exploité cette propriété de voisinage en introduisant une étape préalable de graphe d'adjacence. Ils aboutissent ainsi à considérer l'équation de la chaleur sur des variétés et à interpréter la matrice affinité gaussienne comme une discrétisation de l'opérateur de Laplace-Beltrami. Nadler et al. (2005) donnent une interprétation probabiliste basée sur un modèle de diffusion. Tous ces résultats sont établis asymptotiquement pour un grand nombre de points. Cependant, d'un point de vue numérique, la classification spectrale partitionne correctement un ensemble fini de points.

Nous proposons donc une nouvelle interprétation où les données représenteront la discrétisation de sous-ensembles. Ainsi, les vecteurs propres de la matrice gaussienne seront, pour une bonne valeur de σ, la représentation discrète de fonctions à support sur un seul de ces sous-ensembles.

2 Interprétation

Comme les éléments spectraux de la classification spectrale ne fournissent pas explicitement de critère topologique pour un ensemble discret de données, nous revenons à une formulation continue où les classes sont des sous-ensembles disjoints comme le montre la figure 1. Ainsi on propose de relier la classification d'un ensemble fini de points à une partition d'ouverts de l'espace $\mathbb{R}^p$ par la définition suivante.

Définition 1 (Classification compatible) *Soit un ouvert Ω formé de k composantes connexes distinctes $\Omega_i,\ i \in \{1,..,k\}$ tel que : $\Omega = \bigcup_{i=1}^{k} \Omega_i$. Soit $\mathcal{P}$ un ensemble de points $\{x_i\}_{i=1}^{N}$ de l'ouvert Ω. On note par $\mathcal{P}_j$, pour $j = \{1,..,k\}$, l'ensemble non vide des points de $\mathcal{P}$ appartenant à la composante connexe Ω_j de Ω : $\mathcal{P}_j = \Omega_j \cap \mathcal{P}, \forall j \in \{1,..,k\}$.*
Soit $\mathcal{C} = \{C_1,..,C_{k'}\}$ un partitionnement de l'ensemble $\mathcal{P}$, c'est-à-dire $\forall i \neq j,\ \mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ et $\mathcal{P} = \bigcup_{i=1}^{k'} \mathcal{C}_i$. Alors $\mathcal{C}$ est appelé classification compatible si $k' = k$ et $\forall j = \{1,..,k'\}, \exists i \in \{1,..,k\},\ C_j = \mathcal{P}_i$.

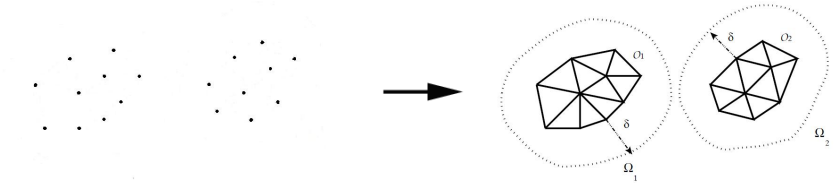


FIG. 1 – *Principe de l'interprétation de la classification spectrale via les éléments finis*

Rappelons que le coefficient de l'affinité gaussienne entre deux points x_i et x_j de $\mathbb{R}^p$, noté A_{ij}, est défini par $A_{ij} = \exp\left(-\|x_i - x_j\|^2/2\sigma^2\right)$. Une relation entre l'affinité A_{ij} et le noyau de Green K_H de l'équation de la chaleur sur $\mathbb{R}_+^* \times \mathbb{R}^p$, défini par $K_H(t,x) = (4\pi t)^{-\frac{p}{2}} \exp\left(-\|x\|^2/4t\right)$, peut être directement établie comme suit :

$$A_{ij} = (2\pi\sigma^2)^{\frac{p}{2}} K_H\left(\sigma^2/2, x_i - x_j\right),\ \forall (i,j) \in \{1,..,N\}. \tag{1}$$

Ce retour à une formulation continue est effectué à l'aide des Elements Finis (figure 1) introduisant ainsi un pas de discrétisation, noté h, et un opérateur d'interpolation de Lagrange Π_h permettant le passage de $C^0(\bigcup_{i=1}^{k} \bar{\mathcal{O}}_i)$ à l'espace d'approximation V_h de dimension finie. Ainsi, les vecteurs propres de la matrice affinité A sont interprétés comme des fonctions propres d'un opérateur. En effet, avec les Elements Finis dont les noeuds correspondent aux données d'origines, une représentation d'une fonction L^2 est donnée par sa valeur nodale. Donc on peut interpréter la matrice d'affinité gaussienne A et ses vecteurs propres comme les représentations respectives d'un opérateur L^2 et d'une fonction L^2. L'opérateur dont la représentation en Eléments Finis concorde avec la définition de A est, d'après (1), le noyau de l'équation de la chaleur sur $\mathbb{R}_+^* \times \mathbb{R}^p$. A partir de cette équation (1), la propriété géométrique suivante peut être démontrée.

Proposition 1 (Propriété géométrique) *Soit v_{n_i} une fonction propre de $S_D(t)$ associée à la valeur propre λ_{n_i} telle que $S_D(t)v_{n_i} = \lambda_{n_i} v_{n_i}$ pour $i \in \{1, ..k\}$. Pour $W_{n_i} = \Pi_h v_{n_i} \in V_h$ et tout $h^{(3p+2)/2} < t < \delta^2$, il existe $\alpha > 0$ tel que W_{n_i} verifie :*

$$\alpha(A + \mathbb{I}_N)\, W_{n_i} = e^{-\lambda_{n_i} t} W_{n_i} + \psi(t, h),$$

où $h > 0$ pas de discrétisation, $\|\psi(t,h)\|_2 \to 0$ quand $(h,t) \to 0$ et $\delta \to 0$.

Comme le spectre de l'opérateur S_H (convolution par K_H) est essentiel, on ne peut pas directement interpréter les vecteurs propres de A comme une représentation des fonctions propres de S_H. Cependant, en introduisant K_D, le noyau de Green de l'équation de la chaleur suivante sur un domaine borné Ω avec conditions de Dirichlet sur $\partial\Omega$ et $f \in L^2(\Omega)$:

$$(\mathcal{P}_\Omega) \begin{cases} \partial_t u - \Delta u = 0 \text{ in } \mathbb{R}^+ \times \Omega, \\ u(t=0) = f, \text{ in } \Omega, \\ u = 0, \text{ on } \mathbb{R}^+ \times \partial\Omega, \end{cases}$$

une estimation de la différence entre K_H et K_D peut être établie sur un domaine borné $\mathcal{O}$ où Ω contient strictement $\mathcal{O}$. Maintenant, l'opérateur S_D (convolution par K_D : $(S_D(t)f)(x) = (K_D(t,\cdot) *_x f)(x)$) admet des fonctions propres (v_{n_i}) dans $H_0^1(\Omega)$. Ces v_{n_i} ont la propriété d'avoir leurs supports inclus sur une seule des composantes connexes. Ensuite, on montre que l'opérateur S_H admet les v_{n_i} comme fonctions propres plus un résidu. Enfin, en utilisant l'approximation par les Eléments Finis (projection par Π_h dans l'espace d'approximation V_h) et une condensation de masse, on montre (proposition 1) que les vecteurs propres de A sont une représentation de ces fonctions propres W_{n_i} plus un residu (noté ψ) fonction du paramètre t et du pas de discrétisation h.

3 Expérimentations numériques

On considère sur la figure 2 (a) un exemple représentant deux couronnes où les classes ne sont pas séparables par hyperplans. Les figures 2 (b)-(c) représentent la propriété géométrique des fonctions propres de l'opérateur S_D associées à la première valeur propre sur chaque composante connexe. Pour montrer la propriété d'invariance géométrique, la corrélation ω entre les fonctions propres discrétisées v_{n_i} et les vecteurs propres de A est représentée sur la figure 2 (d) en fonction du paramètre t. Les bornes de définition du paramètre t, indiquées par les lignes verticales noires, indiquent un intervalle où le coefficient de corrélation ω est maximal respectivement pour les deux composantes connexes. Enfin, pour une valeur de t appartenant à cet intervalle, les vecteurs propres de la matrice affinité pour lesquels le coefficient de projection avec les fonctions propres de S_D (b)-(c) respectives est maximal sont représentés sur la figure 2 (e)-(f) montrant la conservation de la propriété géométrique.

4 Conclusion

En interprétant la matrice d'affinité gaussienne comme la discrétisation du noyau de la chaleur définie sur l'espace entier et en utilisant les éléments finis, les vecteurs propres de

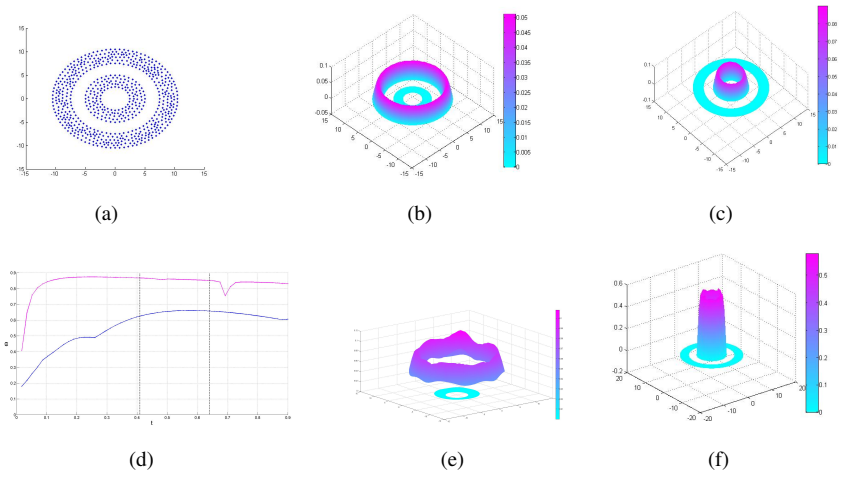


FIG. 2 – *(a) Ensemble de données ($N = 669$), (b)-(c) Fonctions propres discrétisées de S_D, (d) Corrélation ω en fonction de t, (e)-(f) Vecteurs propres issus de A.*

la matrice affinité sont la représentation asymptotique de fonctions dont le support est inclus dans une seule composante connexe. Cette interprétation a permis de définir un intervalle de valeurs pour le paramètre de l'affinité gaussienne dans lesquel les propriétés de classification sont vérifiées.

Références

Belkin, M. et P. Niyogi (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems 14*(3).

Nadler, B., S. Lafon, R. Coifman, et I. Kevrekidis (2005). Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck operators. *Arxiv preprint math.NA/0506090*.

Ng, A. Y., M. I. Jordan, et Y. Weiss (2002). On spectral clustering : analysis and an algorithm. *Proc.Adv.Neural Info.Processing Systems*.

Shi, J. et J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(8), 888–905.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing 17*(4), 395–416.

Summary

The Spectral Clustering consists in creating, from the spectral elements of a Gaussian affinity matrix, a low-dimension space in which data are grouped into clusters. This unsupervised method is mainly based on Gaussian affinity measure, its parameter and its spectral elements. However, questions about the separability of clusters in the projection space and the spectral parameter choices remain open. We propose a new interpretation of spectral clustering for a finite discrete data set via Partial Differential Equations and Finite Elements theory.

Comparaison topologique de mesures de proximité

Rafik Abdesselam, Djamel Abdelkader Zighed

Laboratoire ERIC, Université Lumière Lyon 2,
5, Avenue Pierre Mendès-France, 69676 Bron Cedex
rafik.abdesselam@univ-lyon2.fr, http://eric.univ-lyon2.fr/~rabdesselam/fr/
abdelkader.zighed@univ-lyon2.fr, http://eric.univ-lyon2.fr/~zighed

Résumé. Le choix d'une mesure de proximité entre objets a un impact direct sur les résultats de toute opération de classification, de comparaison, d'évaluation ou de structuration d'un ensemble d'objets. Pour un problème donné, l'utilisateur est amené à choisir une parmi les nombreuses mesures de proximité existantes. Or, selon la notion d'équivalence choisie, comme celle basée sur les préordonnances, certaines sont plus ou moins équivalentes. Nous proposons une nouvelle approche de comparaison de mesures de proximité, basée sur l'équivalence topologique. Ce nouveau concept d'équivalence fait appel à la structure de voisinage local. Nous illustrons le principe de cette approche sur un exemple simple.

1 Introduction

On fait souvent appel aux mesures de proximité lorsqu'on veut comparer, évaluer ou structurer un ensemble d'objets. Ces mesures sont-elles toutes équivalentes ? Cette problématique est importante, en effet, est-ce que, par exemple, la façon dont on mesure la similarité ou la dissimilarité entre objets affecte les résultats d'une classification en groupes ? La comparaison de mesures de proximité permet de vérifier si elles ont des propriétés communes Lerman (1967), Lesot et al. (2009). Pour comparer deux mesures de proximité, l'approche consiste jusque-là, à comparer les valeurs des matrices de proximité induites Batagelj et Bren (1995).

Dans Lerman (1967), l'auteur s'intéresse aux préordres induits par deux mesures de proximité et évalue leur degré de ressemblance par la concordance entre les préordres induits sur l'ensemble des couples d'objets. D'autres auteurs, Schneider et Borlund (2007) évaluent l'équivalence entre deux mesures par un test statistique entre les matrices de proximité.

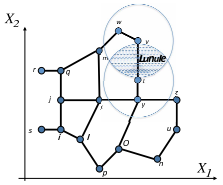
Nous proposons d'évaluer statistiquement l'équivalence entre deux mesures de proximité à partir des matrices d'adjacence associées et de regrouper ces mesures de proximité en classes selon leurs similitudes . Si la structure de voisinage entre objets, induite par une mesure de proximité ne change pas par rapport à celle d'une autre mesure de proximité, cela signifie que les ressemblances locales entre individus n'ont pas changé. Dans ce cas, on dira que les deux mesures de proximité sont en équivalence topologique.

2 Graphe topologique

Considérons un ensemble $E = \{x, y, z, \ldots\}$ de $n = |E|$ objets plongés dans R^p. On peut, au moyen d'une mesure de proximité u définir une relation binaire de voisinage V_u sur $E \times E$.

Pour construire cette relation de voisinage, on peut, par exemple, construire l'Arbre de Longueur Minimum (ALM), le Graphe de Gabriel (GG) ou encore le Graphe des Voisins Relatifs (GVR) Toussaint (1980), dont tous les couples de points voisins vérifient la propriété suivante : $u(x, y) \leq \max(u(x, z), u(y, z))$; $\forall z \in E - \{x, y\}$. Dans ce cas, $V_u(x, y) = 1$ sinon $V_u(x, y) = 0$. Où, V_u est la matrice d'adjacence associée au graphe GVR, formée de 0 et de 1. Ce qui signifie, sur le plan géométrique, que l'hyper-Lunule (intersection des deux hypersphères centrées sur les deux points) est vide.

Mesure	Formule
u_1 - Euclidean	$u_E(x, y) = \sqrt{\sum_{j=1}^{p}(x_j - y_j)^2}$
u_2 - Mahalanobis	$u_{Mah}(x, y) = \sqrt{(x - y)^t \sum^{-1}(x - y}$
u_3 - Manhattan	$u_{Man}(x, y) = \sum_{j=1}^{p} \lvert x_j - y_j \rvert$
u_4 - Chi-square	$u_{\chi^2}(x, y) = \sum_{j=1}^{p} \frac{(x_j - m_j)^2}{m_j}$
u_5 - Minkowski	$u_{Min_\gamma}(x, y) = (\sum_{j=1}^{p} \lvert x_j - y_j \rvert^\gamma)^{\frac{1}{\gamma}}$
u_6 - Tchebytchev	$u_{Tch}(x, y) = \max_{1 \leq j \leq p} \lvert x_j - y_j \rvert$
u_7 - Cosine Dissimilarity	$u_{Cos}(x, y) = 1 - \frac{<x, y>}{\lVert x \rVert \lVert y \rVert}$
u_8 - Pearson's Correlation	$u_\rho(x, y) = 1 - \lvert \rho(x, y) \rvert$
u_9 - Canberra	$u_{Can}(x, y) = \sum_{j=1}^{p} \frac{\lvert x_j - y_j \rvert}{\lvert x_j \rvert + \lvert y_j \rvert}$
u_{10} - Squared Chord	$u_{SC}(x, y) = \sum_{j=1}^{p} (\sqrt{x_j} - \sqrt{y_j})^2$
u_{11} - Weighted Euclidean	$u_{WE}(x, y) = \sqrt{\sum_{j=1}^{p} \alpha_j (x_j - y_j)^2}$
u_{12} - Jeffrey Divergence	$u_{JD}(x, y) = \sum_{j=1}^{p} (x_j \log \frac{x_j}{m_j} + y_j \log \frac{y_j}{m_j})$
u_{13} - Normalized Euclidean	$u_{NE}(x, y) = \sqrt{\sum_{j=1}^{p} \frac{(x_j - y_j)^2}{\sigma_j^2}}$



V_u	..	x	y	z	..
..	..	..	..	..	..
x	..	1	1	0	..
y	..	1	1	1	..
z	..	0	1	1	..
t	..	0	1	0	..
u	..	0	0	1	..
..	..	..	..	..	..

TAB. 1 – *Mesures de proximité - Exemple de GVR et de sa matrice d'adjacence.*

Pour toute propriété de voisinage (ALM, GG, GVR, etc.), chaque mesure de proximité u génère une structure topologique sur les objets dans E qui est totalement décrite par une matrice d'ajacence V_u. Nous limitons ce travail à la comparaison des mesures de proximité dans R^p, présentées dans le Tableau 1, avec la structure topologique GVR.

- Comparaison de deux mesures de proximité

Soient V_{u_i} et V_{u_j} les matrices d'adjacence associées à deux mesures de proximité u_i et u_j. Pour comparer le degré d'équivalence topologique entre deux mesures de proximité, nous proposons de tester si les matrices d'adjacence associées sont statistiquement différentes ou pas, en utilisant un test non paramétrique sur données appariées. Ces matrices, binaires et symétriques d'ordre n, sont dépliées selon deux vecteurs de composantes appariées, formées des $n(n-1)/2$ valeurs supérieures (ou inférieures) de la diagonale. Le degré d'équivalence topologique entre deux mesures de proximité est évalué à partir du coefficient de concordance de Kappa Cohen (1960), calculé sur le tableau 2×2 de contingence formé par les deux vecteurs :

$$\kappa = \kappa(V_{u_i}, V_{u_j}) = \frac{\Pi_o - \Pi_e}{1 - \Pi_e} \qquad \text{avec} \qquad \begin{cases} \Pi_o : \text{proportion observée} \\ \Pi_e : \text{proportion aléatoire} \end{cases}$$

Nous formulons ensuite, l'hypothèse nulle $H_0 : \kappa = 0$, d'indépendance d'accord ou de concordance. La concordance est d'autant plus élevée que sa valeur tend vers $+1$, parfaite ou maximale si $\kappa = 1$. Il est égal à -1 dans le cas d'une discordance parfaite.

- Classification de mesures de proximité

Soient $D_{u_i}(E \times E)$ et $D_{u_j}(E \times E)$ les tableaux de distances associés aux mesures de proximité u_i et u_j. Chacune de ces distances engendre une structure topologique sur les objets E. Une telle structure est parfaitement décrite par sa matrice d'adjacence. Pour mesurer le degré de ressemblance entre les graphes, il suffit de compter le nombre de discordances entre les deux matrices d'adjacence V_{u_i} et V_{u_j} associées aux deux structures toplogiques :

$$S(V_{u_i}, V_{u_j}) = \frac{1}{n^2} \sum_x \sum_y \delta_{ij}(x,y) \quad \text{avec} \quad \delta_{ij}(x,y) = \begin{cases} 1 & \text{si } V_{u_i}(x,y) = V_{u_j}(x,y) \\ 0 & \text{sinon} \end{cases}$$

La valeur 1 de la mesure de similarité S signifie que les deux matrices d'adjacence sont identiques et par conséquent, la structure topologique induite par les deux mesures est la même. Dans ce cas, on parle d'équivalence topologique entre les deux mesures de proximité. La valeur 0 signifie que la topologie a totalement changé. A partir de cette mesure S, nous pouvons comparer et classer les mesures de proximité selon leur degré de ressemblance.

- Exemple d'application

L'approche proposée est illustrée à partir d'un jeu de données relativement simple, celui des Iris de Fisher (UCI-Repository).

S κ	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}
u_E	**1**	.456	.845	.835	.767	.341	.588	.753	**1**	.753	.753	.400	.720
u_{Mah}	.876	**1**	.336	.456	.410	.378	.301	.301	.456	.222	.301	.357	.357
u_{Man}	.964	.840	**1**	.767	.705	.301	.689	.767	.845	.767	.767	.281	.811
u_{Min}	.964	.876	.947	**1**	.845	.258	.506	.670	.835	.670	.670	.400	.720
u_{Tch}	.947	.858	.929	.964	**1**	.378	.611	.767	.767	.767	.767	.432	.735
u_{Cos}	.858	.858	.840	.840	.858	**1**	.176*	.341	.341	.341	.341	.880	.240
u_{Can}	.911	.840	.929	.893	.911	.822	**1**	.753	.588	.753	.753	.080*	.640
u_{SC}	.947	.840	.947	.929	.947	.858	.947	**1**	.753	**1**	**1**	.320	.640
u_{WE}	**1**	.876	.964	.964	.947	.858	.911	.947	**1**	.753	.753	.400	.720
u_{χ^2}	.947	.840	.947	.929	.947	.858	.947	**1**	.947	**1**	**1**	.320	.640
u_{JD}	.947	.840	.947	.929	.947	.858	.947	**1**	.947	**1**	**1**	.320	.640
u_ρ	.867	.849	.831	.867	.867	.973	.796	.849	.867	.849	.849	**1**	.300
u_{NE}	.938	.849	.956	.938	.938	.831	.920	.920	.938	.920	.920	.840	**1**

κ : partie supérieure de la diagonale, S : partie inférieure de la diagonale. *Non significatif avec un risque d'erreur $\leq 5\%$

TAB. 2 – $S(V_{u_i}, V_{u_j})$ *: Similarités &* $\kappa(V_{u_i}, V_{u_j})$ *: Coefficients Kappa.*

La partie supérieure de la diagonale du tableau 2 présente les valeurs exactes de la statistique de test de Kappa. On peut conclure, avec un risque d'erreur de 5%, que seules les mesures des couples (u_{Cos}, u_{Can}) et (u_{Can}, u_ρ) ne sont pas significativement équivalente en topologie. A noter que les mesures des couples (u_E, u_{WE}), (u_{SC}, u_{χ^2}), (u_{SC}, u_{JD}) et (u_{χ^2}, u_{JD}) sont en parfaite équivalence topologique ($\kappa = 1$).

Les valeurs des similarités, partie inférieure de la diagonale du tableau 2, sont assez proches de 1 et les mesures de proximité des couples (u_E, u_{WE}), (u_{SC}, u_{JD}), (u_{SC}, u_{χ^2}) et (u_{χ^2}, u_{JD}) sont en parfaite équivalence topologique ($S(V_{u_i}, Vu_j) = 1$). On peut visualiser ces mesures de proximité en appliquant, par exemple, une classification hiérarchique ascendante selon le critère de Ward, Figure 1.

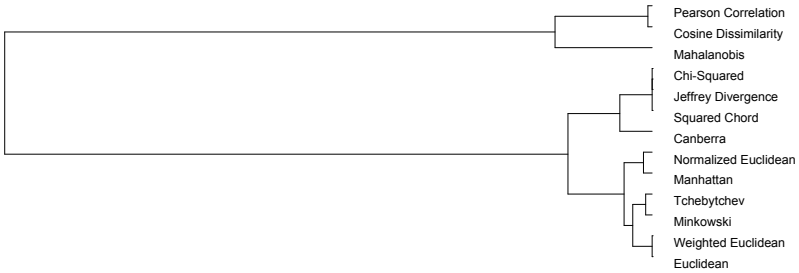


FIG. 1 – *Classification des mesures de proximité*

3 Conclusion et perspectives

Ce travail propose une nouvelle approche d'équivalence topologique entre mesures de proximité, basée sur la notion de graphes de voisinage. L'application d'un test non paramétrique sur les matrices d'adjacence associées aux mesures de proximité, a permis de donner une signification statistique et de valider ou pas l'équivalence topologique, c'est-à-dire, si vraiment elles induisent ou pas la même structure de voisinage sur les objets. Cette approche s'étend à d'autres types de données (binaires, qualitatives). Nous envisageons d'étudier l'influence des données et de la structure de voisinage sur les résultats ainsi que l'effet du choix de la méthode de classification sur les groupes de mesures de proximité.

Références

Batagelj, V. et M. Bren (1995). Comparing resemblance measures. *Journal of classification 12*, 73–90.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *20*, 27–46.

Lerman, I. (1967). *Indice de similarité et préordonnance associée, Ordres*. Travaux du séminaire sur les ordres totaux finis, Aix-en-Provence.

Lesot, M.-J., M. Rifqi, et H. Benhadda (2009). Similarity measures for binary and numerical data: a survey. *IJKESDP 1*(1), 63–84.

Schneider, J. et P. Borlund (2007). Matrix comparison, part 2: Measuring the resemblance between proximity measures or ordination results by use of the mantel and procrustes statistics. *Journal American Society for Information Science and Technology 58*(11), 1596–1609.

Toussaint, G. (1980). The relative neighbourhood graph of a finite planar set. *Pattern recognition 12*(4), 261–268.

Summary

According to the notion of equivalence, like the one based on pre-ordering, some of the proximity measures are more or less equivalent, which means that they produce, more or less, the same results. We introduce a new approach to comparing proximity measures. It is based on topological equivalence which exploits the concept of local neighbors.

Classification multi-critère fondée sur des distances pondérées de Tchebycheff pour données relationnelles

S. Queiroz*, F.A.T. De Carvalho*
Y. Lechevallier**

*Centro de Informática – CIn/UFPE, Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária, 50.740-560, Recife - PE, Brésil
{srmq,fatc}@cin.ufpe.br,
**INRIA, Paris-Rocquencourt - 78153 Le Chesnay cedex, France
Yves.Lechevallier@inria.fr

Résumé. Nous présentons un nouvel algorithme de classification capable de tenir en compte simultanément plusieurs tableaux de dissimilarité. L'algorithme utilise un critère d'agrégation non linéaire, les distances pondérées de Tchebycheff, plus approprié que les combinaisons linéaires pour la construction de solutions de compromis. Nous présentons une application pratique, les résultats obtenus ont été assez conformes aux données utilisées.

1 Introduction

Habituellement, on considère deux types de données qui peuvent être utilisées pour effectuer la tâche de classification automatique : données caractéristiques et données relationnelles. Quand chaque élément est décrit par un vecteur de valeurs quantitatives ou qualitatives, l'ensemble de ces vecteurs est appelé « données caractéristiques ». En revanche, lorsque nous décrivons une relation entre chaque paire d'objets, l'ensemble des relations est appelé « données relationnelles ». Le cas le plus courant de données relationnelles, c'est quand on a une matrice de dissimilarités. Dans plusieurs situations, nous n'avons pas une seule mesure de dissimilarité entre les paires d'objets, mais un vecteur de dissimilarités pour chaque paire, ce que nous appelons « problème multi-critère ». Pour traiter ce type de problème, Frigui et al. (2007) ont proposé CARD, un algorithme fondé sur les algorithmes de classification floue NERF et FANNY. Lechevallier et al. (2010) ont proposé MRDCA, un algorithme de classification dure, qui étend au cas multi-critère l'algorithme mono-critère de De Carvalho et al. (2009). Les deux méthodes utilisent des moyennes pondérées comme fonction d'agrégation.

Dans cet article, nous proposons une nouvelle méthode de classification automatique dure pour le cas multi-critère à partir de données relationnelles. La méthode proposée, WRDCA, modifie l'algorithme MRDCA, en utilisant des distances pondérées de Tchebycheff au lieu de moyennes pondérées. Comme il est connu dans la littérature de multi-critère, l'optimisation d'une fonction fondée sur des distances pondérées de Tchebycheff est plus appropriée pour construire des solutions de compromis (Steuer et Choo, 1983; Wierzbicki, 1986). Cependant, étant un critère non-linéaire, son optimisation n'est pas triviale. Nous montrons comment le critère peut être optimisé en utilisant de la programmation linéaire.

2 Classification multi-critère fondée sur des distances pondérées de Tchebycheff pour données relationnelles

Soient $E = \{e_1, \ldots, e_n\}$ un ensemble de n items à partitionner en K clusters et p matrices de dissimilarité $n \times n$ $(D_1, \ldots, D_p)$, où $D_j[i,l] = d_j(e_i, e_l)$ fournit la dissimilarité entre les objets e_i et e_l selon la matrice de dissimilarité D_j $\forall j = 1, \ldots, p$. Supposons que le prototype g_k de chaque cluster C_k est un élément de E.

Dans WRDCA, on trouve une partition $P = (C_1, \ldots, C_K)$ de E en K clusters et le prototype correspondant $g_k \in E$ pour chaque cluster C_k en P tel que le critère d'adéquation J entre les clusters et leurs prototypes respectifs (fonction objectif) est minimisé. J est défini par :

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

$d^{(k)}(e_i, g_k)$ est la dissimilarité entre un item $e_i \in C_k$ et le prototype $g_k \in E$ paramétrisé par le vecteur de poids $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^p)$ où λ_k^j est le poids de la matrice de dissimilarité D_j pour le cluster C_k, et $d_j(e_i, g_k)$ est la dissimilarité selon la matrice j entre l'item $e_i \in C_k$ et le prototype du cluster $g_k \in E$ $\forall j = 1, \ldots, p$. La matrice de pondération de la pertinence λ est composée par K vecteurs de poids $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^p)$, et change à chaque itération. L'algorithme WRDCA alterne les trois étapes suivantes, jusqu'à la convergence à une valeur de J qui correspond à un minimum local :

- **Étape 1 : Définition des meilleurs prototypes** [P et λ sont restées inchangées]
  Proposition 1. Le $g_k = e_l \in E$ qui minimise J est calculé pour chaque cluster C_k par :

$$l = \arg\min_{h=1}^{n} \sum_{e_i \in C_k} \max_{j=1}^{p} \lambda_k^j d_j(e_i, e_h) \tag{2}$$

- **Étape 2 : Définition de la meilleure matrice** λ [P et g sont restés inchangés]
  Proposition 2. Le vecteur $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^p)$ optimal de chaque cluster k peut être calculé par la résolution du problème de programmation linéaire min-max (mmLP) :

$$\text{Minimiser} \sum_{e_i \in C_k} \max_{j=1}^{p} \lambda_k^j d_j(e_i, g_k), \text{ sujet aux restrictions :} \tag{3}$$

$$0 \leq \lambda_k^j \leq 1 \forall j = 1, \ldots, p \quad \text{et} \quad \sum_{j=1}^{p} \lambda_k^j = 1$$

  Un problème mmLP peut être transformé en un problème de programmation linéaire entière mixte (MILP). Il suffit de noter que : (Burks et Sakallah, 1993)

  1. Une restriction max du type $x_i = \max(x_j, x_k)$ est équivalente à :

$$x_i \geq x_j, x_i \geq x_k \quad \text{et} \quad x_i - x_j \leq c_i M, x_i - x_k \leq (1 - c_i)M \tag{4}$$

     où c_i est une variable entière 0-1 et M est une constante positive large.

  2. Un max d'arité p peut être transformé dans des max binaires, en observant que :

$$\max_{j=1}^{p} \lambda_k^j d_j(e_i, g_k) = \max(\lambda_k^1 d_1(e_i, g_k), m(2)), \text{ où} \tag{5}$$

$$\begin{cases} m(y) = \lambda_k^p d_p(e_i, g_k) \text{ si } y = p \\ m(y) = \max(\lambda_k^y d_y(e_i, g_k), m(y+1)) \text{ si } y \neq p \end{cases}$$

– **Étape 3 : Définition de la meilleure partition** [g et λ sont restés inchangés]
Proposition 3. La partition $P = (C_1, \ldots, C_K)$ qui minimise le critère J est mise à jour selon la règle d'allocation suivante :

$$C_k = \{e_i \in E : d^{(k)}(e_i, g_k) < d^{(h)}(e_i, g_h) \forall h \neq k\} \; \forall k = 1, \ldots, K \tag{6}$$

si le minimum n'est pas unique, e_i sera attribué au cluster avec le plus petit index.

WRDCA effectue les trois étapes de manière itérative, jusqu'à ce que $J(P, \lambda, g)$ atteigne une valeur stable, ce qui caractérise un minimum local. L'algorithme est résumé ci-dessous :

1. **Initialisation**
   (i) Fixer le nombre K de clusters ; **(ii)** Sélectionner de manière aléatoire K objets distincts $g_k \in E$; **(iii)** Fixer la matrice de pondération de la pertinence λ, où $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^p) = (1, \ldots, 1)$; **(iv)** Obtenir la partition $P = (C_1, \ldots, C_K)$, selon (6).
2. **Étape 1 : définition des meilleurs prototypes** [P et λ sont restés inchangés]
   Calculer les prototypes $g_k \in E$ de chaque cluster C_k selon l'équation (2).
3. **Étape 2 : définition de la meilleur matrice** λ [P et g sont restés inchangés]
   Pour chaque k calculer le vecteur λ_k modélisé comme un problème mmLP (3, 4, 5).
4. **Étape 3 : définition de la meilleur partition** [g et λ sont restés inchangés]
   Construire la nouvelle partition $P' = (C'_1, \ldots, C'_K)$ selon (6) et vérifier la convergence :
   convergence $\leftarrow$ true ;
   pour $i = 1$ jusqu'à n faire
   si e_i appartenait au cluster C_m en P et appartient en P' au cluster C'_k avec $k \neq m$
   convergence $\leftarrow$ false ;
   $P \leftarrow P'$;
5. **Critère d'arrêt**
   Si *convergence* = true alors ARRÊTER. Si non, aller en 2. (Étape 1).

3 Application

En mai 2008, le quotidien Los Angeles Times avec l'entreprise KTLA ont mené une enquête pour évaluer les opinions des adultes en Californie sur les questions liées au mariage homossexuel. Les résultats ont été divulgués pour la population dans son ensemble, et aussi pour 22 sous-groupes spécifiques. Pour mesurer la dissimilarité entre deux groupes e_i et e_j de la population par rapport à une question (critère) q, nous utilisons le coefficient d'affinité (Bacelar-Nicolay, 2000), ce qui signifie considérer les vecteurs de m valeurs (m est le nombre de modalités des réponses de la question q) correspondants aux fréquences de chacune des modalités. La dissimilarité entre e_i et e_j selon q est alors donnée par $d_q(e_i, e_j) = 1 - \sum_{l=1}^{m} \sqrt{\frac{f_l^q(e_i)}{f^q(e_i)} \times \frac{f_l^q(e_j)}{f^q(e_j)}}$, où $f_l^q()$ est la fréquence de la modalité l de la question q dans le groupe, tandis que $f_q() = \sum_{l=1}^{m} f_q^l()$. Nous avons considéré 10 questions du questionnaire.

L'algorithme proposé a été exécuté pour $K = 1, \ldots, 10$. Pour chaque k, 100 exécutions ont été effectuées et le meilleur résultat selon le critère de pertinence J a été choisi. Le Tableau 1 présente les clusters obtenus pour $K = 5$, le nombre de clusters considéré comme le plus approprié selon la méthode de Da Silva (2009) pour déterminer le meilleur nombre de clusters.

Nous voyons que le regroupement reflète la similitude espérée entre les points de vue des sous-groupes de la population, par exemple, nous pouvons observer que les groupes CONS, AF/REP et EVAN (conservateurs, républicains, évangéliques) sont dans le même groupe.

Clusters
0 : DEG+, 35-44, KN/GL, LIB
1 : ALL, 45-64, <COL, WHITE, NON/WHT, MALE, FEMALE, REG, AF/IND
2 : MOD, 18-34, N/EVAN, AF/DEM
3 : CONS, EVAN, AF/REP
4 : 65, DK/GL

TAB. 1 – *Clusters obtenus ($K = 5$).*

Références

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

Burks, T. M. et K. A. Sakallah (1993). Min-max linear programming and the timing analysis of digital circuits. In *ICCAD'93*, pp. 152–155.

Da Silva, A. (2009). *Analyse de données évolutives : application aux données d'usage Web*. Thèse de doctorat, 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.

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

Lechevallier, Y., F. A. T. De Carvalho, T. Despeyroux, et F. M. De Melo (2010). Clustering of multiple dissimilarity data tables for documents categorization. In *COMPSTAT'2010 19th International Conference on Computational Statistics*, pp. 1263–1270.

Steuer, R. et E.-U. Choo (1983). An interactive weighted tchebycheff procedure for multiple objective programming. *Math. Prog. 26*, 326–344.

Wierzbicki, A. (1986). On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spektrum 8*, 73–87.

Summary

We present a new algorithm that is capable of partitioning a set of objects taking into simultaneous consideration multiple dissimilarity matrices. The algorithm uses a non-linear aggregation criterion, weighted Tchebycheff distances, more appropriate than linear combinations for constructing compromise solutions.A practical application is shown, the obtained results were pretty coherent with the data we used.

Extraction de connaissances hiérarchisées à partir d'images multirésolutions : application à la télédétection

Camille Kurtz*

*Université de Strasbourg, LSIIT, UMR CNRS 7005, Strasbourg, France
ckurtz@unistra.fr

Résumé. Différents systèmes satellitaires sont maintenant disponibles et produisent une importante masse de données utilisée pour l'observation de la Terre. Pour mieux comprendre la complexité de la surface terrestre, il devient courant d'utiliser plusieurs données provenant de capteurs différents. Cependant, la résolution spatiale de ces données n'est pas forcément équivalente, ce qui induit que le contenu sémantique de ces images peut varier. Ainsi, il est souvent difficile d'analyser automatiquement, et de manière conjointe, ces données complexes. Dans cet article nous présentons une approche permettant de tirer partie de l'aspect multirésolution de ces données au sein du processus de classification.

1 Introduction

Depuis quelques années les données issues de capteurs satellitaires deviennent de plus en plus accessibles. Différents systèmes satellitaires sont actuellement disponibles et produisent une importante masse de données utilisée pour l'observation de la Terre. Dans le domaine de la cartographie urbaine, les experts utilisent ces données pour analyser le territoire à plusieurs niveaux d'échelle dans le but d'extraire différents niveaux d'objets d'intérêt (*e.g.*, quartiers, blocs urbains, objets urbains individuels). Un moyen d'analyser automatiquement le contenu de ces images consiste à classifier ces données (d'une manière supervisée ou non) en fonction des valeurs radiométriques associées à chacun de ces pixels. Ce type d'approches est bien adapté à l'extraction d'objets d'intérêt homogènes à partir d'images à faibles résolutions spatiales.

Avec les données de dernière génération ((V)HSR - (Very) High Spatial Resolution), les objets d'intérêt apparaissent de plus en plus complexes, car ils sont souvent composés d'ensembles de pixels hétérogènes. Ainsi, dû à la grande complexité de ces données, les résultats fournis par les méthodes classiques deviennent de moins en moins intéressants (effet poivre et sel, trop grands nombres de classes, *etc.*). Pour traiter ces données, de nouvelles méthodes d'analyse (basées « objets » (Baatz et al., 2008)) ont été proposées. Le principe est d'essayer de re-construire les objets d'intérêt de la scène avant de les analyser (*i.e.*, de les classifier). Pour ce faire, l'image est découpée en segments par un processus de segmentation. Ces segments sont alors caractérisés par des attributs variés (informations spectrales, de texture ou de forme) puis sont ensuite regroupés par des algorithmes de classification.

Les méthodes objets sont bien adaptées à l'extraction d'objets simples (*e.g.*, maisons individuelles, arbres, routes) car elles supportent un certain seuil d'hétérogénéité dans ces objets

d'intérêt. En revanche, elles ne sont pas applicables directement pour extraire des objets complexes de niveaux sémantiques plus élevés (*e.g.*, blocs ou quartiers urbains, parcs) car ces derniers apparaissent comme trop hétérogènes.

Parallèlement, pour mieux comprendre la complexité de la surface terrestre, il devient courant d'utiliser des données provenant de capteurs différents (Forestier et al., 2008). L'expert dispose fréquemment d'ensembles d'images multirésolutions offrant des vues différentes de la scène (*i.e.*, à plusieures échelles) pouvant faciliter l'extraction des objets d'intérêt complexes.

Dans ces travaux, nous présentons une approche permettant de tirer partie de l'aspect multirésolution de ces données au sein du processus de classification. Basée sur la segmentation et sur la classification non supervisée des données, cette approche permet d'extraire différents niveaux de connaissances organisées hiérarchiquement. L'originalité de cette approche réside dans le fait d'extraire ces connaissances d'une manière descendante à travers la résolution : les connaissances les plus grossières (relatives aux objets les plus complexes) sont extraites à partir des images aux plus faibles résolutions spatiales puis sont progressivement affinées *via* les résolutions les plus fines.

2 Méthodologie

La méthodologie proposée permet de segmenter n images d'une même scène à différentes résolutions spatiales, en partant de la plus faible pour aller jusqu'à la plus élevée, permettant ainsi différents niveaux d'interprétation (Figure 1). Dans le cas standard, trois images sont considérées ($n = 3$) : une image MSR (30–5 m), une image HSR (3–1 m) et une image VHSR ($<$ à 1 m). Cette méthodologie de segmentation fonctionne en n étapes (une par résolution) produisant n segmentations. Chaque étape est composée (1) d'une sous-étape de segmentation basée sur des exemples et (2) d'une sous-étape de classification multirésolution.

À chaque étape r, le résultat de l'étape précédente $r - 1$ (un ensemble de régions regroupées en c clusters) est projeté dans la résolution courante et est traité comme entrée de la méthode. Le principe est alors de décomposer (c'est-à-dire de segmenter) ces c familles sémantiques dans la résolution courante afin de les affiner. Pour ce faire, à chaque étape r, l'approche de segmentation basée sur des exemples est appliquée c fois afin de partitionner l'ensemble des c familles sémantiques fournies par l'étape $r - 1$. Toutes les régions composant une même famille sémantique (*i.e.*, toutes les régions contenues dans un même cluster) sont ainsi segmentées d'une manière similaire. Une fois que toutes les régions, composant ces c familles sémantiques, ont été décomposées, il est possible de créer une partition globale de l'image en regroupant toutes les régions résultantes de cette étape. Ces régions sont ensuite classifiées par une classification non supervisée (*via* une approche multirésolution) en c ensembles homogènes de régions de même sémantique. Ce résultat est alors projeté dans une image de résolution plus fine pour que la méthodologie y soit ré-appliquée dans le but d'affiner ces régions et d'obtenir un niveau d'analyse encore plus fin. Les deux approches principales composant cette méthodologie ((1) l'approche de segmentation basée sur des exemples et (2) l'approche de classification multirésolution) sont détaillées ci-dessous.

Une approche de segmentation basée sur des exemples Cette approche de segmentation prend en entrée $k \geq 2$ imagettes (représentant k zones différentes mais de même sémantique) et retourne k segmentations. Pour l'une de ces k imagettes, un Binary Partition Tree (BPT,

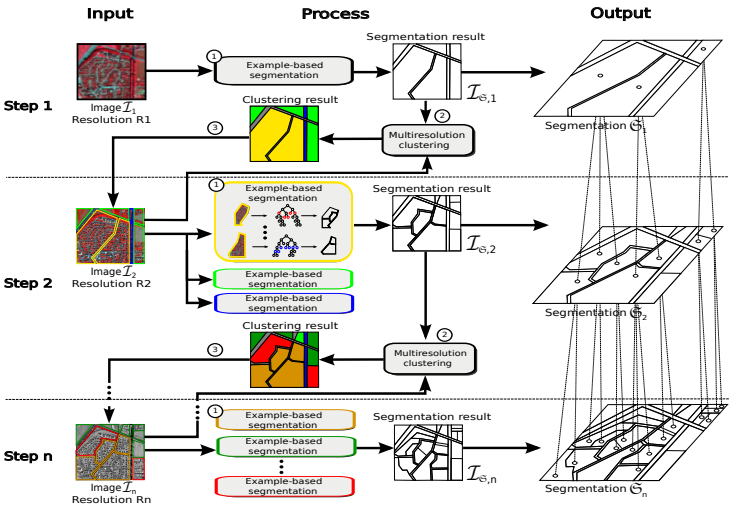


FIG. 1 – *Méthodologie d'extraction d'objets complexes à partir d'images multirésolutions.*

(Salembier et Garrido, 2000)) est construit puis l'utilisateur définit interactivement une coupe à travers cet arbre (produisant ainsi une segmentation). Cette segmentation peut alors être utilisée comme exemple pour segmenter les $k-1$ imagettes restantes. Ainsi, pour chacune de ces $k-1$ imagettes, un BPT est construit, puis une coupe est automatiquement réalisée au sein de cet arbre « mimant » la coupe exemple réalisée précédemment par l'utilisateur.

Il est alors possible d'appliquer plusieurs fois cette approche, dans une même image, pour extraire différents type d'objets ayant des sémantiques et des niveaux d'échelle différents. Pour plus de détails, les lecteurs pourront se référer à (Kurtz et al., 2011).

Une approche de classification multirésolution Nous avons ici utilisé l'approche de classification multirésolution introduite dans (Kurtz et al., 2010). La méthodologie consiste à classifier (par le biais d'un algorithme de classification non-supervisée) les régions extraites à la plus faible résolution en fonction de leurs compositions en terme de clusters formés dans l'image à haute résolution. Pour ce faire, les régions de l'image à faible résolution sont caractérisées *via* un histogramme de composition, en terme de clusters « radiométriques » formés dans l'image à haute résolution. L'algorithme K-MEANS est ensuite appliqué pour regrouper (en c clusters homogènes) les régions extraites à la résolution r se décomposant de manière similaire à la résolution $r+1$.

3 Expérimentations

La méthode a été appliquée sur trois jeux de données multirésolutions (chaque jeu de données étant composé de trois images - MSR, HSR, VHSR) afin d'y extraire des hiérarchies d'objets complexes à trois niveaux (quartiers, blocs urbains, objets urbains). Le nombre de clusters à extraire correspond ici au nombre de classes thématiques attendues par l'expert ($c = 8$ pour

les quartiers, $c = 10$ pour les blocs et $c = 15$ pour les objets urbains). Les résultats ont ensuite été comparés quantitativement à des cartes de vérité terrain (produites par l'expert) à différentes échelles. Nous avons choisi d'évaluer nos résultats en utilisant l'indice Kappa qui permet d'estimer le pourcentage de pixels correctement classifiés par rapport à une carte de vérité terrain. Les résultats obtenus se sont montrés encourageants (en moyenne, environ 75% des pixels sont correctement classifiés). Pour plus de détails, se référer à (Kurtz et al., 2011).

4 Conclusion et perspectives

La méthodologie proposée permet d'extraire, d'une manière descendante et sans information *a priori*, des hiérarchies d'objets complexes et hétérogènes à partir d'ensembles d'images multirésolutions. L'approche de segmentation basée sur des exemples permet d'adapter « localement » le processus de segmentation, ceci afin d'extraire différents niveaux d'agrégats d'objets au sein d'une même image. De plus, le fait d'extraire d'une manière descendante ces objets, permet de s'abstraire des problèmes liés à la complexité des données.

La méthodologie a été appliquée à l'extraction d'objets urbains complexes ; nous envisageons par la suite de l'utiliser pour caractériser des phénomènes naturels complexes comme les glissements de terrain.

Références

Baatz, M., C. Hoffmann, et G. Willhauck (2008). Progressing from object-based to object-oriented image analysis. In T. Blaschke, S. Lang, et G. Hay (Eds.), *Object-Based Image Analysis*, LNGC, Chapter 1, pp. 29–42. Springer.

Forestier, G., C. Wemmert, et P. Gançarski (2008). Multi-source images analysis using collaborative clustering. *EURASIP Journal on Advances in Signal Processing 2008*, 1–11.

Kurtz, C., N. Passat, P. Gançarski, et A. Puissant (2010). Multiresolution region-based clustering for urban analysis. *International Journal of Remote Sensing 31*(22), 5941–5973.

Kurtz, C., N. Passat, A. Puissant, et P. Gançarski (2011). Hierarchical segmentation of multiresolution remote sensing images. In *Proceedings of the International Symposium on Mathematical Morphology*, Volume 6671 of *LNCS*, pp. 343–354. Springer.

Salembier, P. et L. Garrido (2000). Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing 9*(4), 561–576.

Summary

Different satellite systems produce an important mass of heterogeneous data used for Earth observation. To better understand the complexity of the Earth's surface, it becomes more common to use data from different sensors. However, the spatial resolutions of these data are not necessarily equivalent, which indicates that the semantic contents of the images may differ. Thus, it is often difficult to automatically analyze, in a joint fashion, these complex data. We present an approach to take advantage of the multiresolution aspect of data.

Identification des divisions logiques de fichiers logs

Hassan Saneifar*,** Stéphane Bonniol ** Pascal Poncelet*, Mathieu Roche*

*LIRMM, CNRS, Université Montpellier 2; **Satin Technologies

1 Introduction

Plusieurs domaines d'application comme la Recherche d'Information (RI) ou la traduction automatique utilisent des méthodes de segmentation de textes. La segmentation de texte correspond au découpage d'un texte en unités plus petites. Il existe trois catégories principales de méthodes de segmentation : thématique, fenêtre, et discours. La segmentation thématique consiste à identifier différents thèmes véhiculés par le texte, pour le segmenter en des blocs thématiques (Tarek, 2003). Dans la deuxième catégorie, la segmentation s'effectue selon des fenêtres (des lignes ou des phrases) de taille fixe ou variable. Avec l'approche appelée "passage de discours", la segmentation s'effectue sur la base de la structure logique (unités de discours) de documents comme les paragraphes ou les sections (Kaszkiel et Zobel, 2001).

Le choix du type de segmentation dépend des objectifs et du domaine d'application. Nous traitons ici des données textuelles complexes, en particulier des fichiers logs. Ces données, issues du monde industriel, représentant une source principale d'information sont utilisés dans les systèmes RI (Saneifar et al., 2009). Les caractéristiques de ces données, tel que l'aspect multi-sources, multi-vocabulaire et multi-structures les différencient de documents classiques écrits en langue naturel. Considérant ces caractéristiques, nous nous intéressons aux méthodes de "passages de discours" qui identifient les unités logiques comme des segments. Il existe des solutions afin d'identifier les unités logiques classiques telles que les paragraphes. Nous pouvons également exploiter les éléments marquant les unités logiques, comme les lignes blanches ou les alinéas. Ces éléments sont appelés les divisions logiques. Or, dans les fichiers logs, n'existant pas de notions telles que paragraphe ou section, les unités logiques classiques sont difficile à identifier. Pourtant il existe, par exemple, des structures comme des tableaux, des blocs de données et des chaînes de caractères particulières marquant le début de nouvelles informations. Ces structures textuelles, plus complexes que des unités logiques classiques, sont utilisées afin de regrouper des idées et des informations. Ainsi, nous les considérons comme des unités logiques des fichiers logs.

Notre objectif est donc de proposer une approche de reconnaissance des Divisions Logiques (DL) dans les données textuelles complexes. Ainsi, nous cherchons à mettre en place un classifieur qui associe les lignes d'un corpus non expertisé à une classe : classe des lignes représentant une DL (positive) et classe des lignes non associées à une DL (négative). Contrairement aux tâches de classification de textes qui sont fondées sur les contenus (voir (Pessiot et al., 2004)), la reconnaissance d'une DL doit s'effectuer selon la structure et la mise en page des documents. Ainsi, le challenge est de définir un ensemble de descripteurs (features) pertinents qui caractérisent les unités logiques des documents, ici les fichiers logs. Bien que dans

cet article nous nous appuyons sur des données de type "fichier logs", notre approche est applicable sur tous les types de données textuelles qui ont des unités logiques complexes.

La section 2 présente notre approche de reconnaissance des DL fondée sur un système d'apprentissage supervisée. Nous développons dans les sections 3 la méthode automatique d'acquisition des descripteurs. La section 4 est consacrée aux expérimentations.

2 Classification pour la reconnaissance des divisions logiques

Dans les textes classiques, nous pouvons, par exemple, considérer un "*alinéa*" comme une des caractéristiques permettant d'identifier l'unité logique de paragraphe. Afin de pouvoir modéliser les unités logiques complexes, nous identifions leurs *caractéristiques syntaxiques*. Celles-ci permettent de distinguer le début d'une unité logique (c.-à-d. une DL) des autres lignes dans les documents. Nous appelons ces caractéristiques syntaxiques les *descripteurs*. Dans ce but, nous avons opté pour une méthode automatique d'acquisition des descripteurs dans un corpus expertisé. Nous développons cette méthode dans la section 3.
Une fois l'ensemble des descripteurs déterminé, nous considérons chaque ligne du corpus expertisé comme une instance positive ou négative. Chaque ligne est représentée sous forme d'un vecteur booléen dont chaque élément identifie la présence ou l'absence d'un des descripteurs autour de la ligne. Ainsi, nous obtenons un jeu de données d'apprentissage en fonction des descripteurs. Ensuite, un processus d'apprentissage automatique supervisé fondé sur une méthode de classification peut être appliqué. Cela permet de créer un modèle de classification des lignes de corpus en deux classes (début de segment / non début de segment). Ce modèle de classification sera ultérieurement utilisé afin d'associer les lignes d'un nouveau corpus non expertisé à une des classes positive ou négative.

3 Acquisition des descripteurs

Nous cherchons à identifier des patrons syntaxiques qui différencient une ligne marquant le début d'une unité logique des autres lignes. La figure 1 représente un exemple de deux unités logiques dans les fichiers logs. Les lignes surlignées représentent le début de deux unités. De manière simple, nous pouvons caractériser le début de cette unité logique par un patron tel que "<- - -><string><fin :>" qui signifie *une ligne commençant par une série de "-", suivi par une chaîne de caractères fini par un " :"*. Nous considérons ce patron comme un *descripteur*.

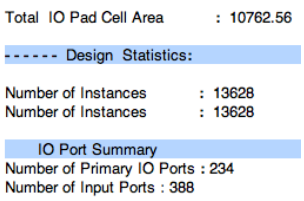


FIG. 1 – *Deux unités logiques dans un fichier logs.*

Pourtant, nous avons besoin d'un ensemble de descripteurs car un seul ne suffit pas pour caractériser pertinemment une DL. Nous cherchons également à concevoir une méthode *automatique* d'acquisition des descripteurs. Ainsi, nous avons décidé d'utiliser les n-grammes pour

caractériser le début des unités logiques. Dans le domaine du TAL (Traitement Automatique du Langue), un n-grammes correspond à une série d'items dans un texte où les items peuvent être des lettres ou des mots. Les n-grammes sont souvent utilisés comme des descripteurs dans des tâches de classification de documents textuels (Tan et al., 2002). Les n-grammes permettent de modéliser le *contenu* ainsi que *l'enchainement des mots* dans un document. Par exemple, en extrayant les tri-grammes (de lettres) dans la première ligne surlignée de la figure 1, nous obtenons "`---`", "`---`", "`␣De`", "`sig`", "`n␣S`", "`tat`", "`ist`", "`ics`", "`:`". *Or*, dans notre contexte, nous ne nous intéressons qu'à la *structure* des documents. Cela signifie que nous ne cherchons pas à identifier les unités logiques selon leur contenu (les mots ou les lettres) *mais* selon leurs structures textuelles (*les ponctuations, les symboles, les mises en page, etc.*). Cette nécessité nous a conduit à définir et proposer un *nouveau type original* de grammes que nous appelons *vs-grammes généralisés*. Ainsi, un vs-grammes est une série de caractères alphanumériques et non-alphanumériques défini de la manière suivante :

- si le gramme contient une série de caractères alphanumériques, il se termine par un caractère non-alphanumérique [1]. Le gramme suivant commence par le caractère non-alphanumérique.
- si le gramme commence par une série de caractères non-alphanumérique, il se termine par un caractère alphanumérique. Le gramme suivant commence par le caractère alphanumérique.

En prenant l'exemple précédent (figure 1), nous obtenons les vs-grammes suivants sur le premier segment : "`------ D`", "`Design Statistics :`". Contrairement aux tri-grammes précédemment extraits, les vs-grammes modélisent bien la composition des caractères dans cette ligne. Autrement dit, ces vs-grammes expriment une composition de caractères telle qu'une chaîne de "-" suivie par une lettre et une chaîne de lettres terminant par un " **:**", ce qui marque le début d'une unité logique dans les fichiers logs. Les vs-grammes sont toujours sensibles au contenu des textes. Par exemple, ici, le deuxième vs-gramme extrait, présente une série de lettres *composée* des mots "`Design`" et "`Statistics`". Or, la connaissance essentielle à prendre en compte est la présence d'une chaîne de lettres. De la même manière, le nombre de "-" dans l'autre vs-grammes n'est pas informatif. C'est la raison pour laquelle nous généralisons les vs-grammes en remplaçant les suites de lettres et de symboles par leurs équivalents en expression régulière. Ainsi, sur cet exemple, nous obtenons les vs-grammes généralisés suivants : "\-+ \w+", "\w+ **:**". Nous constituons l'ensemble des descripteurs en extrayant des vs-grammes généralisés dans une fenêtre de lignes autour des lignes marquant le début des unités logiques. Ainsi, en prenant le premier segment de la figure 1, nous obtenons les descripteurs suivants : D_1(\-+ \w+, 0), D_2(\w+ **:**, 0), D_3(\w+ \s+ **:**, -2) et D_4(**:** \w+, -2). Pour chaque descripteur, le chiffre après le patron représente le numéro de ligne dans la fenêtre. Le zéro correspond à la ligne même du début de segment. Nous procédons de la même manière pour les autres lignes marquant les DL afin de créer l'ensemble des descripteurs.

4 Expérimentations

Nous avons évalué la performance de notre approche en terme de précision et de rappel du modèle de classification. Le corpus d"apprentissage est constitué de 19 fichiers logs différents issus du monde industriel. Les fichiers logs contiennent des données réelles et sont différents en terme de contenu et surtout de structure. Le corpus d'apprentissage est de taille 1.1 Mo et contient au total 19638 lignes. Nous présentons ici les résultats obtenus en utilisant les algorithmes de classification qui ont donné les meilleurs résultats : l'arbre de décision C4.5 et

1. Les espaces s'ajoutent systématiquement aux grammes.

KPPV. Pour appliquer les algorithmes, nous utilisons les implémentations intégrées au logiciel WEKA. Afin d'évaluer la performance de classification, nous utilisons la méthode de validation croisée (10 niveaux). Le tableau 1 présente la performance des classifieurs. Avec les

Classe	Précision	Rappel	F-Score
Pos	0.92	0.74	**0.82**
Neg	0.96	0.98	**0.97**

Classe	Précision	Rappel	F-Score
Pos	0.94	0.75	**0.84**
Neg	0.97	0.98	**0.97**

TAB. 1 – *Performance de classification - C4.5 (Gauche) et KPPV (Droite)*

KPPV, nous obtenons une précision égale à 0.94 dans la classe positive et égale à 0.97 dans la classe négative. Selon les résultats obtenus, nous argumentons que les vs-grammes généralisés représentent bien les caractéristiques syntaxiques des unités logiques d'un document.

5 Conclusions

Nous avons présenté une approche de segmentation des fichiers logs (les textes ayant des unités logiques complexes) fondée sur l'apprentissage supervisée. Dans notre approche nous avons constitué un ensemble de descripteurs où chacun présente une des caractéristiques syntaxiques des unités logiques. Afin de créer l'ensemble des descripteurs, nous avons proposé une méthode d'acquisition automatique des descripteurs qui utilise les vs-grammes généralisés présentés dans cet article. Nous avons réussi à reconnaître les unités logiques avec un F-Score égal à 0.84.

Références

Kaszkiel, M. et J. Zobel (2001). Effective ranking with arbitrary passages. *J. Am. Soc. Inf. Sci. Technol. 52*, 344–364.

Pessiot, J.-F., M. Caillet, M.-R. Amini, et P. Gallinari (2004). Apprentissage non-supervisé pour la segmentation automatique de textes. In *CORIA*, pp. 213–228.

Saneifar, H., S. Bonniol, A. Laurent, P. Poncelet, et M. Roche (2009). Terminology extraction from log files. In *DEXA'09*, Lecture Notes in Computer Science, pp. 769–776. Springer.

Tan, C.-M., Y.-F. Wang, et C.-D. Lee (2002). The use of bigrams to enhance text categorization. *Inf. Process. Manage. 38*, 529–546.

Tarek, O. (2003). La segmentation des documents techniques en amont de l'indexation : définition d'un modèle. *Revue d'Information Scientifique et Technique (RIST) vol. 13*(no1), 79–94.

Summary

In the segmentation method called discourse passages, the recognition of logical divisions in documents is essential. It is more difficult in the documents with logical units different from those found in classic texts such as paragraph or section. This is due to the fact that such classic logical units are not significant in some specialized documents such as log files that are studied in this article. Thus, we propose an automatic method to characterize the complex logical units found in this type of document according to their syntactic characteristics. Then, a supervised learning process is in place in order to recognize the logical units according to their characteristics. Experimental results on the recognition of complex logical units in the log files from the industrial world are encouraging.

Approche symbolique pour l'extraction de thématiques: application à un corpus issu d'appels téléphoniques

Raja Haddad*,**, Filipe Afonso*, Edwin Diday ***

*Syrokko, Aéropôle Roissy CDG, 95731 Roissy
(haddad, afonso)@syrokko.com
**LAMSADE, ***CEREMADE, Université de Paris 9 Dauphine,75775 Paris
diday@ceremade.dauphine.fr

Résumé. Nous présentons une stratégie d'extraction automatique de thématiques, à partir d'un corpus de documents textuels, basée sur les méthodes d'analyse des données symboliques. L'originalité de la méthode est qu'elle découvre les thèmes importants avec des outils de classification et de visualisation adaptés aux données symboliques, en partant des données brutes sans utiliser des règles lexicales. Cette stratégie est appliquée à un corpus de documents issus d'appels téléphoniques reçus par le service clientèle d'une grande société.

1 Introduction

L'extraction de thématiques à partir des données issues de conversations téléphoniques est une tâche délicate surtout du fait du nombre important de mots qui peuvent apparaitre dans le corpus. Afin d'exploiter ces données, nous avons mis en œuvre une stratégie qui tire profit des méthodes d'Analyse de Données Symboliques (ADS) (Diday, 2008). Ces méthodes offrent une alternative puissante aux méthodes classiques en décrivant les unités statistiques par des données dites symboliques conservant la variation interne des individus qui les composent.

Cet article s'organise en quatre sections. La première offrira une brève présentation de l'état de l'art de l'extraction des thématiques à partir d'un corpus issu de la retranscription d'appels téléphoniques. La deuxième section présentera la stratégie symbolique mise en œuvre. Les résultats de l'application de la stratégie sont présentés dans la troisième section. Enfin, la quatrième section comportera la conclusion et quelques perspectives.

2 Extraction de thématiques à partir d'appels téléphoniques

L'extraction d'information à partir d'un corpus issu de la retranscription de conversations téléphoniques est un domaine d'application important des études de *text-mining*. Il existe plusieurs travaux qui ont été élaborés pour résoudre ce type de problèmes. La plupart de ces travaux propose une combinaison entre l'étude lexicale et celle statistique du corpus afin d'obtenir un résultat satisfaisant. Parmi ces travaux, nous pouvons citer (Bozzi et al., 2009) et (Cailliau et Poudat, 2008).

Est-il possible de trouver les thématiques, contenues dans ce type de corpus, sans utiliser l'analyse lexicale ? En utilisant les méthodes et outils d'ADS nous avons mis en œuvre une stratégie d'extraction de thématiques basée uniquement sur l'étude statistique d'un corpus de conversations téléphoniques permettant de répondre par l'affirmative à cette question.

3 Stratégie basée sur l'ADS

Notre stratégie est basée sur l'utilisation d'une méthodologie symbolique appliquée directement aux données textuelles brutes. Ainsi, aucun traitement sémantique préalable n'est appliqué aux données. Les principales étapes de notre stratégie sont :

- La construction des données symboliques à partir des données initiales représentées par un tableau qui associe les documents à leurs mots. Cette étape est faite à travers :
  - La description des "documents" (considérés comme les concepts au sens de l'ADS) par la distribution de leurs mots pondérés par le TF*IDF [1].
  - La classification des documents en utilisant le module "ClustSyr" du logiciel SYR [2]. Ce module implémente l'extension de l'algorithme des nuées dynamiques recouvrantes, de type OKM (Cleuziou, 2008), aux données symboliques.
  - La description des mots par leurs distributions dans les classes de documents pondérées par le TF*IDF et leur classification en utilisant "ClustSyr".
- La sélection des mots caractéristiques de chaque classe de mots en utilisant le module "HistSyr" du logiciel SYR. Ce module implémente une méthode qui permet de sélectionner les k modalités les plus discriminantes des concepts. Cette sélection est faite en cherchant pour chaque concept les modalités qui le différencient par rapport aux autres.
- L'épurage visuel des classes de mots qui consiste en la suppression des classes de mots jugées non pertinantes par l'analyste. Cette étape est réalisée grâce à l'utilisation du module "NetSyr" du logiciel SYR. Ce module implémente l'extention aux données symboliques de l'analyse en composantes principales ; il permet aussi de représenter des réseaux liant les classes de mots et de visualiser des partitions et des recouvrements.
- La labellisation des classes de mots retenues pour obtenir les thèmes. Cette étape est faite par usage de NETSYR en donnant, manuellement de façon interactive, un label illustratif à chaque classe de mots en se basant sur le sens de ces mots.
- La description des classes de documents par les thèmes induits des classes de mots.
- L'épurage visuel et la labélisation des classes de documents grâce à NetSyr.

4 Application sur les données réelles

En partant du fichier initial, contenant deux colonnes où l'une représente l'identifiant du document et l'autre contient un mot, nous construisons les fichiers de données symboliques que nous allons analyser. La figure 1 représente les étapes de construction des classes de documents et des classes de mots (voir section 3). Afin de réduire le nombre de mots représentatifs des

1. *Term Frequency-Inverse Document Frequency* : c'est une méthode permettant d'évaluer l'importance d'un terme contenu dans un corpus. Sa valeur est en fonction de la fréquence du mot dans le document et dans le corpus.
2. SYR : c'est un logiciel implémenté par la société Syrokko. Il offre un ensemble de modules qui facilitent l'ADS.

différentes classes de mots, nous avons utilisé "HistSyr". Après cette étape, les classes de mots sont décrites par 15 mots au maximum.

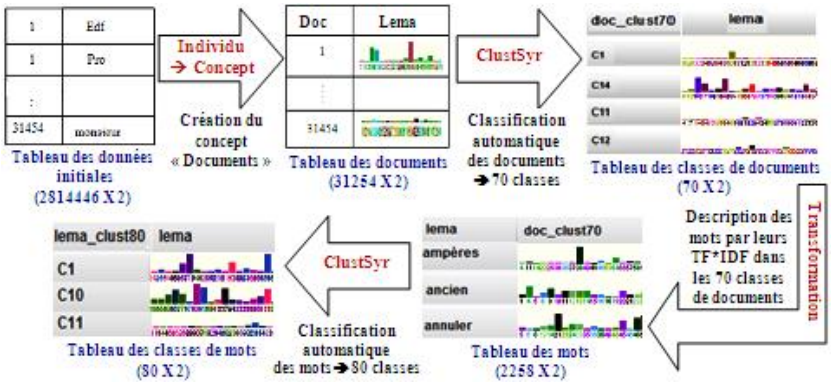


FIG. 1 – *Méthodologie de création des tableaux de données symboliques de l'étude.*

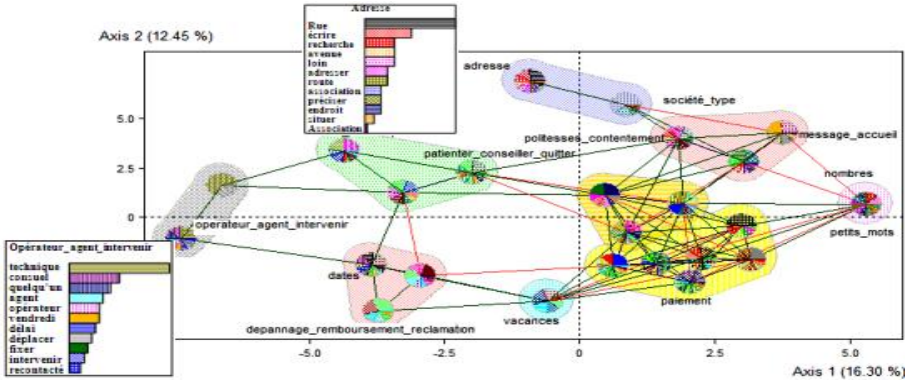


FIG. 2 – *Projection des thèmes dans NetSyr et visualisation de leurs descriptions.*

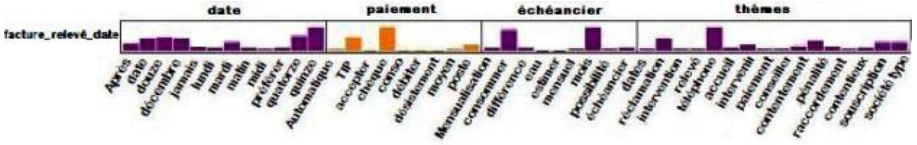


FIG. 3 – *Exemple de description de la classe de documents "facture_ relevé_ date".*

NetSyr a été ensuite utilisé pour l'épurage visuelle et la labellisation des classes de mots en thèmes. Finalement, 25 thèmes ont été conservés, comme : Politesse_ contentement, vacances, relance_ contencieux, etc. Dans NetSyr, les thèmes peuvent être positionnés les uns par rapport aux autres et l'affichage d'un réseau permet de visualiser la proximité entre eux grâce à des dissimilarités pouvant être calculées sur plusieurs axes au choix. Un clic sur chaque thème permet d'avoir sa description détaillée sous forme d'histogramme (voir figure 2).

En se basant sur les tableaux décrivant les classes de documents par leurs mots et les thèmes par leurs mots, nous avons construit un tableau représentant les classes de documents décrits par les thèmes. Ensuite nous avons appliqué la méthode "HistSyr" pour sélectionner les thèmes les plus discriminants de chaque classe de documents. Enfin en utilisant NetSyr pour l'épurage visuel nous avons obtenu 19 classes de documents à partir des 70. Ces classes de documents sont décrites par leurs thèmes qui sont eux-mêmes décrits à un niveau inférieur par la distribution de leurs mots (voir figure 3).

5 Conclusion

L'ADS ne prétend pas proposer sa propre stratégie indépendante de text-mining puisqu'elle ne considère pas les aspects sémantiques qui sont une part considérable et indispensable d'une telle étude. Elle peut cependant offrir une aide précieuse grâce aux possibilités offertes par ses principes et ses outils dans l'étude logique de différents concepts sur plusieurs niveaux.

Une perspective immédiate aux travaux qui ont été présentés s'attachera au développement d'un algorithme permettant de faire automatiquement la classification et la sélection des documents, des mots et des thèmes d'intérêt. Un tel algorithme constituerait une méthode de co-clustering adaptée au traitement des données symboliques, dont les applications potentielles dépasseraient le cadre des données textuelles.

Références

Bozzi, L., P. Suignard, et C. Waast-Richard (2009). Segmentation et classification non supervisée de conversations téléphoniques automatiquement retranscrites. In *Actes de la conférence TALN'09 - Session Posters*, Senlis, France.

Cailliau, F. et C. Poudat (2008). Caractérisation lexicale des contributions clients agents dans un corpus de conversations téléphoniques retranscrites. In *Actes des Journées internationales d'Analyse statistique des Données Textuelles (JADT 2008)*, Lyon, France., pp. 267–275. Presses universitaires de Lyon. ISBN :978-2-7297-0810-8.

Cleuziou, G. (2008). An extended version of the k-means method for overlapping clustering. In *19th International Conference on Pattern Recognition (ICPR'2008)*, pp. 1–4.

Diday, E. (2008). The state of the art in symbolic data analysis : overview and future. *Chapter 1, in E.Diday, M. Noirhomme-Fraiture (2008) editors "Symbolic Data Analysis and the SODAS Software". 457 Pages. Wiley. ISBN 87-0-470-01883-5*.

Summary

We present a strategy, based on symbolic data analysis methods, to automatically extract thematic from a corpus of text documents. The originality of this method is the discovering of the important themes using classification and visualization tools adapted to symbolic data, without applying lexical rules to the initial data. This strategy is applied to a corpus of documents obtained by the transcription of conversations between customers and agents of a French company.

Une version batch de l'algorithme SOM pour des données de type intervalle

F.A.T. De Carvalho*, L.D.S. Pacifico*

*Centro de Informatica - CIn/UFPE, Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária, 50.740-560, Recife - PE, Brésil
{fatc,ldsp}@cin.ufpe.br

Résumé. Ce travail présente une version batch de l'algorithme SOM pour des données de type intervalle avec pondération automatique des variables. Des applications sur des tableaux de données de type intervalle montrent l'intérêt de cet algorithme.

1 Introduction

Dans un tableau de données chaque cellule contient soit une valeur numérique (correspondant à une variable quantitative) soit une catégorie (ordonnée ou non) correspondant à une variable qualitative. L'analyse des données symboliques (Bock et Diday, 2000) a introduit des nouvelles variables dites "symboliques" qui permettent de tenir compte de la variabilité et/ou de l'incertitude présente dans les données. Par conséquent, dans un tableau de données symboliques une cellule peut contenir un intervalle, un ensemble de catégories ou encore une distribution de poids (fréquences).

Les variables de type intervalle sont souvent rencontrées dans la pratique : un intervalle peut décrire la plus petite et la plus grande valeur d'une mesure concernant un individu pendant une journée ou encore l'étendue des salaires dans une entreprise. Il peut aussi indiquer que la valeur exacte d'une mesure ne peut pas être obtenue, mais que cette valeur est dans cet intervalle.

Les cartes de Kohonen (Kohonen, 1994), nommées SOM, sont une des méthodes de classification non supervisées les plus utilisées car elles réalisent en même temps un partitionement des objets et réduit la dimensionalité de l'espace de représentation de ces objets. De plus elles ont la capacité de produire des représentations structurées des classes obtenues en réalisant un positionnement spatial des prototypes de ces classes.

Nous nous intéressons à l'extension des cartes auto-organisatrices de Kohonen aux données de type intervalle. Une extensions de la version stochastique des cartes SOM pour données de type intervalle a été proposé par Bock (2003). Nous proposons ici une extension de la version batch des cartes auto-organisatrices de Kohonen introduite par Badran et al. (2005) pour des données de type intervalle avec pondération automatique des variables.

Pour montrer l'intérêt de cet algorithme, nous l'appliquons à trois tableaux de données de type intervalle décrivant, respectivement, des espèces de poisson, des modèles de voitures et quelques villes de la planète.

2 Une version batch de l'algorithme SOM pour des données de type intervalle

Soit $E = \{e_1, \ldots, e_n\}$ l'ensemble des individus, chaque individu étant représenté par un vecteur d'intervalles $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})\,(i = 1, \ldots, n)$, où $x_{ij} = [a_{ij}, b_{ij}] \in \Im = \{[a, b] : a, b \in \Re,\ a \leq b\}$. Chaque neurone de la carte auto-organisatrice est représenté par un prototype décrit par un vecteur d'intervalles $\mathbf{w}_r = (w_{r1}, \ldots, w_{rp})\,(r = 1, \ldots, m)$, où $w_{rj} = [\alpha_{rj}, \beta_{rj}] \in \Im$.

Notre version de l'algorithme SOM pour des données de type intervalle avec pondération automatique des variables, basé sur la version batch de SOM (Badran et al., 2005), alterne iterativement trois étapes (représentation, pondération et affectation). Dans chacune de ces étapes tous les objets de E sont présentés à la carte auto-organisatrice. La fonction de coût de cet algorithme est donnée par :

$$J = \sum_{i=1}^{n} \sum_{r=1}^{m} K^T\left(\delta(f(\mathbf{x}_i), r)\right) d^2_{\boldsymbol{\lambda}_r}(\mathbf{x}_i, \mathbf{w}_r) \tag{1}$$

où

$$d^2_{\boldsymbol{\lambda}_r}(\mathbf{x}_i, \mathbf{w}_r) = \sum_{j=1}^{p} \lambda_{rj}[(a_{ij} - \alpha_{rj})^2 + (b_{ij} - \beta_{rj})^2] \tag{2}$$

est le carré d'une distance Euclidienne adaptative (Diday et Govaert, 1977; De Carvalho et Lechevallier, 2009) entre vecteurs d'intervalles parametrisé par un vecteur de pondération $\boldsymbol{\lambda}_r = (\lambda_{r1}, \ldots, \lambda_{rp})$ sur les variables. Les vecteurs de pondération $\boldsymbol{\lambda}_r\,(r = 1, \ldots, m)$ changent à chaque iteration et sont différent d'un neurone pour un autre.

f est la fonction d'identification qui est définie de E dans $\{1, , \ldots, m\}$. Cette fonction associe à chaque objet de E sa classe d'appartenance.

K^T est la fonction de voisinage des cartes auto-organisatrices, la valeur depend de la proximité topologique des deux classes et du paramètre T qui joue le rôle d'une température. La proximité topologique est souvent mesurée par une distance δ définie a priori entre deux classes.

La fonction

$$d^T_{\boldsymbol{\Lambda}}(\mathbf{x}_i, \mathbf{w}_{f(\mathbf{x}_i)}) = \sum_{r=1}^{m} K^T\left(\delta(f(\mathbf{x}_i), r)\right) d^2_{\boldsymbol{\lambda}_r}(\mathbf{x}_i, \mathbf{w}_r) \tag{3}$$

est une somme pondérée de distances entre cet objet $\mathbf{x}_i$ et l'ensemble des prototypes $\mathbf{w}_r\ \forall r = 1, \ldots, m$.

Pour T étant fixé, J est minimisée itérativement en trois étapes : représentation, pondération et affectation.

Dans **l'étape de représentation**, la fonction d'identification f et la matrice de pondération $\boldsymbol{\Lambda}$ sont fixées. La matrice de pondération $\boldsymbol{\Lambda}$ est composée des m vecteurs de pondération $\boldsymbol{\lambda}_r$ = $(\lambda_{r1}, \ldots, \lambda_{rp})$. La fonction de coût J est minimisée par rapport aux prototypes. Les composantes $w_{rj} = [\alpha_{rj}, \beta_{rj}]\,(j = 1, \ldots, p)$ du prototype $\mathbf{w}_r = (w_{r1}, \ldots, w_{rp})$ sont calculées pour chaque neurone par :

$$\alpha_{rj} = \frac{\sum_{i=1}^{n} K^T\left(\delta(f^T(\mathbf{x}_i), r)\right) a_{ij}}{\sum_{i=1}^{n} K^T[\delta(f^T(\mathbf{x}_i), r)]} \text{ et } \beta_{rj} = \frac{\sum_{i=1}^{n} K^T\left(\delta(f^T(\mathbf{x}_i), r\right) b_{ij}}{\sum_{i=1}^{n} K^T[\delta(f^T(\mathbf{x}_i), r)]} \tag{4}$$

Dans **l'étape de pondération**, les prototypes et la fonction d'identification sont fixés. La fonction de coût J est minimisée par rapport aux vecteurs de pondération $\boldsymbol{\lambda}_r = (\lambda_{r1}, \ldots, \lambda_{rp})$ pour $(r = 1, \ldots, m)$ sous les contraintes $\lambda_{rj} > 0$ et $\prod_{j=1}^{p} \lambda_{rj} = 1$. Les vecteurs de pondération sont calculés par

$$\lambda_{rj} = \frac{\{\prod_{h=1}^{p}\left(\sum_{i=1}^{n} K^T\left(\delta(f(\mathbf{x}_i), r)\right)\left[(a_{ih} - \alpha_{rh})^2 + (b_{ih} - \beta_{rh})^2\right]\right)\}^{\frac{1}{p}}}{\sum_{i=1}^{n} K^T\left(\delta(f(\mathbf{x}_i), r)\right)\left[(a_{ij} - \alpha_{rj})^2 + (b_{ij} - \beta_{rj})^2\right]} \tag{5}$$

Dans **l'étape d'affectation**, les prototypes et la matrice de pondération sont fixés. La fonction de coût J est minimisée par rapport à la fonction d'identification f et chaque individu $\mathbf{x}_i$ est afecté au neurone le plus proche :

$$r = f(\mathbf{x}_i) = arg \min_{1 \leq h \leq m} d^T_{\boldsymbol{\Lambda}}(\mathbf{x}_i, \mathbf{w}_h) \tag{6}$$

L'algorithme.

1) Initialisation
   Fixer le nombre m de neurones ou classes, la fonction de distance topologique δ, la fonction de voisinage K^T avec T_{min} et T_{max} et le nombre d'iterations N_{iter}. Faire $t \leftarrow 0$ et $T = T_{max}$;
   Selectioner au hasard m prototypes distincts $\mathbf{w}_r^{(0)} \in E\,(r = 1, \ldots, m)$;
   Faire la carte $L(m, \mathbf{W}^{(0)})$, où $\mathbf{W}^{(0)} = (\mathbf{w}_1^{(0)}, \ldots, \mathbf{w}_m^{(0)})$;
   Initialiser $\boldsymbol{\Lambda}^{(0)}$=($\boldsymbol{\lambda}_1^{(0)}$,...,$\boldsymbol{\lambda}_m^{(0)}$) avec $\boldsymbol{\lambda}_r^{(0)} = (1, \ldots, 1)\,(r = 1, \ldots, m)$;
   Affecté chaque individu $\mathbf{x}_i$ au plus proche neurone (cluster) selon l'équation (6).
2) *Étape 1 : Représentation.*
   Faire $t \leftarrow t + 1$ et $T = T_{max}(\frac{T_{min}}{T_{max}})^{\frac{t}{N_{iter}-1}}$
   Calculer les prototypes $\mathbf{w}_r^{(t)}$ $(r = 1, \ldots, m)$ selon l'équation (4)
3) *Étape 2 : Pondération.*
   Calculer les composants $\lambda_{rj}^{(t)}$ du vecteur de pondération $\boldsymbol{\lambda}_r^{(t)}$ $(r = 1, \ldots, m; j = 1, \ldots, p)$ selon l'équation (5)
4) *Étape 3 : Affectation.* Affecter chaque individu $\mathbf{x}_i\,(i = 1, \ldots, n)$ au neurone le plus proche selon l'équation (6) ;
5) *Critère d'arret.*
   Se $T = T_{min}$ alors STOP ; sinon aller vers 2 (Étape 1 : Représentation).

3 Applications

Nous mesurons la performance de cet algorithme en termes de taux global d'erreur de classification, en le comparant avec la version sans pondération (sans l'étape 2), sur trois tableaux de données symboliques (De Carvalho et Lechevallier, 2009). Le tableau *poisson* décrit 12 espèces de poisson par 13 variables de type intervalle. Ces espèces sont groupées en 4 classes a

priori ayant respectivement, 4, 4, 2 et 2 espèces. Le tableau *voiture* décrit 33 modèles de voitures par 8 variables de type intervalle. Ces modèles sont groupées en 4 classes a priori ayant, respectivement, 10, 8, 7 et 8 modèles. Enfin, le tableau *température* décrit 35 villes par 12 variables de type intervalle. Ces villes sont groupées en 2 classes a priori ayant, respectivement, 15 et 20 individus. Dans cette application les paramètres de l'algorithme ont été les suivants : $N_{Iter} = 500$; δ : distance Euclidienne ; fonction noyau $K^T(\delta(c,r)) = exp(-\frac{(\delta(c,r))^2}{2T^2})$. La structure *a priori* est une grille à deux dimensions de taille $2\times$ 3, 2×5 et 2×8, pour, respectivement, les tableaux de données *poisson*, *voiture* et *température*. L'algorithme a été exécuté 100 fois et le meilleur résultat a été choisi selon la fonction de coût J. La Table 1 montre les performances de l'algorithme SOM sans (S.P.) et avec (A.P.) pondération appliqué à ces tableaux de données pour des différentes valeurs des paramètres T_{min} et T_{max}. Les bons résultats sur ces tableaux de données montrent l'intérêt de la pondération automatique des variables.

TAB. 1 – *Taux global d'erreur de classification*

	Poisson		Voiture		Température	
$T_{min} : T_{max}$	S.P	A.P	S.P	A.P	S.P	A.P
$0.1 : 1.0$	0.416	0.416	0.333	**0.181**	0.000	0.000
$0.2 : 2.0$	0.416	**0.250**	0.212	**0.181**	0.000	0.000
$0.3 : 3.0$	0.416	**0.250**	0.242	**0.181**	0.000	0.000
$0.4 : 4.0$	0.416	**0.250**	0.303	**0.212**	0.000	0.000
$0.5 : 5.0$	0.333	0.333	0.303	0.303	0.000	0.000
$0.6 : 6.0$	0.500	**0.416**	0.333	**0.181**	0.000	0.000

Références

Badran, F., M. Yacoub, et S. Thiria (2005). Self-organizing maps and unsupervised classification. In G. Dreyfus (Ed.), *Neural Networks. Methodology and Applications*, pp. 379–442. Springer.

Bock, H.-H. (2003). Clustering algorithms and kohonen maps for symbolic data. *Journal of the Japanese Society of Computational Statistics 15*, 217–229.

Bock, H.-H. et E. Diday (2000). *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*. Berlin: Springer Verlag.

De Carvalho, F.-A.-T. et Y. Lechevallier (2009). Dynamic clustering of interval-valued data based on adaptive quadratic distances. *IEEE Transactions on Systems, Man and Cybernetics. Part A. Systems and Humans 39*, 1295–1306.

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

Kohonen, E. (1994). *Self-Organizing Maps*. Berlin: Springer Verlag.

Summary

This paper gives an adaptation of the self-organizing maps for interval-valued data with automatic weighting of the interval-valued variables. This algorithm is applied to interval-valued data sets in order to show its usefulness.

Une adaptation des cartes auto-organisatrices aux tableaux de dissimilarité multiples

Francisco de A.T. de Carvalho*, Anderson B.S. Dantas*, Yves lechevallier**

*Centro de Informatica - CIn/UFPE, Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária, 50.740-560, Recife - PE, Brésil
{fatc,absd}@cin.ufpe.br
**INRIA, Paris-Rocquencourt - 78153 Le Chesnay cedex, France
Yves.Lechevallier@inria.fr

Résumé. Cet article propose une adaptation des cartes auto-organisatrices permettant d'utiliser simultanément plusieurs tableaux de dissimilarités. Nous montrerons l'interêt de cet algorithme en l'appliquant à la classification automatique des deux jeux de données de l'UCI *http://archive.ics.uci.edu/ml*.

1 Introduction

Les cartes de Kohonen (Kohonen, 1994), nommées SOM, sont une des méthodes de classification automatique les plus utilisées car elles réalisent en même temps un partitionement des objets et réduit la dimensionalité de l'espace de représentation de ces objets. De plus elles ont la capacité de produire des représentations structurées des classes obtenues en réalisant un positionnement spatial des prototypes de ces classes.

Cet article introduit une adaptation des cartes auto-organisatrices à un ensemble d'objets décrits par plusieurs tableaux de dissimilarités. Elle est basée sur l'algorithme des cartes auto-organisatrices appliqué à un tableau unique de dissimilarité (El Golli et al., 2006) et sur l'algorithme des nuées dynamiques utilisant les distances adaptatives (Diday et Govaert, 1977; De Carvalho et Lechevallier, 2009).

L'idée générale est que chaque matrice de dissimilarités joue un rôle collaboratif (Pedrycz, 2002) dans le processus de consensus (Leclerc et Cucumel, 1987) afin d'obtenir une partition unique. L'influence de ces différentes matrices de dissimilarités n'est pas identique pour la recherche des classes de la partition finale. Cette pertinence est quantifiée par un vecteur ou une matrice de poids associé à chaque classe et à chaque matrice de dissimilarités qui est mise à jour par un apprentissage tout au long du déroulement de l'algorithme.

Pour montrer l'interêt de cet algorithme, nous l'appliquons à la classification automatique des deux jeux de données de l'UCI (Frank et Asuncion, 2010).

2 Une version batch de l'algorithme SOM sur les matrices de dissimilarités

Soient $E = \{e_1, \ldots, e_n\}$ un ensemble de n objets ou exemples et p matrices de dissimilarités $\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_p$ définies sur E. Dans notre approche nous supposons que tous les prototype g_l des classes C_l appartiennent à l'ensemble des objets E, i.e., $g_l \in E \, \forall l = 1, \ldots, c$, où c est le nombre de classes fixé a priori.

Notre algorithme de partitionnement, basé sur la version batch de SOM, alterne iterativement trois étapes (représentation, pondération et affectation). Durant chacune de ces étape tous les objets de E sont présentés à la carte auto-organisatrice. La fonction de coût optimisée par notre algorithme est donnée par :

$$J = \sum_{i=1}^{n} \sum_{l=1}^{c} K^T[\delta(f(e_i), l)] D_{\boldsymbol{\lambda}_l}(e_i, g_l) = \sum_{i=1}^{n} \sum_{l=1}^{c} K^T[\delta(f(e_i), l)] \sum_{j=1}^{p} \lambda_{lj} d_j^2(e_i, g_l) \quad (1)$$

où

- $D_{\boldsymbol{\lambda}_l}(e_i, g_l) = \sum_{j=1}^{p} \lambda_{lj} d_j^2(e_i, g_l$ est la dissimilarité entre un objet e_i de E et le prototype $g_l \in E$ de la classe C_l paramétrisé par le vecteur de pondération $\boldsymbol{\lambda}_l = (\lambda_{l1}, \ldots, \lambda_{lp})$,
- λ_{lj} est la pondération associée à la matrice de dissimilarité $\mathbf{D}_j$ pour la classe C_l,
- $d_j(e_i, g_l)$ est la mesure de dissimilarité locale d_j liée à la matrice $\mathbf{D}_j$ entre un exemple e_i et le prototype de la classe $g_l \in E$.
- f est la fonction d'identification qui est définie de E dans $\{1, , \ldots, c\}$. Cette fonction associe à chaque objet de E sa classe d'appartenance.
- $\delta(k, l)$ est la proximité topologique sur la grille entre la classe C_k et la classe C_l.
- $T = T_{max}(\frac{T_{min}}{T_{max}})^{\frac{t}{N_{iter}}}$ est une fonction décroissante du nombre d'itérations t déja réalisées.
- K^T est la fonction de voisinage des cartes auto-organisatrices, la valeur depend de la proximité topologique δ entre les deux classes et du nombre T.

Si la fonction de voisinage $K^T(a, b)$ est égale à 1 si $a = b$ et 0 sinon alors la fonction de coût est égale à la fonction de coût de la méthode *K-Medoid* (Kaufman et Rousseeuw, 1990). Dans les autres cas la fonction $d_{(\mathbf{\Lambda},\, \mathrm{T})}$

$$d^2_{(\mathbf{\Lambda},\, \mathrm{T})}(e_i, g_{f(e_i)}) = \sum_{l=1}^{c} K^T[\delta(f(e_i), l)] D_{\boldsymbol{\lambda}_l}(e_i, g_l) = \sum_{j=1}^{p} \sum_{l=1}^{c} K^T[\delta(f(e_i), l)] \lambda_{lj} d_j^2(e_i, g_l) \quad (2)$$

est une somme pondérée de distances entre cet objet e_i et l'ensemble des prototypes $g_l \in E \, \forall l = 1, \ldots, c$ sur l'ensemble des matrices de dissimilarités $\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_p$ définies sur E.

Pour $T =$ étant fixé, la fonction de coût J est minimisée itérativement en fonction des trois étapes suivantes : représentation, pondération et affectation.

Dans **l'étape de représentation**, la fonction d'identification f et la matrice de pondération $\mathbf{\Lambda}$ sont fixées. La matrice de pondération $\mathbf{\Lambda}$ est composée des c vecteurs de pondération $\boldsymbol{\lambda}_l = (\lambda_{l1}, \ldots, \lambda_{lp})$.

La fonction de côut J est minimisée par rapport aux prototypes. Le prototype g_l de la classe C_l est l'objet $e_m \in E$ qui minimise le critère J, d'où :

$$m = \arg \min_{1 \leq h \leq n} \sum_{i=1}^{n} K^T[\delta(f(e_i), l)] \sum_{j=1}^{p} \lambda_{lj} d_j^2(e_i, e_h) \tag{3}$$

Dans **l'étape de pondération**, les prototypes $g_l \in E \, \forall l = 1, \ldots, c$ et la fonction d'identification f sont fixés.

La fonction de coût J est minimisée par rapport aux vecteurs de pondération $\boldsymbol{\lambda}_l = (\lambda_{l1}, \ldots, \lambda_{lp})$ pour $(l = 1, \ldots, c)$ sous les contraintes $\lambda_{lj} > 0$ et $\prod_{j=1}^{p} \lambda_{lj} = 1$.

Les vecteurs de pondération sont calculés par

$$\lambda_{lj}^{(t)} = \frac{\left\{\prod_{h=1}^{p} \left(\sum_{i=1}^{n} K^T[\delta(f(e_i), l)] d_h^2(e_i, g_l)\right)\right\}^{\frac{1}{p}}}{\sum_{i=1}^{n} K^T[\delta(f(e_i), l)] d_j^2(e_i, g_l)} \tag{4}$$

Dans **l'étape d'affectation**, les prototypes $g_l \in E \, \forall l = 1, \ldots, c$ et la matrice de pondération $\boldsymbol{\Lambda}$ sont fixés.

La fonction de coût J est minimisée par rapport à la fonction d'identification f et chaque objet e_i est affecté à la classe ou au neurone le plus proche :

$$r = f(e_i) = \arg \min_{1 \leq h \leq c} \sum_{l=1}^{c} K^T[\delta(h, l)] \sum_{j=1}^{p} \lambda_{lj} d_j(e_i, g_l) \tag{5}$$

L'algorithme.

1) *Initialisation :*
   Fixer le nombre c de neurones ou classes, la fonction de distance topologique δ, la fonction de voisinage K^T avec T_{min} et T_{max} et le nombre maximum d'iterations N_{iter}.
   Selectioner au hasard c objets de E qui seront les prototypes $g_l^{(0)} \in E \, (l = 1, \ldots, c)$;
   Initialiser la matrice de pondération $\boldsymbol{\lambda}_l{}^{(0)} = (1, \ldots, 1) \, (l = 1, \ldots, c)$;
   Faire $t \leftarrow 0$ puis calculer T ;
   Affecter chaque individu e_i au neurone le plus proche selon l'équation (5).
2) *Étape 1 : Représentation.*
   Faire $t \leftarrow t + 1$
   Calculer les prototypes $g_l^{(t)} \, (l = 1, \ldots, c)$ selon l'équation (3)
3) *Étape 2 : Pondération.*
   Calculer les composants $\lambda_{rj}^{(t)}$ la matrice de pondération $\boldsymbol{\lambda}_l^{(t)} \, (l = 1, \ldots, c; j = 1, \ldots, p)$ selon l'équation (4)
4) *Étape 3 : Affectation.*
   Affecter chaque individu $e_i \, (i = 1, \ldots, n)$ à la classe la plus proche selon l'équation (5) définir la fonction d'identification f.
5) *Critère d'arrêt.*
   puis calculer T si $t = N_{iter}$ alors STOP ; sinon aller vers 2 (Étape 1 : Représentation).

3 Application aux jeux de données “iris” et “thyroid”

Nous considérons ici deux jeux de données quantitatives de l'UCI “machine learning repository” : “iris” (150 individus, 4 variables et 3 classes de 50 individus) et “thyroid” (215 individus, 5 variables et 3 classes avec 150, 35 et 30 individus). Les parametres de l'algorithme ont été les suivantes : $N_{Iter} = 500$; $T_{max} = 2$; $T_{min} = 0.2$; grille de taille $2 \times 5 = 10$ neurones. La fonction noyau a été $K^T(\delta(c,r)) = exp(-\frac{(\delta(c,r))^2}{2T^2})$ et δ est la distance Euclidienne. Deux tableaux de dissimilarités ont été calculés à partir des deux tableaux originaux en utilisant la distance euclidienne sur toutes les variables. L'algorithme décrit dans El Golli et al. (2006) a été appliqué à ces tableaux de dissimilarité. Ensuite, nous avons créé, respectivement, 4 et 5 tableaux de dissimilarités en utilisant la distance euclidienne, respectivement, sur chacune des 4 et 5 variables quantitatives des jeux de données “iris” et “thyroid”. L'algorithme introduit dans cet article a été appliqué aux 3 et 4 tableaux de dissimilarités simultanement.

Ces algorithmes ont été executés 100 fois et le meilleur résultat, selon la fonction de coût, a été choisi. Les partitions finales en 10 classes données pour chacun de deux algorithmes a été comparé avec la partition *a priori* en 3 classes de chaque jeux de données. Le critère de comparaison a été l'indice corrigé de Rand (CR). Pour l'algorithm décrit dans El Golli et al. (2006), les résultats ont étÍes suivantes : $CR = 0.428$ (“iris”) et $CR = 0.364$ (“thyroid”). Pour l'algorithme introduit dans cet article : $CR = 0.462$ (“iris”) et $CR = 0.634$ (“thyroid”). Les résultats obtenus sur ces deux exemples montrent l'interêt de la méthode proposée.

Références

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

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

El Golli, A., F. Rossi, B. Conan-Guez, et Y. Lechevallier (2006). Une adaptation des cartes auto-organisatrices pour des données décrites par un tableau de dissimilarités. *Revue Statistique Appliquée LIV*, 33–64.

Frank, A. et A. Asuncion (2010). UCI machine learning repository.

Kaufman, L. et P. Rousseeuw (1990). *Finding Groups in Data*. New York: Wiley.

Kohonen, E. (1994). *Self-Organizing Maps*. Berlin: Springer Verlag.

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 Letters 23*, 675–686.

Summary

This paper gives an adaptation of the self-organizing maps which takes into account simultaneously several dissimilarity matrices. This algorithm is applied to two data sets from the UCI machine learning repository in order to show its usefulness.

Classification non supervisée à deux niveaux guidée par le voisinage et la densité

Guénaël Cabanes*

*LIPN, Institut Galilée, 99 Av. J.B. Clément, 93430 Villetaneuse

1 Introduction

La classification non supervisée (ou *clustering*) est un outil important en analyse exploratoire de données non étiquetées, mais un problème extrêmement difficile. Il existe de nombreuses méthodes de classification non supervisée. En particulier, les algorithmes de "classification à deux niveaux" procèdent en deux phases pour identifier les groupes dans une base de données. Dans la première phase du processus, l'algorithme estime des référents représentant des micro-groupes. Dans la deuxième phase, les partitions associées à chaque référent sont utilisées pour former la classification finale des données en utilisant une méthode de classification traditionnelle. Cependant, le déroulement séquentiel des deux niveaux s'accompagne inévitablement d'une perte d'information sur la structure des données. L'objectif de ce travail est donc de proposer une méthode de classification à deux niveaux simultanés, qui s'appuie sur l'apprentissage d'une SOM (Self Organizing Map : Kohonen (2001)) tout en conservant le maximum d'information sur la structure des données. Pour cela nous proposons de détecter la structure des données pendant l'apprentissage de la SOM.

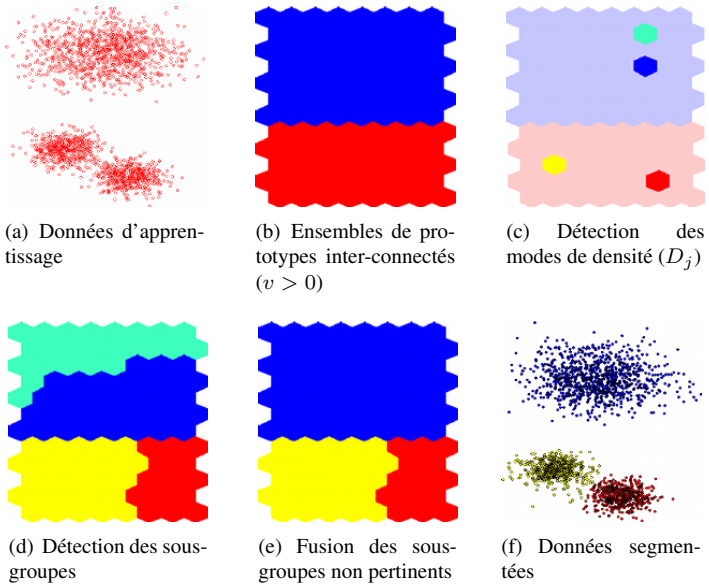
2 Classification guidée par le voisinage et la densité

Nous proposons une approche basée à la fois sur la distance et sur la détection des modes de densités, s'appuyant sur le fait qu'un regroupement de données peut être défini comme une région de l'espace de représentation localement dense en données, entourée par une région de faible densité (Ultsch, 2005; Pamudurthy et al., 2007). Cette approche est particulièrement efficace lorsque les groupes se touchent ou en présence de bruit. Elle est aussi efficace pour la détection des groupes non convexes. De plus, la méthode proposée regroupe automatiquement les données, c'est-à-dire que le nombre de groupes est déterminé automatiquement pendant le processus d'apprentissage, aucune hypothèse a priori sur le nombre de groupes n'est exigée. Cette approche a été évaluée sur un jeux de problèmes fondamentaux représentant différentes difficultés pour la classification et montre d'excellents résultats comparé aux approches classiques.

L'algorithme DS2L-SOM (Density-based Simultaneous 2-Level – SOM : Cabanes et Bennani, 2008) apprend simultanément les prototypes (référents) d'une carte auto-organisatrice et

sa segmentation en utilisant à la fois des informations sur les distances et la densité des données. Chaque connexion de voisinage est associée à une valeur réelle v qui indique la pertinence de la connexion entre deux neurones. Ainsi, à la fin de l'apprentissage, un ensemble de prototypes inter-connectés sera une bonne représentation de sous-groupes bien séparés de l'ensemble des données. Nous proposons en outre d'associer à chaque unité j une estimation de la densité locale des données D_j, de manière à détecter les fluctuations de densité qui définissent les frontières entre groupes en contact. Pour chaque donnée, cette valeur de densité sera augmentée pour tous les prototypes, en fonction de la distance euclidienne entre le prototype et les données.

A la fin du processus d'apprentissage, nous utilisons un algorithme de raffinement qui utilise les informations de connexions et de densités pour détecter les groupes (voir Figure 1). Les neurones qui sont reliés par des connexions de voisinage tels que $v > 0$ définissent des groupes bien distincts. Nous utilisons alors une méthode de "Watersheds" (Vincent et Soille, 1991) sur la carte de la densité de chacun de ces groupes pour détecter les zones de faible densité à l'intérieur des groupes bien séparés, de façon à caractériser les sous-groupes définis par la densité. Nous utilisons pour chaque paire de sous-groupes adjacents un indice "densité-dépendant" (Yue et al., 2004) pour déterminer si une zone de faible densité est un indicateur fiable de la structure des données, ou si elle doit être considérée comme une fluctuation aléatoire de la densité. On peut noter que, dans l'algorithme DS2L-SOM, l'estimation de la densité locale des données est faite pendant la formation de la carte, c'est-à-dire qu'il n'est pas nécessaire de conserver les données en mémoire.



(a) Données d'apprentissage

(b) Ensembles de prototypes inter-connectés ($v > 0$)

(c) Détection des modes de densité (D_j)

(d) Détection des sous-groupes

(e) Fusion des sous-groupes non pertinents

(f) Données segmentées

FIG. 1 – *Algorithme de raffinement. Chaque prototype est représenté par un hexagone.*

Les résultats expérimentaux sur des bases de données artificielles et réelles de grandes dimensions montrent que DS2L-SOM est capable de retrouver sans erreur la classification attendue et le bon nombre de groupes (voir Figure 2 pour un exemple).

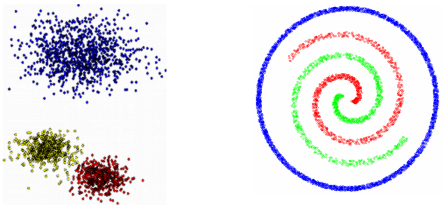


FIG. 2 – *Exemples de classifications automatiques obtenues avec DS2L-SOM*

3 Application

Nous avons souhaité appliquer nos nouvelles méthodes de classification non supervisée à l'analyse de données issues de la recherche expérimentale, de façon à montrer son efficacité pour l'extraction de connaissances dans ce domaine et la découverte de résultats scientifiques. Notre objectif est l'étude de la dynamique d'un déménagement d'une colonie de fourmis. Nous souhaitons caractériser et analyser l'évolution de l'ensemble de la colonie, comprendre comment les fourmis agissent et répartissent leur rôle pendant le processus du déménagement. Pour cela nous avons mis au point un dispositif de suivi RFID (Radio Frequency IDentification). Les Tags RFID, collés sur chaque fourmi, peuvent être détectés par un lecteur RFID capable d'enregistrer la présence d'un grand nombre de Tags en un seul scan et de les identifier (Cabanes et al., 2008).

Le dispositif expérimental pour cette expérience est composé de deux nids de trois salles et d'une zone de fourragement connectées linéairement par six tunnels (Figure 3). Chaque tunnel est équipé de lecteurs RFID qui détectent le passage et la direction des individus taggés entre les salles. La position d'un individu peut être déduite sans ambiguïté par l'information fournie par les lecteurs dans les tunnels.

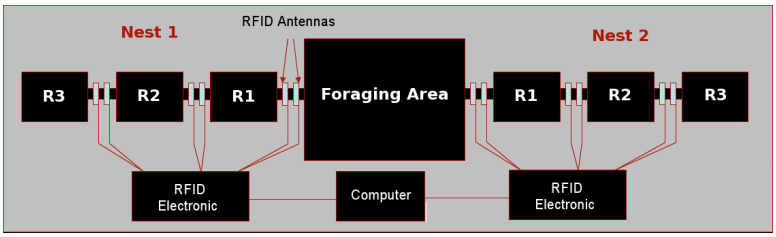


FIG. 3 – *Le dispositif RFID expérimental et un exemple de détection enregistrée.*

Au temps $t = 0$ nous ouvrons le premier nid et allumons une lampe à néon produisant une lumière puissante (répulsive pour les fourmis), puis nous enregistrons le déplacement de la colonie jusqu'à ce que la totalité du couvain ait été déplacée dans le second nid (environ 4 heures).

Dans le but de définir des patterns comportementaux et de détecter les changements de comportement au fil du temps, nous appliquons l'algorithme DS2L-SOM sur des fenêtres temporelles glissantes décrivant chaque séquence de déplacement individuel. A la fin du processus d'apprentissage, huit groupes de patterns comportementaux apparaissent $A_i (1 \leq i \leq 8)$. Les

séquences d'activité de chaque fourmi ont été étiquetées en fonction de ces comportements, puis des paramètres globaux ont été extraits, traçant le profil de la colonie de fourmis pendant la phase de migration (Figure 4).

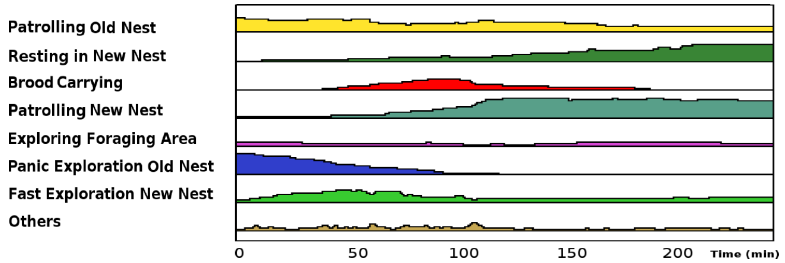


FIG. 4 – *Évolution du nombre d'individus impliqués dans les différentes activités.*

D'un point de vue éthologique, les résultats précédents sont d'une grande aide pour comprendre comment les tâches sont distribuées pendant le déménagement d'un nid. En fait, nous obtenons une description très précise de la dynamique de l'ensemble de la colonie pendant toutes les phases de la migration, ce qui nous permet d'émettre de fortes hypothèses au sujet de la fonction des différents comportements pendant la phase de déménagement du nid.

Références

Cabanes, G. et Y. Bennani (2008). A local density-based simultaneous two-level algorithm for topographic clustering. In *IJCNN*, pp. 1176–1182.

Cabanes, G., Y. Bennani, C. Chartagnat, et D. Fresneau (2008). Topographic connectionist unsupervised learning for RFID behavior data mining. In *IWRT'08*, pp. 63–72.

Kohonen, T. (2001). *Self-Organizing Maps*. Berlin : Springer-Verlag.

Pamudurthy, S. R., S. Chandrakala, et C. C. Sakhar (2007). Local density estimation based clustering. *Prodeeding of International Joint Conference on Neural Networks*, 1338–1343.

Ultsch, A. (2005). Clustering with SOM : U*C. In *Procedings of the Workshop on Self-Organizing Maps*, pp. 75–82.

Vincent, L. et P. Soille (1991). Watersheds in digital spaces : An efficient algorithm based on immersion simulation. *IEEE Trans. Pattern Anal. Mach. Intell. 13*, 583–598.

Yue, S.-H., P. Li, J.-D. Guo, et S.-G. Zhou (2004). Using greedy algorithm : DBSCAN revisited II. *Journal of Zhejiang University SCIENCE 5*(11), 1405–1412.

Summary

The work presented here concerns the development of approaches based on self-organizing map (SOM) for the discovery and monitoring of class structures in the data. We also present a real application for the monitoring of individuals in an RFID device. It is a study of the spatiotemporal behavior of an ant colony during emigration.

Symbolic Data Analysis and Formal Concept Analysis

Mehdi Kaytoue*, Sergei O. Kuznetsov**, Amedeo Napoli*, Géraldine Polaillon***

* LORIA – Campus Scientifique, B.P. 239 – Vandœuvre-lès-Nancy – France
Mehdi.Kaytoue@loria.fr;Amedeo.Napoli@loria.fr
** HSE – Pokrovskiy Bd. 11 – 109028 Moscow – Russia
skuznetsov@yandex.ru
*** E3S Supélec – 3 rue Joliot-Curie – 91192 Gif sur Yvette – France
Geraldine.Polaillon@supelec.fr

Abstract. Formal concept analysis (FCA) can be used for designing concept lattices from binary data for knowledge discovery purposes. Pattern structures in FCA are able to deal with complex data. In addition, this formalism provides a concise and an efficient algorithmic view of the formalism of symbolic data analysis (SDA).

1 Introduction

Many classification problems can be formalized by means of a *formal context*, a binary relation between an object set and an attribute set indicating whether an object has or does not have an attribute (see Ganter and Wille (1999)). According to the so-called *Galois connection*, one may classify within *formal concepts* a set of objects sharing a same maximal set of attributes, and vice-versa. Concepts are ordered in a lattice structure called *concept lattice* within the Formal Concept Analysis (FCA) framework. FCA can be used for a number of purposes like knowledge formalization and acquisition, ontology design, and data mining. To handle complex data in FCA, *pattern structures* have been proposed as a generalization of formal contexts to complex data (see Kuznetsov (2009); Kaytoue et al. (2011)). On the other hand, Symbolic Data Analysis (SDA, see Bock and Diday (2000)) aims at analyzing data such as numbers, intervals, sets of discrete values, etc. An object is described by a vector of values with each dimension corresponding to a variable, and each variable may be of different type. In Brito (1994); Brito and Polaillon (2005), the problem is addressed of building concept lattices by formalizing "symbolic objects" in SDA and properly defined Galois connections between these symbolic individuals and their descriptions. The links between the FCA and SDA approaches still remain unclear. Although both methods show the same behavior when working on the same data, the goal of this paper is to discuss how the SDA formalism for building concept lattices can be taken into account in FCA in a universal way, to facilitate comprehension and future extension (see also Agarwal et al. (2011)).

The paper is organized as follows. Section 2, 3 respectively present SDA, and pattern structures. Both approaches are compared and discussed in Section 4. Limited by space, we assume that the reader is familiar with FCA (see Ganter and Wille (1999)).

2 Symbolic Galois Lattices in Symbolic Data Analysis

	y_1	y_2
ω_1	[75, 80]	[1, 2]
ω_2	[60, 80]	[1, 1]
ω_3	[50, 70]	[2, 2]
ω_4	[72, 73]	[1, 2]

TAB. 1 –

The formalism of "Symbolic Data Analysis" was introduced and fully described (among others) in Brito (1994); Brito and Polaillon (2005). Due to place restrictions, we will not go into the details of SDA and we will briefly introduce some basic elements necessary for understanding this paper with the help of an example, see Table 1. Let $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ be a set of objects described by two variables y_1 with range $O_1 = \{[75, 80], [60, 80], [50, 70], [72, 73]\}$ and y_2 with range $O_2 = \{[1, 2], [1, 1], [2, 2]\}$. Then $(y_1 \subseteq [70, 80])$ can be considered as an "intensional description" –or elementary symbolic object– whose "extension" is the set $\{\omega_1, \omega_4\}$. Then an "assertion object" –that could be termed as a (generalized) symbolic object– is a conjunction of such elementary symbolic objects. For example, $d_1 = (y_1 \subseteq [60, 80]) \wedge (y_2 \subseteq [1, 2])$ describes the set $ext(d_1) = \{\omega_1, \omega_2, \omega_4\}$.

A partial ordering between description can be defined as follows: if d_1 and d_2 are two (generalized) intensional descriptions, then $d_1 \leq d_2 \Leftrightarrow ext(d_1) \subseteq ext(d_2)$. Further, Galois connections can be defined between $\wp(\Omega)$ and A depending on the choice of a "generalization operator" for building the upper bound of two assertions objects (see Brito (1994); Brito and Polaillon (2005)).

3 Pattern Concept Lattices

Pattern structures are introduced in Ganter and Kuznetsov (2001) in full compliance with FCA and can be thought of as a "generalization" of formal contexts to complex data from which a concept lattice can be built without any *a priori* scaling.

Formally, let G be a set of objects, $(D, \sqcap)$ be a semi-lattice of object descriptions, and $\delta : G \rightarrow D$ be a mapping. $(G, (D, \sqcap), \delta)$ is called a *pattern structure*. Elements of D are called *patterns* and are ordered by ordering relation $\sqsubseteq$, i.e. given $c, d \in D$, we have $c \sqsubseteq d \Longleftrightarrow c \sqcap d = c$. We use the operator $(.)^{\square}$ to derive the following images, where operator $(.)^{\square}$ corresponds to operator $(.)'$ in standard FCA:

$$A^{\square} = \bigsqcap_{g \in A} \delta(g), \text{ for any } A \subseteq G,$$
$$d^{\square} = \{g \in G \mid d \sqsubseteq \delta(g)\}, \text{ for any } d \in (D, \sqcap)$$

These operators form a Galois connection between $(\mathfrak{P}(G), \subseteq)$ and $(D, \sqsubseteq)$. $(.)^{\square\square}$ is a closure operator. *Pattern concepts* of $(G, (D, \sqcap), \delta)$ are pairs of the form (A, d), $A \subseteq G$, $d \in D$, such that $A^{\square} = d$ and $A = d^{\square}$, and d is called a *pattern intent* while A is a *pattern extent*. When partially ordered by $(A_1, d_1) \leq (A_2, d_2) \Leftrightarrow A_1 \subseteq A_2$ $(\Leftrightarrow d_2 \sqsubseteq d_1)$, the set of all pattern concepts forms a complete lattice called a *pattern concept lattice*. An example is given in the next section. Standard FCA algorithms need slight modification to compute the pattern concept lattice, see e.g. Ganter and Kuznetsov (2001); Kaytoue et al. (2011).

	y_1	y_2	y_3
g_1	[75,80]	[1,2]	{a,b}
g_2	[60,80]	[1,1]	{d,e}
g_3	[50,70]	[2,2]	{a,c}
g_4	[72,73]	[1,2]	{a}

TAB. 2 –

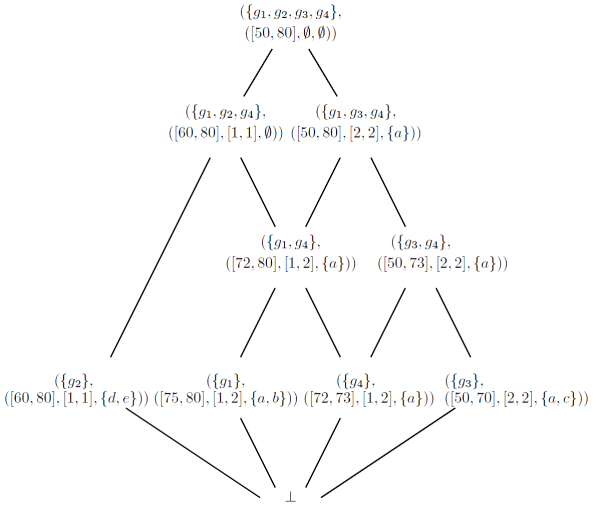


FIG. 1 – *Pattern concept lattice designed from Table 2.*

4 Symbolic Galois Lattices with Pattern Structures

SDA works on data tables where each column corresponds to a variable y_i. Pattern structures consider $(D, \sqcap)$ as corresponding to one variable in terms of SDA. Thus, given a set $Y = \{y_1, ..., y_p\}$ of p variables, we consider the direct product $(D, \sqcap) = (D_{y_1}, \sqcap_{y_1}) \times ... \times (D_{y_p}, \sqcap_{y_p})$ of all semi-lattices $(D_{y_i}, \sqcap_{y_i})$ for each $y_i \in Y$. $(D, \sqcap)$ is a semi-lattice itself containing all possible descriptions of objects and sets of objects, and corresponds to the set of possible intensional descriptions in SDA. The partial ordering $\sqsubseteq$ in $(D, \sqcap)$ is such that, for any $c, d \in D$, $c \sqcap d = c \iff c \sqsubseteq d$. Then a pattern $d \in D = (d_1, ..., d_p)$ is called a *pattern vector*. For any $c, d \in D$: $c \sqcap d = (c_1 \sqcap_{y_1} d_1, ..., c_p \sqcap_{y_p} d_p)$ and $c \sqsubseteq d \Leftrightarrow c_i \sqsubseteq_{y_i} d_i \; \forall i = 1...p$. A dimension i of a pattern vector corresponds to a variable y_i which may have a different type. For example, considering intervals, let us define define $\sqcap_{y_1}$ as interval convexification, i.e. with $a_1, b_1, a_2, b_2 \in \mathbb{R}$: $[a_1, b_1] \sqcap_y [a_2, b_2] = [min(a_1, a_2), max(b_1, b_2)]$ and $[a_1, b_1] \sqsubseteq_y [a_2, b_2] \Leftrightarrow [a_1, b_1] \supseteq [a_2, b_2]$. Based on this partial ordering of descriptions, the general Galois connection defined for pattern structures allows to compute pattern concepts and lattices from heterogeneous data.

The example in Table 2 can be represented as a pattern structure $(G, (D, \sqcap), \delta)$ where $G = \{g_1, g_2, g_3, g_4\}$ and $\delta(g_1) = ([75, 80], [1, 2], \{a, b\})$. Descriptions contain two interval-valued variables: y_1 where ordering is based on interval intersection, y_2 where ordering is based on interval convexification, and one categorical multi-valued variable y_3 where ordering is based on inclusion. For example, $\{g_1, g_3\}^{\square\square} = ([50, 80], [2, 2], \{a\})$ and $\{g_1, g_3\}^{\square\square} = \{g_1, g_3, g_4\}$. Hence, $(\{g_1, g_3, g_4\}, ([50, 80], [2, 2], \{a\})$ is a pattern concept of $(G, (D, \sqcap), \delta)$.

The links with SDA formalism are natural but the algorithmic machinery is not the same at all: algorithms building pattern structures are very efficient and can easily build the SDA lattices (see Kaytoue et al. (2011)), but the converse is not true. Moreover, pattern structures consider object descriptions in their original form and propose any kind of partial ordering

between descriptions (compare with intersection and union, the actual two types of partial ordering in SDA).

5 Conclusion

Pattern structures allow to directly consider complex data, avoiding to represent descriptions as symbolic/assertion objects. One general Galois connection is sufficient to consider several data-types, hence it is not required to define a new Galois connection for different data-types and description generalization operations (with union and intersection in SDA). Indeed, the main core of pattern structures lies in defining an appropriate semi-lattice operation inducing a partial order of descriptions. This is rather simple with numerical and categorical data as illustrated in this paper.

Avoiding discretization and loss of information, generally leads to a great amount of concepts. In SDA, it is shown how to reduce concept lattices to simpler hierarchies with reduction techniques based on quality criteria defined in SDA, but this requires to work with a concept lattice already computed, which can be bottleneck for very large databases. On the other hand, pattern structures propose to project object descriptions to "simpler ones" before computation, allowing to reduce the number of concepts. This gives interesting perspectives of research to consider well studied SDA quality criteria within pattern structures.

References

Agarwal, P., M. Kaytoue, S. O. Kuznetsov, A. Napoli, and G. Polaillon (2011). Symbolic Galois Lattices with Pattern Structures. In *Proceedings of RSFDGrC-2011*, LNAI 6743, pp. 191–198. Springer.

Bock, H.-H. and E. Diday (Eds.) (2000). *Analysis of Symbolic Data*. Springer.

Brito, P. (1994). Order structure of symbolic assertion objects. *IEEE Transactions on Knowledge and Data Engineering 6*(5), 830–834.

Brito, P. and G. Polaillon (2005). Structuring probabilistic data by Galois lattices. *Mathématiques et sciences humaines 169*, 77–104.

Ganter, B. and S. O. Kuznetsov (2001). Pattern Structures and Their Projections. In *Proceedings of ICCS-2001*, LNCS 2120, pp. 129–142. Springer.

Ganter, B. and R. Wille (1999). *Formal Concept Analysis*. Springer.

Kaytoue, M., S. O. Kuznetsov, A. Napoli, and S. Duplessis (2011). Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis. *Information Science 181*(10), 1989–2001.

Kuznetsov, S. O. (2009). Pattern Structures for Analyzing Complex Data. In *Proceedings of RSFDGrC-2009*, LNAI 5908, pp. 33–44. Springer.

Résumé

L'analyse formelle de concepts (FCA) est utilisée pour construire des treillis de concepts à partir de tables de données binaires pour des besoins de découverte de connaissances. Les structures de patrons en FCA sont capables de prendre en compte des données complexes et de plus fournissent une vue concise et algorithmique efficace sur le formalisme des objets symboliques (SDA).

Homogénéité dans l'analyse conceptuelle : un cadre commun pour variables numériques, ordinales et modales

Géraldine Polaillon*, Paula Brito**

*SUPELEC Science des Systèmes (E3S) - Département Informatique
Plateau de Moulon, 3 rue Joliot Curie, 91192 Gif-sur-Yvette cedex, France,
geraldine.polaillon@supelec.fr
**Faculdade de Economia & LIAAD/INESC-Porto L.A., Universidade do Porto
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal, *mpbrito@fep.up.pt*

Résumé. Le cadre de ce travail est l'analyse de données par les treillis de Galois. Les données peuvent avoir des valeurs ordonnées, intervalles ou prendre la forme de distribution de probabilités/fréquences. Elles sont traitées dans un cadre commun par un opérateur de généralisation calculant les intensions par intervalles. Pour les données de distribution, les concepts sont plus homogènes et plus facilement interprétables que ceux obtenus précedemment.

1 Analyse de données par les treillis de Galois

Soit E un ensemble d'individus décrits par des variables quantitatives (réelles ou à valeurs intervalle), ordinales et modales. Les variables modales permettent d'associer à chaque individu une distribution de probabilité/fréquence sur un ensemble fini de modalités.

L'utilisation des treillis de Galois en analyse des données est d'abord due à Barbut et Monjardet (1970) et a largement été popularisée par Ganter et Wille (1999). Par la suite, des travaux ont approfondi différents aspects, qui ne rentrent pas dans le cadre de ce papier. Soient deux applications, f de $A \subseteq E$ vers $B \subseteq O$, où O est l'ensemble des attributs, et g de $B \subseteq O$ vers $A \subseteq E$. Le couple (f, g) constitue une correspondance de Galois entre $(P(E), \subseteq)$ et $(P(O), \subseteq)$. Un concept est un couple (A, B) où $A \subseteq E, B \subseteq O, A = g(B)$ et $B = f(A)$; A est appelé l'*extension*, et B l'*intension*. Cette approche a été appliquée à des variables non binaires sous condition préalable d'un recodage des données, ce qui augmente de façon prohibitive la taille des données.

Brito (1994) a défini des correspondances de Galois pour des variables quantitatives (classiques ou non). Cette approche permet de traiter directement les données sans recodage. Elle a été étendue aux variables modales (Brito et Polaillon (2005)). Les variables ordinales ont été traitées par Pfaltz (2007), utilisant une approche similaire à celle proposée dans Brito et Polaillon (2005). D'autres auteurs, e.g. Assaghir et al. (2009), proposent de traiter des données multi-valuées en regroupant les valeurs similaires par rapport à un seuil donné.

Dans ce papier, nous proposons un cadre commun pour les variables quantitatives (réelles ou à valeurs intervalle), ordinales et modales, définissant un opérateur de généralisation qui

calcule les intensions par intervalles de valeurs. Les prochaines sections détaillent la généralisation pour chaque type de variable.

2 Variables quantitatives et variables à valeur intervalle

Soit $E = \{\omega_1, ..., \omega_n\}$ l'ensemble de n individus ou objets, $Y_1, \ldots, Y_p$ des variables réelles ou à valeur intervalle et $Y_j(\omega_i) = [l_{ij}, u_{ij}]$. Les variables réelles sont considérées comme un cas particulier des variables à valeur intervalle, car $Y_j(\omega_i) = x = [x, x]$.

Soit $A = \{\omega_1, \ldots, \omega_h\} \subseteq E$. La généralisation par l'union est définie (Brito (1994)) par $f : P(E) \to I^p$ où I est l'ensemble des intervalles de $\mathbb{R}$, muni de l'ordre de l'inclusion, telle que $f(A) = (I_1, \ldots, I_p)$, avec $I_j = [Min\{l_{ij}\}, Max\{u_{ij}\}]$, $\omega_i \in A$, $j = 1, \ldots, p$, i.e., I_j est le plus petit intervalle qui contient toutes les valeurs prises par les éléments de A pour la variable Y_j. Soit $g : I^p \to P(E)$ définie par $g((I_1, \ldots, I_p)) =$
$\{\omega_i \in E : Y_j(\omega_i) \subseteq I_j, j = 1, \ldots, p\}$. Le couple (f, g) est une correspondance de Galois.
De même, on peut généraliser par l'intersection en définissant f et g par : $f^* : P(E) \to I^p$, $f(A) = (I_1, \ldots, I_p)$, avec $I_j = [Max\{l_{ij}\}, Min\{u_{ij}\}]$ si $Max\{l_{ij}\} \leq Min\{u_{ij}\}, \omega_i \in A$, $I_j = \oslash$ sinon, $j = 1, \ldots, p$ (i.e., I_j est le plus grand intervalle qui est contenu dans tous les valeurs-intervalle prises par les éléments de A pour la variable Y_j) et $g^* : I^p \to P(E)$ avec $g^*((I_1, \ldots, I_p)) = \{\omega_i \in E : Y_j(\omega_i) \supseteq I_j, j = 1, \ldots, p\}$. Le couple (f^*, g^*) constitue également une correspondance de Galois.

Exemple 1 : Soient 3 individus caractérisés par deux variables, âge et température, représentés par $\omega_1 = (40, [37, 38])$, $\omega_2 = (26, [39, 40])$, $\omega_3 = (35, [37, 38])$. Considérons $A = \{\omega_2, \omega_3\}$. La généralisation par l'union donne $f(A) = ([26, 35], [37, 40])$, qui décrit des individus dont l'âge varie entre 26 et 35 ans et dont la température varie entre 37 et 40 degrés ; $g(([26, 35], [37, 40])) = \{\omega_2, \omega_3\} = A$. Ici, $(A, ([26, 35], [37, 40]))$ est un concept.

3 Variables modales

Deux correspondances sont aussi presentées pour les variables modales (Brito et Polaillon (2005)). Soit $Y_1, \ldots, Y_p$ des variables modales, $O_j = \{m_{j1}, \ldots, m_{jk_j}\}$ avec k_j le nombre de modalités de la variable Y_j, M_j l'ensemble des distributions sur O_j, pour $j = 1, \ldots, p$ et $M = M_1 \times \ldots \times M_p$. Pour la variable Y_j, et l'individu $\omega_i \in E$, $Y_j(\omega_i) = \left\{m_{j1}(p_{j1}^{\omega_i}), \ldots, m_{jk_j}(p_{jk_j}^{\omega_i})\right\}$, où $p_{jk_\ell}^{\omega_i}$ est la probabilité/fréquence associée à la modalité $m_{j\ell}$ ($\ell = 1, \ldots, k_j$) de la variable Y_j, et l'individu ω_i. Soit $A = \{\omega_1, \ldots, \omega_h\} \subseteq E$.
La généralisation par le maximum est, pour chaque modalité $m_{j\ell}$, le maximum de ses probabilités/fréquences. Soit $f : P(E) \to M$, telle que $f(A) = (d_1, \ldots, d_p)$, avec $d_j = \{m_{j1}(t_{j1}), \ldots, m_{jk_j}(t_{jk_j})\}$, où $t_{j\ell} = Max\{p_{j\ell}^{\omega_i}, \omega_i \in A\}, \ell = 1, \ldots, k_j$. L'intension d'un ensemble $A \subseteq E$ est interprétée comme décrivant des objets pour lesquels *au plus* $t_{j\ell}$ cas présentent la modalité $m_{j\ell}, \ell = 1, \ldots, k_j, j = 1, \ldots, p$. Le couple (f, g) avec $g : M \to P(E)$ telle que, pour $d_j = \{m_{j1}(p_{j1}), \ldots, m_{jk_j}(p_{jk_j})\}$, $g((d_1, \ldots, d_p))$ =
$\left\{\omega_i \in E : p_{j\ell}^{\omega_i} \leq p_{j\ell}, \ell = 1, \ldots, k_j, j = 1, \ldots, p\right\}$, constitue une correspondance de Galois.
De même, nous généralisons par le minimum en prenant, pour chaque modalité, le minimum de ses probabilités/fréquences. Soit $f^* : P(E) \to M$, $f^*(A) = (d_1, \ldots, d_p)$, avec

$d_j = \{m_{j1}(v_{j1}), \ldots, m_{jk_j}(v_{jk_j})\}$, où $v_{j\ell} = Min\{p_{j\ell}^{\omega_i}, \omega_i \in A\}, \ell = 1, \ldots, k_j$, et qui représente des objets pour lesquels *au moins* $v_{j\ell}$ cas présentent la modalité $m_{j\ell}, \ell = 1, \ldots, k_j, j = 1, \ldots, p$. Le couple (f^*, g^*) avec $g^* : M \to P(E)$ telle que, pour $d_j = \{m_{j1}(p_{j1}), \ldots, m_{jk_j}(p_{jk_j})\}$, $g^*((d_1, \ldots, d_p)) = \left\{\omega_i \in E : p_{j\ell}^{\omega_i} \geq p_{j\ell}, \ell = 1, \ldots, k_j, j = 1, \ldots, p\right\}$ constitue également une correspondance de Galois.

Exemple 2 : Soient 4 groupes d'élèves synthétisés pour une note par catégories, a : note $<$ 10, b : note entre 10 et 15, c : note $>$ 15 comme suit : groupe 1 : $(a(0.2), b(0.6), c(0.2))$; groupe 2 : $(a(0.3), b(0.3), c(0.4))$; groupe 3 : $(a(0.1), b(0.6), c(0.3))$; groupe 4 : $(a(0.3), b(0.6), c(0.1))$; groupe 5 : $(a(0.5), b(0.3), c(0.2))$. L'intension de l'ensemble des groupes 1 et 2 par le maximum est $\{a(0.3), b(0.6), c(0.4)\}$, décrivant des groupes d'élèves dont au plus 30% ont a, au plus 60% ont b, et au plus 40% ont c. L'extension comprend les groupes 1, 2, 3 et 4. L'intension de l'ensemble des groupes 1 et 2 par le minimum est $\{a(0.2), b(0.3), c(0.2)\}$, décrivant des groupes d'élèves dont au moins 20% ont a, au moins 30% ont b, et au moins 20% ont c. L'extension comprend les groupes 1, 2 et 5.

4 Cadre commun : la généralisation par intervalle

Nous proposons de traiter les variables numériques (classiques ou à valeur intervalle), ordinales et modales dans un cadre unique de généralisation par intervalle.
Dans le cas de données numériques, on retrouve la généralisation par l'union.
Pour les variables modales, ceci revient à considérer, pour chaque modalité un intervalle de variation de sa probabilité/fréquence. En effet, la généralisation par le maximum ou par le minimum, définie dans la section 3, méne rapidement à une surgénéralisation, qui produit des intensions $f(A)$, $A \subseteq E$, non informatives.
Soit $M_j^I = \{m_{j\ell}(I_{j\ell}), \ell = 1, \ldots, k_j\}, m_{j\ell} \in O_j, I_{j\ell} \subseteq [0, 1]$ et $M^I = M_1^I \times \ldots \times M_p^I$.
La généralisation est alors définie par $f^I : P(E) \to M^I$, $f^I(A) = (d_1, \ldots, d_p)$, avec $d_j = \{m_{j1}(I_{j1}), \ldots, m_{jk_j}(I_{jk_j})\}$, où $I_{j\ell} = \left[Min\{p_{j\ell}^{\omega_i}\}, Max\{p_{j\ell}^{\omega_i}\}\right], \omega_i \in A, \ell = 1, \ldots, k_j, j = 1, \ldots, p$ et $g^I : M^I \to E, g((d_1, \ldots, d_p)) = \left\{\omega_i \in E : p_{j\ell}^{\omega_i} \in I_{j\ell}, \ell = 1, \ldots, k_j, j = 1, \ldots, p\right\}$.
Le couple (f^I, g^I) constitue une correspondance de Galois.
Sur les données de l'exemple 2, la généralisation par intervalle des groupes 1 et 2 est donnée par $\{a\,[0.2, 0.3]\,, b\,[0.3, 0.6]\,, c\,[0.2, 0.4]\}$, décrivant des groupes où entre 20% et 30% des élèves ont a, entre 30% et 60% ont b, et entre 20% et 40% ont c ; l'extension ne comprend maintenant que les groupes 1 et 2.
Le cas des variables ordinales a été traité par Pfaltz (2007), effectuant la généralisation par le maximum ou le minimum. Pour permettre plus de flexibilité, l'auteur propose de faire le choix de l'opérateur pour chaque variable. Cependant, il faut à chaque fois choisir un des opérateurs de généralisation, et la surgénéralisation n'est pas évitée. Nous proposons une généralisation des variables ordinales en considérant un intervalle de valeurs.

Exemple 3 : Considérons 4 individus ayant notés 3 films, Film 1, Film 2, Film 3 : ω_1 : (5, 5, 4) ; ω_2 : (5, 4, 4) ; ω_3 : (1, 2, 2) ; ω_4 : (2, 1, 1). L'intension par le maximum des individus 1 et 2 est $(5, 5, 4)$, décrivant des individus qui notent au plus 5 les Films 1 et 2 et au plus 4 le Film 3 ; l'extension correspondante comprend les utilisateurs 1,2,3,4. L'intension par le minimum des individus 3 et 4 est $(1, 1, 1)$ décrivant des individus qui notent au moins 1 chaque film ;

l'extension comprend tous les utilisateurs. La généralisation par intervalle des individus 1 et 2 donne l'intension $([5,5],[4,5],[4,4])$, qui décrit des individus attribuant des notes élevées à tous les Films ; de même pour les individus 3 et 4, $([1,2],[1,2],[1,2])$, décrit des individus attribuant des notes basses à tous les Films ; les extensions ne contenant pas d'autres individus, $(\{\omega_1,\omega_2\},([5,5],[4,5],[4,4])$, et $(\{\omega_3,\omega_4\},([1,2],[1,2],[1,2])$ sont des concepts.

Avec les opérateurs classiques, les classes homogènes n'apparaissent pas, car la surgénéralisation entraine des extensions larges. On évite ce problème en prenant l'intervalle de valeurs.

5 Conclusion

Nous proposons une méthode de généralisation pour des données numériques, ordinales et modales, basée sur des intervalles, définissant ainsi un cadre commum. D'une part, les concepts sont plus homogènes qu'avec des opérateurs de géneralisation par maximum et/ou minimum et d'autre part, des variables de type différent peuvent être traitées directement ensemble. L'approche proposée pour les variables ordinales permet d'analyser des tableaux de préférences, ouvrant des perspectives pour les systèmes de recommandation. La généralisation par intervalles est aussi à étudier en classification supervisée. Le nombre de concepts étant souvent très important, nous voulons identifier les concepts les plus pertinents, à fin d'éliminer l'effet d'individus atypiques. Dans la suite de nos travaux nous proposerons une méthode d'identification de concepts stables, utilisant un principe proche de la validation croisée.

Références

Assaghir, Z., M. Kaytoue, N. Messai, et A. Napoli (2009). On the mining of numerical data with formal concept analysis and similarity. In *Proc. SFC*, pp. 121–124.

Barbut, M. et B. Monjardet (1970). *Ordre et Classification, Algèbre et Combinatoire, Tomes I & II*. Paris : Hachette.

Brito, P. (1994). Order structure of symbolic assertion objects. *IEEE Trans. on Knowledge and Data Engineering 6*(5), 830–835.

Brito, P. et G. Polaillon (2005). Structuring probabilistic data by galois lattices. *Math. & Sci. Hum. / Mathematics and Social Sciences 169*(1), 77–104.

Ganter, B. et R. Wille (1999). *Formal Concept Analysis, Mathematical Foundations*. Springer.

Pfaltz, J. (2007). Representing numeric values in concept lattices. In J. Diatta, P. Eklund, et M. Liquiere (Eds.), *Pr. 5th Int. Conf. Concept Lattices and their Applications*, pp. 260–269.

Summary

This work concerns data analysis by Galois lattices. We consider real and interval-valued data, ordinal data, as well as data that consist on probability/frequency distributions. Data are considered on a common framework, by a generalisation operator that determines intensions by intervals. For distribution and ordinal data, the concepts are more homogeneous and easier to interpret than those obtained by using previously proposed operators.

Modélisation de données symboliques et application au cas des intervalles

Edwin Diday*

*CEREMADE, Université de Paris 9 Dauphine,75775 Paris
diday@ceremade.dauphine.fr

Résumé. L'Analyse des données symboliques est une fille de la Classification Automatique car son but est d'étudier des classes comme "individus". Ces "individus" sont décrits par des données symboliques (intervalles, histogrammes etc.) prenant ainsi en compte la variation interne des classes qu'ils représentent de façon non réductibles à des nombres ou à des catégories. Il s'agit ici d'étudier le lien entre la densité des données symboliques et les paramètres des lois qu'elles induisent.

1 Introduction

L'idée générale est qu'en Analyse des Données Symboliques (voir par exemple (Bertrand et Goupil, 2000), (Diday et Noirhomme, 2008)), les points de l'espace symbolique sont des descriptions de classes, de concepts (au sens des treillis de Galois) ou de catégories qui ont un sens et qui ont donc une certaine homogénéité interne. On peut donc en ADS utiliser cette chance de la cohérence interne des classes, en associant un modèle à chaque point de l'espace symbolique initial. Pour cela, on pose $Y = \varphi(X)$ où X est la variable aléatoire qui associe à chaque classe un point de l'espace des descriptions symboliques réduit pour simplifier à IR^k muni d'une mesure de probabilité et Y est la variable aléatoire qui associe à chaque point de l'espace symbolique la valeur des paramètres de la loi qui leur est associée. La loi de Y donne donc une vision des données symboliques dans l'espace des paramètres qui les modélise. Par exemple, dans le cas où chaque classe est décrite par un intervalle (d'âge ou de poids, par exemple), X associe à chaque classe un point de IR^2 considéré comme l'espace des données symboliques. En modélisant chaque intervalle par deux paramètres classiques sous hypothèse d'uniformité, on peut définir Y qui associe donc à chaque intervalle un point de IR^2 également associé à ces deux paramètres. La loi de Y donne donc une vision modélisée des données symboliques qui peut donner des informations intéressantes sur la moyenne, la variance etc. des classes, complémentaires à celle de l'espace symbolique initial. De façon générale, on peut modéliser l'espace symbolique de trois façons : soit au niveau des données symboliques elles-mêmes (dans un espace de distributions aléatoires (Diday et Vrac, 2005) en utilisant les copules (Soubdhan et al., 2009), en modélisant par une loi de Dirichlet, dans un espace d'intervalles (Brito et Duarte Silva, 2011) exprimés par le milieu et l'écart), soit au niveau de l'espace des paramètres du modèle associé à chaque donnée symbolique ((Bertrand et Goupil, 2000) dans le cas d'intervalles de lois uniformes, (Le-Rademacher et Billard, 2011) plus généralement), soit dans le cas où il existe une bijection $\varphi : Y = \varphi(X)$ en passant du modèle sur les données

symboliques au modèle sur les données paramétriques (et réciproquement) grâce à une formule de changement de variable classiquement utilisée en calcul d'intégrales. Cette bijection a été considérée dans (Le-Rademacher et Billard, 2011) mais seulement dans le cas discret qui débouche naturellement sur une égalité des densités de Y et de X. C'est l'objet de cet article de considérer cette troisième voie dans le cas général (i.e., non discret), en la concrétisant par l'exemple des données intervalles modélisées par des lois uniformes indépendantes.

2 Lien entre densités quand les V.A. sont liées par une bijection

Le résultat classique suivant permet de relier la densité de deux variables aléatoires quand elles sont reliées par une bijection.

Proposition 1

Soient deux variables aléatoires X, Y de distribution F et G continues et de densité f et g telles que $Y = \varphi(X)$. Sous l'hypothèse que φ est bijective et dérivable, on a :

$$g(y) = f(\varphi^{-1}(y))/|D| \tag{1}$$

où $D = \partial\varphi(x)/\partial x_1, ..., \partial x_p$ est la matrice jacobienne et |D| est le Jacobien de φ .

3 Application aux données symboliques de type intervalles

X est une variable aléatoire à valeur intervalle : X(w)=$(X_{min}, X_{max})(w)$=(x_{min}, x_{max}). La fonction de répartition de X est définie par $F = (F_{min}, F_{max})$ = Prob$(F_{min} < x_{min}, F_{max} < x_{max})$ et sa densité se définit par $f = \partial F/\partial x^j_{min}\partial x^j_{max}$. On sait d'autre part, que sous hypothèse d'uniformité la densité d'un intervalle est caractérisée par deux paramètres dont l'un est la moyenne $m(w) = (x_{min} + x_{max})/2$ et l'autre est la variance $v(w) = (x_{min} - x_{max})^2/12$. Soit φ telle que $\varphi(X) = (\varphi_m(X), \varphi_v(X)) = (m, v)$ et Y est la variable aléatoire $Y = \varphi(X)$ avec $Y = (Ym, Yv)$ où $Y_m = \varphi_m(X)$ et $Y_v = \varphi_v(X)$ de fonction de répartition G définie par : $G(y_m, y_v)$ =Prob$(Y_m < y_m, Y_v < y_v)$ et sa densité se définit par $g = \partial G/\partial y_{min}\partial y_{max}$.

Proposition 2

$$g(\varphi(x)) = 6f(x)/(x_{max} - x_{min}) \tag{2}$$

Démonstration

Il est facile de voir que φ est bijective du fait de la contrainte $x_{min} < x_{max}$. D'autre part, comme

$D = \partial y_1/\partial x_1.\partial y_2/\partial x_2 - \partial y_2/\partial x_1.\partial y_1/\partial x$

$D = \partial\varphi_m(X)/\partial x_{min}.\partial\varphi_v(X)/\partial x_{max} - \partial\varphi_m(X)/\partial x_{max}.\partial\varphi_v(X)/\partial x_{min}.$

$D = \frac{1}{2}2(x_{max} - x_{min})/12 - \frac{1}{2}2(x_{min} - x_{max})/12 = x_{max} - x_{min})/6.$

Il en résulte de (1) que $g(y) = 6f(x)/(x_{max} - x_{min})$

Cas général de p variables à valeur intervalle

Dans le cas général de p variables à valeur intervalle, on note $X = (X_1, ..., X_p)$, $X_j = (X^j_{min}, X^j_{max})$ et $X_j(w) = (x^j_{min}, x^j_{max})$.

La fonction de répartition de X est définie par F : $F(x^1_{min}, x^1_{max}, ..., x^p_{min}, x^p_{max}) =$ Prob$(F^1_{min} < x^1_{min}, F^1_{max} < x^1_{max}, ..., F^p_{min} < x^p_{min}, F^p_{max} < x^p_{max})$ et sa densité se définit par $f = \partial F/\partial x^1_{min}\partial x^1_{max}...\partial x^p_{min}\partial x^p_{max}$.
En supposant l'indépendance des lois uniformes, φ est telle que $\varphi(X) = (\varphi^1_m(X), \varphi^1_v(X), ..., \varphi^p_m(X), \varphi^p_v(X)) = (m^1, v^1, ..., m^p, v^p)$ avec $m^j(w) = (x^j_{min} + x^j_{max})/2$ et $v^j(w) = (x^j_{min} - x^j_{max})^2/12$.
On note $Y = (Y_1, ..., Y_p)$, avec $Y_j = (Y^j_m, Y^j_v)$, autrement dit, $Y = (Y^1_m, Y^1_v, ..., Y^p_m, Y^p_v)$, la v.a. $Y = \varphi(X)$ avec $Y^j_m = \varphi^j_m(X)$ et $Y^j_v = \varphi^j_v(X)$ de fonction de répartition $G = (Y^1_m, Y^1_v, ..., Y^p_m, Y^p_v)$:définie par $G(y^1_m, y^1_v, ..., y^p_m, y^p_v) =$ Prob$(Y^1_m < y^1_m, Y^1_v < y^1_m, , Y^p_m < y^p_m, Y^p_v < y^p_m)$ et sa densité se définit par $\partial g = \partial G/\partial y^1_m\partial y^1_v...\partial y^p_m\partial y^p_v$

Proposition 3

$$g(y) = 6^p f(x)/\prod_{j=1}^{p}(x^j_{max} - x^j_{min}) \; où \; x = \varphi^{-1}(y) \tag{3}$$

Démonstration
On voit d'abord que φ est bijective du fait des contraintes $x^j_{min} < x^j_{max}$. On démontre facilement par récurrence que le jacobien est égal à :
$D = \prod_{j=1}^{p}(\partial Y^j_m/\partial x^j_{min}.\partial Y^j_v/\partial x^j_{max} - \partial Y^j_v/\partial x^j_{min}.\partial Y^j_m/\partial x^j_{max})$. Or ,
$D^j = \partial Y^j_m/\partial x^j_{min}.\partial Y^j_v/\partial x^j_{max} - \partial Y^j_v/\partial x^j_{min}.\partial Y^j_m/\partial x^j_{max}$
$D^j = \partial\varphi^j_m(X^j)/\partial x^j_{min}\varphi^j_v(X^j)/\partial x^j_{max} - \partial\varphi^j_m(X^j)/\partial x^j_{min}.\varphi^j_v(X^j)/\partial x^j_{max}$.
$D^j = \frac{1}{2}2(X^j_{max} - X^j_{min})/12 - \frac{1}{2}2(x^j_{min} - x^j_{max})/12 = (X^j_{max} - X^j_{min})/6$.
D'où $D = \prod_{j=1}^{p}(X^j_{max} - X^j_{min})/6^p$. Il en résulte d'après (1) que :
$g(y) = 6^p f(\varphi^{-1}(y))/\prod_{j=1}^{p}(x^j_{max} - x^j_{min})$ d'où le résultat recherché.

4 Extension à d'autres types de données et modèles, décomposition de mélanges

Nous donnons ici deux pistes de recherche ouvertes. La première consiste à étendre le cas des intervalles au cas de variables symboliques à valeur histogramme. Chaque histogramme étant considéré comme une suite de k intervalles pondérés. On peut alors définir une bijection dérivable en associant à chaque histogramme d'abord le barycentre des mileux de ces intervalles munis de leur pondération, puis les moments d'ordre r associés à ce barycentre. On peut ajouter d'autres paramètres en calculant d'abord le barycentre des variances des intervalles supposés de lois uniformes, puis les moments d'ordre r associés à ce barycentre. Avec r suffisamment grand et en supprimant éventuellement les paramètres redondants, il doit alors être possible d'obtenir une bijection dérivable. La seconde piste consiste à considérer f comme un mélange de lois f_i, il en résulte que g est aussi un mélange de lois g_i. Si l'on suppose de plus qu'une bijection φ_i est associée à chaque loi f_i et que f_i, g_i, φ_i ont respectivement des paramètres notés a_i, b_i, c_i (qui peuvent être des vecteurs) alors, la formule classique de décomposition de mélange se généralise alors sous la forme suivante : $g(y) = \Sigma_i p_i f_i(\varphi_i^{-1}(y), a_i)\varphi_i'(\varphi_i^{-1}(y, c_i), c_i)$. En posant : $g_i(y, b_i) = f_i(\varphi_i^{-1}(y, c_i), a_i)\varphi_i'(\varphi_i^{-1}(y, c_i), c_i)$

on a $g(y, b) = \Sigma_i p_i g_i(y, b_i)$. Les φ_i peuvent être ainsi ajustés à chaque composant du mélange à chaque pas de l'algorithme des nuées dynamiques (qui a l'avantage de ne pas être biaisé au niveau des partitions obtenues) ou EM de façon à maximiser la vraisemblance.

5 Conclusion

Il est intéressant de calculer la densité dans l'espace des données symboliques mais il est aussi intéressant de voir comment varient dans un autre espace des paramètres associés à chaque donnée symbolique, comme la moyenne, la variance, les moments. Il est donc intéressant de voir comment on peut passer simplement d'un espace à l'autre. Ainsi, ayant estimé les paramètres de la loi de f dans un espace de données symboliques, on peut en déduire ceux de la loi de g des paramètres par un calcul approprié pour chaque type de loi à condition qu'il existe une bijection entre les variables aléatoires respectivement associées à f et g. Nous avons fait ici ce calcul dans le cas de données intervalles de lois uniformes indépendantes. Cette contrainte d'indépendance peut être réduite par l'utilisation de copules. Beaucoup reste à faire aussi, pour une extension à d'autres types de données symboliques comme par exemple des histogrammes ou pour adapter la bijection aux lois d'un mélange de densité dans l'espace des données symboliques.

Références

Bertrand, P. et F. Goupil (2000). Descriptive statistics for symbolic data. In *In H.H. Bock, E. Diday (Eds) "Analysis of Symbolic Data"*, pp. 106–124. Springer-Verlag.

Brito, P. et A.P. Duarte Silva (2011). Modelling Interval Data with Normal and Skew-Normal Distributions. In *Journal of Applied Statistics (in press).*,

Diday, E. et M. Noirhomme (2008). *Symbolic Data Analysis and the SODAS software*. 457 pages. Wiley. ISBN 978-0-470-01883-5.

Diday, E. et M. Vrac (2005). Mixture decomposition of distributions by copulas in the symbolic data analysis framework. In *Discrete Applied Mathematics (DAM). Volume 147, Issue1*, pp. 27–41.

Le-Rademacher, J. et L. Billard (2011). Likelihood functions and some maximum likelihood estimators for symbolic data. *Journal of Statistical Planning and Inference 141 1593 1602*.

Soubdhan, T., R. Emilion, et R. Calif (2009). Classification of daily solar radiation distributions. In *Solar Energy 83*, pp. 1056–1063. Elsevier.

Summary

Symbolic Data Analysis is born from Classification as it aims is to consider classes as units. These classes are described by symbolic data (intervals, distributions and the like) in order to take care of the variability of the individuals inside the classes. This paper is devoted to the link between the density distribution of the symbolic data and the density distribution of the parameters of the models attached to each symbolic data.

Maximisation de la modularité pour la classification croisée de données binaires

Lazhar Labiod , Mohamed Nadif

LIPADE, Université Paris Descartes,
45 rue des Saints Pères 75006 Paris, France.
email: Prénom.Nom@parisdescartes.fr

Résumé. La mesure de modularité a été récemment proposée pour la sélection automatique du nombre de classes dans la classification des graphes. En effet, des valeurs élevées de la mesure de modularité ont une bonne corrélation avec le regroupement optimal. Dans ce travail, nous traitons le problème de la classification croisée pour les données binaires et nous proposons une mesure de modularité généralisée et une approximation spectrale de la matrice de modularité.

1 Introduction

Un tableau de données à double entrée $I \times J$ représenté par une matrice A de taille $N \times M$, constitue en général le point de départ de toute analyse, il est le support des observations relevées sur le phénomène que l'on cherche à analyser. Ici, l'ensemble I représente l'ensemble des objets (individus) étudié et l'ensemble J est formé des attributs descriptifs de I, sélectionnés en fonction du problème à traiter. Afin d'exploiter l'information contenue dans A, différentes stratégies sont envisageables. En général les méthodes de classification automatique exploitent des matrices carrées dérivées de A. Elles traitent d'abord soit, l'ensemble des variables décrivant le phénomène, à travers une matrice carrée croisant l'ensemble J avec lui-même, ou l'ensemble des objets à travers une matrice carrée croisant l'ensemble I avec lui même. La troisième et dernière catégorie de méthodes de classification sont celles qui exploitent directement le tableau de base A. Elles ont été introduites pour répondre à des préoccupations sensiblement différentes de celles qui exploitent des matrices carrées dérivées de A. Il s'agit de classifier simultanément les lignes et les colonnes de A pour mettre en évidence la nature de la correspondance sur le croisement des deux ensembles I et J.

Bien que de nombreuses méthodes de classification telles que la classification hiérarchique et les k-moyennes (k-means) cherchent à construire une partition optimale d'objets ou, parfois, des variables, il existe d'autres méthodes, connues sous le nom de méthodes de classification croisée (ou par blocs), qui considèrent les deux ensembles simultanément et réorganisent les données sous forme de de blocs homogènes. Ces dernières années, la classification croisée, également désignée par le *co-clustering* ou le *bi-clustering*, est devenue un enjeu important dans le contexte de l'exploration de données. Dans le domaine de text mining, (Dhillon, 2001) a proposé une méthode spectrale pour la classification croisée exploitant la dualité entre les lignes (documents) et les colonnes (mots). Dans l'analyse des données de biopuces, où les

données sont souvent présentées comme des matrices de niveaux d'expression de gènes dans différentes conditions, la classification croisée permet de regrouper dans des blocs homogènes des gènes et des conditions. Le problème de la classification croisée des données binaires a été également abordé de point de vue probabiliste en utilisant l'approche modèle de mélange (Govaert et Nadif, 2008). La mesure la modularité a été utilisé récemment pour la classification des graphes (Newman et Girvan, 2004). Dans ce papier, nous montrons comment 1) cette mesure modularité peut être étendue pour la classification croisée des données binaires et 2) elle peut être liée à la grande famille des méthodes spectrales.

2 Mesure de modularité généralisée

Considérons la division des données en g blocs disjoints où $g \geq 2$. Nous allons définir une matrice de partition de l'ensemble I, notée R de dimension $N \times g$ et une matrice de partition de l'ensemble J notée C de dimension $M \times g$. Chaque colonne est un vecteur d'indicatrices binaires, tels que $r_{ik} = 1$ si l'objet i appartient à la classe k et 0 sinon, et $c_{jk} = 1$ si l'attribut j appartient à la classe k et 0 sinon. Notez que la somme de chaque ligne est égale à l'unité et les matrices R et C satisfont les propriétés suivantes : $Trace(R^T R) = N$, $R^T R = G = diag(n_1,...,n_k,...,n_g)$ où n_k représente la cardinalité de la classe ligne R_k, et $Trace(C^T C) = M$, $C^T C = F = diag(m_1,...,m_k,...,m_g)$ où m_k représente la cardinalité de la classe colonne C_k. Lorsque $A = (a_{ij})$ est une matrice binaire, pour résoudre le problème de la classification croisée, nous proposons une mesure de modularité généralisée définie par :

$$Q_1(A,R,C) = \frac{1}{2|E|} \sum_{i,j=1}^{N,M} (a_{ij} - \frac{a_{i.}a_{.j}}{2|E|}) \sum_{k=1}^{g} r_{ik}c_{jk} = \frac{1}{2|E|} Trace[R^t(A-\delta)C]. \quad (1)$$

où $2|E| = \sum_{i,j} a_{ij} = a_{..}$ est le poids totale des liens et $a_{i.} = \sum_j a_{ij}$ - le degré de i et $a_{.j} = \sum_i a_{ij}$- le degré de j. $\delta = (\delta_{ij})$, avec $\delta_{ij} = \frac{a_{i.}a_{.j}}{a_{..}}$

L'expression (1) n'est pas équilibrée par les cardinalités des classes en lignes et en colonnes, nous proposons donc une nouvelle mesure que nous appelons modularité normalisée dont la fonction objective est donnée comme suit:

$$\tilde{\mathcal{Q}}_1(A,R,C) = Tr[G^{\frac{-1}{2}} R^t(A-\delta)CF^{\frac{-1}{2}}] = Trace[\tilde{R}^t(A-\delta)\tilde{C}]. \quad (2)$$

où $\tilde{R} = RG^{\frac{-1}{2}}$ et $\tilde{C} = CF^{\frac{-1}{2}}$. Il est facile de vérifier que $\tilde{R}$ et $\tilde{C}$ satisfont les contraintes d'orthogonalité $\tilde{R}^t\tilde{R} = I_g$ et $\tilde{C}^t\tilde{C} = I_g$. Par conséquent, la maximisation de la modularité normalisée est équivalente à la résolution du problème de maximisation suivant :

$$\max_{\tilde{R}^t\tilde{R}=I_g, \tilde{C}^t\tilde{C}=I_g} Trace[\tilde{R}(A-\delta)\tilde{C}^t]. \quad (3)$$

Ce problème d'optimisation peut être traité par des multiplicateurs de Lagrange, ici la solution que nous proposons est basée sur une approximation spectrale de la matrice de modularité. En effet, posons $D_r = diag(A\mathbb{1})$ et $D_c = diag(A^t\mathbb{1})$ (où $\mathbb{1}$ est le vecteur de dimension appropriée dont toutes ses valeurs valent 1). Nous pouvons alors montrer que la matrice de modularité $(A-\delta)$ peut être approximée par les $(g-1)$ vecteurs propres associés aux plus grande

valeurs proprex de la matrice pondérée $\tilde{A} = D_r^{\frac{-1}{2}} A D_c^{\frac{-1}{2}}$ moins le vecteur trivial (correspondant à la plus grande valeur propre $\lambda = 1$). Après quelques développements, nous obtenons l'approximation suivante : $A - \frac{D_r \mathbb{1}\mathbb{1}^t D_c}{a_{..}} \approx \sum_{k=1}^{g-1} \tilde{U}_k \lambda_k \tilde{V}_k^t$ où $\tilde{U}_k = D_r^{\frac{-1}{2}} U_k$ et $\tilde{V}_k = D_c^{\frac{-1}{2}} V_k$. Prenant $\delta = \frac{D_r \mathbb{1}\mathbb{1}^t D_c}{a_{..}}$, on peut approximer $(A - \delta)$ par $\sum_{k=1}^{g-1} \tilde{U}_k \lambda_k \tilde{V}_k^t$. Les principales étapes de l'algorithme spectral utilisé sont décrites ci-après.

Algorithm 1 SpecCo

Input: Matrice de données A, nombre de classes g

Output: Matrices de partitions R et C

2. Définir D_r et D_c comme étant des matrices diagonales $D_r = diag(A\mathbb{1})$ and $D_c = diag(A^t\mathbb{1})$

3. Trouver U,V les $(g-1)$ vecteurs propres à gauche et à droite de $\tilde{A} = D_r^{-\frac{1}{2}} A D_c^{-\frac{1}{2}}$

4. Construire les matrices $\tilde{U}$, $\tilde{V}$ et $Q = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix}$

5. Partitionner les lignes de Q en g classes en utilisant par exemple les $k$means

6. Affecter l'object i à la classe R_k si et seulement si la ligne correspondante Q_i de la matrice Q a été affectée à la classe R_k et affecter l'attribut j à la classe C_k si et seulement si la ligne correspondante Q_j de la matrice Q a été affectée à la classe C_k

3 Expérimentation et validation

Afin de pouvoir évaluer la qualité de la classification croisée obtenue, nous avons utilisé trois bases de données synthétiques de taille 500×300 (Data1: blocs bien séparés, Data2: blocs avec recouvrement et Data3: blocs deséquilibrés) chacune contient 3 classes en lignes et 3 classes en colonnes, générées suivant un mélange de lois de Bernoulli par blocs. Nous avons utilisé la mesure de modularité pour évaluer la qualité de la partition simultanée trouvée en faisant varié le nombre de classes g de 2 à 9. Nous illustrons dans les figures , à gauche les bases, au milieu la partition simultanée obtenue et à droite le comportement du critère de la modularité en fonction de g variant entre 2 et 9. On peut voir que notre méthode reconstitue efficacement tous les blocs (les co-clusters) et cela pour les trois bases. On peut voir également que le comportement du critère de la modularité montre que la valeur maximale de la modularité est en bonne corrélation avec le vrai nombre de classes.

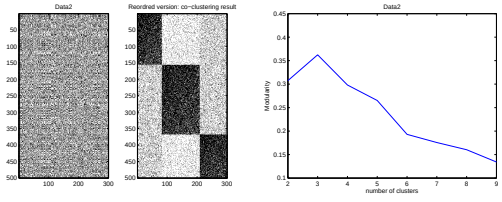


FIG. 1 – – *à gauche: Data1-au milieu: version réordonnée - à droite: Modularité versus le nombre classes*

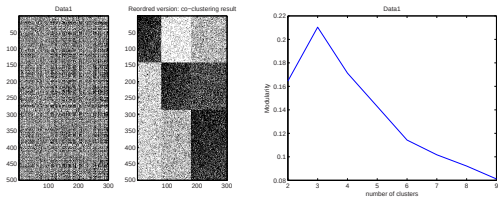


FIG. 2 – – *à gauche: Data2-au milieu: version réordonnée - à droite: Modularité versus le nombre classes*

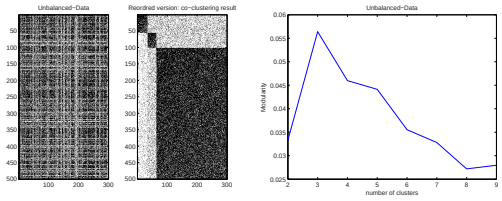


FIG. 3 – – *à gauche: Data3-au milieu: version réordonnée - à droite: Modularité versus le nombre classes*

4 Conclusion

Dans cet article, nous avons proposé un critère de modularité généralisée pour la classification croisée des données binaires. Nous avons traité la maximisation de ce critère par une procédure spectrale. Les résultats expérimentaux obtenus en utilisant différentes données simulées montrent l'efficacité de notre approche en terme de classification ainsi que celle de la mesure de modularité pour déterminer le bon nombre de blocs.

Remerciement : *Cette recherche a été financée par le projet ANR CLasSel ANR-08-EMER-002*

Références

Dhillon. I, "Co-clustering documents and words using bipartite spectral graph partitioning," *ACM SIGKDD International Conference*, San Francisco, USA, pp. 269-274, 2001.

Ding. C, Xiaofeng. H, Hongyuan. Z and Horst. S, "Self-aggregation in scaled principal component space,". Technical Report LBNL–49048. Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA, USA, 2001.

Bach. F. R and Jordan. M. I, "Learning spectral clustering, with application to speech separation," Journal of Machine Learning Research, pp. 1963-2001, 2006

Newman. M and Girvan. M, "Finding and evaluating community structure in networks," Physical Review E., 69, 026113. 2004.

Govaert. G and Nadif. M, "Block clustering with Bernoulli mixture models: Comparison of different approaches," *Computational Statistics and Data Analysis*, 52, pp. 233-3245, 2008.

La classification croisée pour la découverte des services Web

Malika Charrad*, Nadia Yacoubi Ayadi* Mohamed Ben Ahmed*

*Ecole Nationale des Sciences de l'Informatique
malika.charrad@riadi.rnu.tn,
mohamed.benahmed@riadi.rnu.tn

Résumé. Nous proposons dans ce papier d'appliquer la classification croisée à la découverte des services Web. Pour ce faire, nous présentons une variante plus rapide de l'algorithme CROKI2 de classification croisée des tableaux de contingence que nous comparons à la version originale de l'algorithme.

1 Introduction

Avec l'augmentation du nombre des services web disponibles en ligne et leur diversité, le besoin en méchanismes permettant l'organisation et la découverte des services pertinents s'est accru. Plusieurs travaux ont été menés afin d'améliorer les réponses des moteurs de recherche à une requête effectuée par un utilisateur à la recherche d'un service Web (météo, réservation,...). La majorité des travaux sont basés sur la classification simple des services afin de réduire l'espace de recherche en regroupant les services ayant les mêmes fonctionnalités dans des classes. Cette méthode permet de gagner en temps de réponse et de fournir à l'utilisateur, en réponse à sa requête, un ensemble de services au lieu d'un service unique. Dans le même contexte, nous proposons de classifier les services en se basant sur les annotations textuelles. Pour ce faire, nous avons recourt à la classification croisée afin d'identifier des biclasses composées par des services et des annotations textuelles qui sont fortement correlées. Outre l'avantage de la réduction de l'espace de recherche, une telle approche permet de détecter des relations implicites entre les annotations appartenant à la même biclasse. Dans ce papier, nous proposons une variante plus rapide de l'algorithme Croki2 (Govaert, 1983) de classification croisée des tableaux de contingence que nous appliquons sur une base de services extraits de Biocatalogue [1].

2 Classification croisée

La classification croisée consiste à la recherche simultanée de partitions sur l'ensemble de lignes et l'ensemble de colonnes d'un tableau de données. L'algorithme CROKI2 (classification **CRO**isée optimisant le **Khi2** du tableau de contingence) consiste à trouver une partition $P = (P_1, ..., P_K)$ l'ensemble des lignes X en K classes et une partition $Q = (Q_1, ..., Q_L)$ de l'ensemble de colonnes Y en L classes telles que le χ^2 de contingence du nouveau tableau de données soit maximum. La recherche des deux partitions (P, Q) sur les lignes et les colonnes

1. http ://www.biocatalogue.org/

peut se faire selon plusieurs stratégies. La stratégie proposée dans (Govaert, 1983) consiste à alterner l'optimisation sur les lignes puis sur les colonnes en utilisant l'algorithme des nuées dynamiques jusqu'à la convergence.

Pour chaque tirage aléatoire des partitions initiales
Démarrer d'une position initiale (P^0, Q^0, G^0)
Pour chaque itération
Optimisation sur les lignes
Etape de représentation
Etape d'affectation par les nuées dynamiques
Optimisation sur les colonnes
Etape de représentation
Etape d'affectation par les nuées dynamiques

Les étapes de représentation et d'affectation sont définies comme suit :

- Étape d'affectation : consiste à calculer pour chaque objet i de X (resp. j de Y) l'indice k^* (resp. l^*) de la classe d'affectation qui vérifie $k^* = argmin_{k=1...K} d_{\chi^2}(u_i, G_k)$, avec u_i est un vecteur ligne (resp. $l^* = argmin_{l=1...L} d_{\chi^2}(v_j, G_l)$, v_j est un vecteur colonne).
- Étape de représentation : consiste à calculer pour chaque classe k (resp. l) le prototype $G_k = (g_{k1}, ..., g_{kl}, ..., g_{kL})$ (resp. $G_l = (g_{1l}, ..., g_{kl}, ...g_{Kl})$), tel que $g_{kl} = \sum_{i \in P_k} a_{il} = \sum_{i \in P_k} \sum_{j \in Q_l} a_{ij}$ (resp. $g_{kl} = \sum_{j \in Q_l} a_{kj} = \sum_{j \in Q_l} \sum_{i \in P_k} a_{ij}$)

Nous proposons de combiner les deux étapes d'optimisation en une seule étape. Nous remplaçons l'algorithme des nuées dynamiques par une simple affectation de l'individu à la classe la plus proche. Une boucle d'itérations permet d'alterner les étapes d'affectation et de représentation sur les lignes et les colonnes plusieurs fois jusqu'à la convergence. La convergence est atteinte lorsqu'aucun individu sur les lignes et sur les colonnes ne change de classe d'appartenance. L'algorithme suivant explicite le déroulement de la nouvelle variante de l'algorithme Croki2.

Pour chaque tirage aléatoire des partitions initiales
Démarrer d'une position initiale (P^0, Q^0, G^0)
Etape d'affectation des lignes
Etape de représentation des lignes
Etape d'affectation des colonnes
Etape de représentation des colonnes

Sachant que l'algorithme Croki2 débute avec deux partitions initiales tirées au hasard sur les lignes et les colonnes (P^0, Q^0), les résultats obtenus, comme pour toutes les méthodes convergeant vers un optimum local, dépendent de ces partitions initiales. Il est donc nécessaire d'exécuter plusieurs fois l'algorithme afin d'assurer l'indépendance du résultat final des partitions initiales.

2.1 Calcul de la complexité des deux algorithmes

La comparaison entre les deux algorithmes est basée sur plusieurs critères (Charrad, 2010), dont nous retenons ici le critère de la complexité. Il importe de mentionner que les deux algo-

rithmes ont la même aptitude à déterminer la solution optimale.
L'algorithme Croki2 dans sa version originale est itératif à deux niveaux. Le premier niveau d'itérations assure la convergence de l'algorithme des nuées dynamiques. Soient, pour un couple de classes (K, L) donné, $niter11$ et $niter12$ le nombre d'itérations nécessaires pour la convergence de l'algorithme des nuées dynamiques sur les lignes et sur les colonnes respectivement, $niter2$ le nombre d'itérations nécessaires pour alterner l'optimisation sur les lignes et l'optimisation sur les colonnes. Comme l'opération la plus coûteuse dans l'algorithme Croki2 est le calcul des distances, nous estimons le nombre total de calcul de distances dans l'algorithme. Soit le tableau des données de dimensions (N, M). Pour un tirage donné, et pour chaque itération au niveau supérieur (allant de 1 à niter2), il est nécessaire d'effectuer $niter11 \times N \times K$ calculs de distances dans R^L et $niter12 \times M \times L$ calculs de distances dans R^K. Or niter11 et niter12 varient d'une itération à une autre dans la boucle supérieure. Ainsi, le nombre total d'opérations pour Croki2 dans sa version originale est donné par la formule suivante pour un tirage donné :

$$Croki2 : \sum_{i=1}^{niter2} (niter11_i \times N \times K \times L + niter12_i \times M \times L \times K)$$

$$Croki2 : K \times L \times \sum_{i=1}^{niter2} (niter11_i \times N + niter12_i \times M)$$

La nouvelle variante de l'algorithme Croki2 présente un seul niveau d'itérations. Soit $niter$ le nombre d'itérations nécessaires pour la convergence de l'algorithme pour un couple de classes (K, L) donné. Le nombre total d'opérations dans ce cas est :

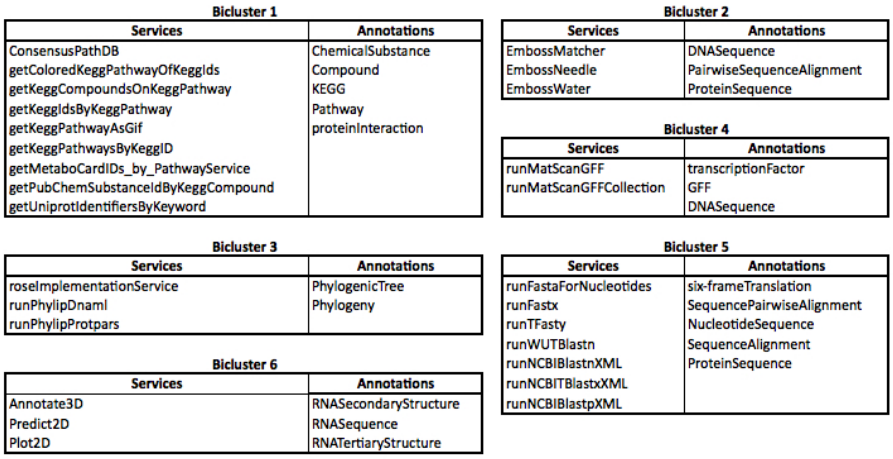
$$Croki2accelere : niter \times (N \times K \times L + M \times L \times K) = niter \times K \times L \times (N + M)$$

3 Application à la découverte des services Web

Notre base des expérimentations est composée des services Web extraits de Biocatalogue, un catalogue de services Web en biologie mettant à la disposition des utilisateurs 2054 services Web. Les services sont sémantiquement annotés à partir de leurs descriptions textuelles. Notre tableau de contingence est alors composé de 98 services décrits par 78 annotations. La construction de cette base est décrite en détail dans Ayadi et al. (2011). L'application de l'algorithme Croki2 amélioré, implémenté sous R, permet de découvrir un ensemble de biclasses dont les meilleures sont identifiées par les critères d'homogénéité et de significativité (Govaert, 1983). A titre d'exemple, les biclasses 2, 3, 4 et 6 sont les plus homogènes (H = 100%) et la biclasse 5 est la plus significative (R=10%). Chaque biclasse est composéé par un sous-ensemble de services et un sous-ensemble d'annotations qui sont fortement correlés.

4 Conclusion

Ce papier propose l'application de la classification croisée dans le contexte de découverte des services Web en bioinformatique. Cette approche repose sur une nouvelle variante de l'al-



Bicluster 1

Services	Annotations
ConsensusPathDB	ChemicalSubstance
getColoredKeggPathwayOfKeggIds	Compound
getKeggCompoundsOnKeggPathway	KEGG
getKeggIdsByKeggPathway	Pathway
getKeggPathwayAsGif	proteinInteraction
getKeggPathwaysByKeggID	
getMetaboCardIDs_by_PathwayService	
getPubChemSubstanceIdByKeggCompound	
getUniprotIdentifiersByKeyword	

Bicluster 2

Services	Annotations
EmbossMatcher	DNASequence
EmbossNeedle	PairwiseSequenceAlignment
EmbossWater	ProteinSequence

Bicluster 4

Services	Annotations
runMatScanGFF	transcriptionFactor
runMatScanGFFCollection	GFF
	DNASequence

Bicluster 3

Services	Annotations
roseImplementationService	PhylogenicTree
runPhylipDnaml	Phylogeny
runPhylipProtpars	

Bicluster 5

Services	Annotations
runFastaForNucleotides	six-frameTranslation
runFastx	SequencePairwiseAlignment
runTFasty	NucleotideSequence
runWUTBlastn	SequenceAlignment
runNCBIBlastnXML	ProteinSequence
runNCBITBlastxXML	
runNCBIBlastpXML	

Bicluster 6

Services	Annotations
Annotate3D	RNASecondaryStructure
Predict2D	RNASequence
Plot2D	RNATertiaryStructure

FIG. 1 – *Exemple de biclasses*

gorithme CROKI2 dont on a montré l'avantage en terme de rapidité. L'application de cet algorithme nous a permis de détecter des biclasses de services ayant les mêmes fonctionnalités. Comme perspective de ce travail, nous envisageons d'affiner cette classification en introduisant les critères non fonctionnels QOS "Quality of Services" (disponibilité, performance, temps de réponse...) dans la classification des services afin de permettre le classement (Ranking) des services dans chaque biclasse.

Références

Ayadi, N., M. Charrad, S. Amdouni, et M. B. Ahmed (2011). A framework for resource annotation and classification in bioinformatics. *4th International Workshop on Resource Discovery, 29 mai-2 juin, Heraklion, Grèce*.

Charrad, M. (2010). *Une approche générique pour l'analyse croisant contenu et usage des sites Web par des méthodes de bipartitionnement*. Thèse de doctorat en informatique, Conservatoire National des Arts et Métiers, Paris, France.

Govaert, G. (1983). Classification croisée. *Thèse de doctorat d'état, Paris*, 463–473.

Summary

In this paper, we propose to apply biclustering for web services discovery. A number of algorithms that perform simultaneous clustering on rows and columns of a matrix have been proposed to date. The goal of simultaneous clustering is to find sub-matrices, which are subgroups of rows and subgroups of columns that exhibit a high correlation. The current paper considers an acceleration of the Croki2 algorithm proposed for contingency table.

Un cadre de factorisation non négative pour la classification croisée

Lazhar Labiod, Mohamed Nadif

LIPADE, Université Paris Descartes,
45 rue des Saints Pères 75006 Paris, France.
email: Prénom.Nom@parisdescartes.fr

Résumé. Nous formalisons le problème de la classification croisée dans le cadre de la factorisation de matrices non négatives (NMF). A partir de la fonction objective optimisée par un double k-means, nous en proposons une nouvelle formulation. Pour optimiser cette dernière, nous développons deux algorithmes basés sur deux règles multiplicatives de mise à jour. En particulier, nous montrons que l'optimisation du critère du double k-means est équivalente à celle d'un critère algébrique de type NMF sous certaines contraintes appropriées. Des expériences numériques montrent l'intérêt de cette approche.

1 Introduction

Même si le problème de la classification croisée n'est pas l'objectif principal de la factorisation de matrices non négatives, cette approche a attiré l'attention de nombreux auteurs et particulièment pour la classification des documents. Différents algorithmes basés sur la tri-factorisation d'une matrice non négative ont été proposés. Étant donnée une matrice non négative A, la tri-factorisation consiste en la recherche d'une décomposition en trois facteurs (matrices), de type USV^T, tout en respectant la contrainte de non négativité de ces matrices. Les matrices U et V jouent les rôles de matrices de classification des lignes et des colonnes, chaque valeur de U et V correspond au degré d'appartenance d'une ligne ou une colonne à une classe donnée. La matrice S permet d'absorber les différences d'échelle de U, V et A. Tous les algorithmes proposés sont itératifs, et peuvent être différenciés par les règles de mise à jour des trois matrices, ces différences sont le résultat de la méthode d'optimisation choisie ou des contraintes supplémentaires imposées sur les trois matrices.

Dans ce papier, nous proposons un nouveau cadre pour la classification croisée fondé sur une formulation de type NMF. Nous considérons que la matrice rectangulaire de données non négatives peut être factorisée en deux facteurs $\mathbf{R}$ et $\mathbf{C}$ au lieu de trois. L'approche proposée optimise une formulation relaxée du critère du double k-means dans un style NMF. Celle-ci sera appelée DNMF (Double non negative matrix factorization) et ODNMF lorsque les contraintes d'orthogonalité sur $\mathbf{R}$ et $\mathbf{C}$ sont considérées. Nous développerons deux nouveaux algorithmes de classification croisée pour les données non négatives basés sur de deux règles multiplicatives de mise à jour.

2 Classification croisée: critère et algorithme

Étant donnée une matrice $A = (a_{ij}) \in \mathcal{R}^{M \times N}$, le but de la classification croisée est de trouver simultanément une partition en K classes $P = \{P_1, \ldots, P_K\}$ de l'ensemble des lignes $I = \{1, \ldots, N\}$ et une partition $Q = \{Q_1, \ldots, Q_L\}$ en L classes de l'ensemble des colonnes $J = \{1, \ldots, M\}$. Les deux partitions P et Q induisent naturellement et respectivement des matrices de classification $R = (r_{ik}) \in \{0,1\}^{N \times K}$ et $C = (c_{j\ell}) \in \{0,1\}^{M \times L}$; $r_{ik} = 1$ (resp. $c_{j\ell} = 1$) , si la ligne $\mathbf{a}_i \in P_k$ (resp. si la colonne $\mathbf{a}^j \in Q_\ell$), et 0 sinon. La réorganisation des lignes et des colonnes suivant P et Q révèle une structure de blocs homogènes. Chaque bloc $A_{k\ell}$ est donc défini par $\{(a_{ij}) | r_{ik} c_{j\ell} a_{ij} = 1\}$. D'autre part, nous notons $S = (s_{k\ell}) \in \mathcal{R}^{K \times L}$ jouant le rôle de représentant de taille réduite de A.

La détection des blocs homogènes en A peut être obtenue par la recherche des trois matrices R,C et S en minimisant $\mathcal{J}(A, RSC^T) = ||A - RSC^T||^2$. Le terme RSC^T caractérise l'information de A qui peut être décrite par une structure de classes. Le problème de classification peut ainsi vu comme un problème d'approximation matricielle où l'objectif est de minimiser l'erreur d'approximation entre les données d'origine A et la matrice reconstruite sur la base de structures de classes. Notons que cette formulation matricielle peut prendre la forme suivante $\mathcal{J}(A, RSC^T) = \sum_{i,j,k,\ell} r_{ik} c_{j\ell} (a_{ij} - s_{k\ell})^2$. Avec P_k, Q_ℓ fixées, il est facile de vérifier que le terme général de S optimale est obtenu par $s_{k\ell} = \frac{\sum_{i,j,k,\ell} r_{ik} c_{j\ell} a_{ij}}{r_k c_\ell}$ où, $r_k = |P_k|$; $c_\ell = |Q_\ell|$ ($s_{k\ell}$ est le barycentre de $A_{k\ell}$.

3 Un cadre NMF pour la classification croisée

En considérant le critère du double k-means comme une factorisation de la matrice A en un produit de matrices de faible rang, on peut formuler les contraintes à imposer à la formulation NMF. Dans le cadre du double kmeans, la fonction objective à minimiser est la distance au carrée entre chaque ligne (chaque colonne) du centre. soit $D_r^{-1} \in \mathcal{R}^{K \times K}$ et $D_c^{-1} \in \mathcal{R}^{L \times L}$ deux matrices diagonales définies par $D_r^{-1} = Diag(r_1^{-1}, \ldots, r_K^{-1})$ et $D_c^{-1} = Diag(c_1^{-1}, \ldots, c_L^{-1})$. En utilisant les matrices D_r,D_c,A,R et C, la matrice de représentation S s'crit : $S = D_r^{-1} R^T A C D_c^{-1}$. Injectons S dans la fonction objective $\mathcal{J}(A, RSC^T)$, l'expression à optimiser devient $||A - \mathbf{R}\mathbf{R}^T A \mathbf{C}\mathbf{C}^T||^2$, où $\mathbf{R} = RD_r^{-0.5}$ et $\mathbf{C} = CD_c^{-0.5}$. Notons que cette formulation est valable même si A n'est pas non négative. D'autre part, il est facile de vérifier que l'approximation $\mathbf{R}\mathbf{R}^T A \mathbf{C}\mathbf{C}^T$ de A est formée par la même valeur dans chaque bloc $A_{k\ell}$. Plus précisément, la matrice $\mathbf{R}^T A \mathbf{C}$ joue le rôle d'un résumé de A et absorbe les différences d'échelle de A, $\mathbf{R}$ et $\mathbf{C}$. Enfin les matrices $\mathbf{R}\mathbf{R}^T A$, $A\mathbf{C}\mathbf{C}^T$ donnent respectivement les vecteurs des moyennes des classes en ligne et en colonne.

3.1 Une nouvelle formulation et Propriétés de $\mathbf{R}$ et $\mathbf{C}$

Le problème de la classification croisée peut être reformulé comme la recherche de $\mathbf{R}$ et $\mathbf{C}$ minimisant $||A - \mathbf{R}\mathbf{R}^T A \mathbf{C}\mathbf{C}^T||^2$. Le calcul de $\mathbf{R}$ et $\mathbf{C}$ est difficile et nécessite un algorithme itératif. Ensuite, et contrairement au double k-means, dans ce qui suit, nous proposons une optimisation continue sous contraintes appropriées générées par des propriétés de $\mathbf{R}$ et $\mathbf{C}$: non négativité, orthonormalité, orthogonalité, bi-stochasticité et idempotence. A partir de ces contraintes, le critère du double k-means peut être reformulé de différentes manières. Ci-après,

nous nous concentrons sur deux formulations : 1) $\text{argmin}_{\mathbf{R},\mathbf{C}\geq 0} ||A - \mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||$ et 2) $\text{argmin}_{\mathbf{R},\mathbf{C}\geq 0} ||A - \mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||^2$ s.t. $\mathbf{R}^T\mathbf{R} = I_K$, $\mathbf{C}^T\mathbf{C} = I_L$. Dans le reste de ce papier, nous allons seulement nous focaliser sur le cas de $A \geq 0$. Nous appellerons la formulation (1) sans contraintes d'orthogonalité DNMF, et la formulation (2) avec les contraintes d'orthogonalité ODNMF. Nous déduirons des règles multiplicatives de mise à jour pour DNMF en utilisant les conditions de Karush-Kuhn-Tucker (KKT). Pour ODMNF, les règles de mise à jour seront dérivées en exploitant le vrai gradient sur les variétés de Stiefel [Edelman *et al.*, 1998].

3.2 Formulations DNMF et ODNM

Dans ce paragraphe nous considérons juste la contrainte de non négativité (DNMF), la fonction objective devient $\text{argmin}_{\mathbf{R},\mathbf{C}\geq 0} ||A - \mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||^2$. Appliquant la théorie d'optimisation standard et les conditions KKT pour trouver les minima, nous introduisons la fonction de Lagrange $\mathcal{L} = ||A - \mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||^2 - Trace(\Lambda\mathbf{R}^T) - Trace(\Gamma\mathbf{C}^T)$ où les matrices Λ et Γ sont les multiplicateurs de Lagrange introduits pour imposer la contrainte de non négativité respectivement sur $\mathbf{R}$ et $\mathbf{C}$. Prenons, $X_{\mathbf{C}} = A\mathbf{C}\mathbf{C}^T$ et $X_{\mathbf{R}} = \mathbf{R}\mathbf{R}^T A$, cela conduit aux règles de mise à jour suivantes :

$$\mathbf{R} \leftarrow \mathbf{R} \odot \frac{2AX_{\mathbf{C}}^T\mathbf{R}}{\mathbf{R}\mathbf{R}^T X_{\mathbf{C}} X_{\mathbf{C}}^T\mathbf{R} + X_{\mathbf{C}} X_{\mathbf{C}}^T\mathbf{R}\mathbf{R}^T\mathbf{R}}, \quad (1)$$

$$\mathbf{C} \leftarrow \mathbf{C} \odot \frac{2X_{\mathbf{R}}^T A\mathbf{C}}{\mathbf{C}\mathbf{C}^T X_{\mathbf{R}} X_{\mathbf{R}}^T\mathbf{C} + X_{\mathbf{R}} X_{\mathbf{R}}^T\mathbf{C}\mathbf{C}^T\mathbf{C}}. \quad (2)$$

Nous dérivons un algorithme pour calculer la relaxation non négative. L'algorithme a les étapes classiques d'une méthode type NMF. Compte tenu d'une solution existante ou d'une estimation initiale, nous améliorons itérativement la solution en mettant à jour les facteurs avec les règles (1) et (2). Pour prouver la convergence de notre algorithme, on peut utiliser le même principe que Lee et Seung [Lee and Seung, 2001] de l'approche de la fonction auxiliaire pour atteindre cet objectif.

Pour dériver les règles multiplicatives de mise à jour sous les contraintes d'orthogonalité sur $\mathbf{R}$ et $\mathbf{C}$ (ODNMF), nous calculons le vrai gradient (ou gradient naturel) sur les variétés de Stiefel [Edelman *et al.*, 1998]. Avec les mêmes notations utilisées dans [Yoo and Choi, 2010], nous obtenons les règles de mise à jour suivantes :
$\mathbf{R} \leftarrow \mathbf{R} \odot \frac{A\mathbf{C}\mathbf{C}^T A^T\mathbf{R}}{\mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T A^T\mathbf{R}}$ et $\mathbf{C} \leftarrow \mathbf{C} \odot \frac{A^T\mathbf{R}\mathbf{R}^T A\mathbf{C}}{\mathbf{C}\mathbf{C}^T A^T\mathbf{R}\mathbf{R}^T A\mathbf{C}}$.

4 Expérimentation

Nous avons utilisé 3 bases dont 2 extraites de Classic3 [Dhillon *et al.*, 2003] qui est une base composée de 3 classes Medline, Cisia, Cranfield. La première Classic30 compte 30 documents tirés au hasard de Classic3 et décrits par 1000 mots. La seconde Classic150 compte par contre 150 documents décrits par 3652 mots. La dernière NG2 extraite de 20-Newsgroup est composée de 500 documents concernant 2 classes talk.politics.mideast and talk.politics.misc. Sur ces bases normalisées en utilisant TF-IDF, nous avons comparé DNMF et ODNM à des méthodes existantes basées sur la tri-factorisation NBVD [Long *et al.*, 2005], ONM3F [Yoo and Choi, 2010], ONMTF [Ding *et al.*, 2006]. Pour évaluer la performance nous avons utilisé

le taux de bons classement (Acc) et l'information mutuelle normalisée (NMI). Les principaux points soulevés par les expériences en termes Acc et NMI (Table 1) sont les suivants. Les algorithmes ODNMF, ONM3F et ONMTF surpassent DNMF et NBVD, puis nous notons l'importance de la contrainte d'orthogonalité, ODNMF apparaît préféreble à ONM3F et ONMTF tandis DNMF est souvent supérieur NBVD.

TAB. 1 – – *Evaluation sur Classic30* $(K = 3, L = 10)$, *Classic150* $(K = 3, L = 10)$ *and NG2 with* $(K = 2, L = 10)$.

dataset	performance measure	DNMF	ODNMF	ONM3F	ONMTF	NBVD
Classic30	Acc	96.67	**100**	**100**	**100**	96.67
	NMI	89.97	**100**	**100**	**100**	89.97
Classic150	Acc	98.66	98.66	**99.33**	98.66	98.66
	NMI	94.04	94.04	**97.02**	94.04	94.04
NG2	Acc	77.6	**86.2**	74.6	74.2	77.4
	NMI	19.03	**43.47**	18.27	16.03	23.31

5 Conclusion

Nous avons montré que le crièrte du double k-means peut être formulé par un problème d'optimisation de la fonction objective sous contraintes appropriées et générées par les propriétés de deux facteurs $\mathbf{R}$ et $\mathbf{C}$. Nous avons développé deux algorithmes de style NMF en utilisant les conditions KKT pour DMNF et en exploitant directement l'information du vrai gradient sur les variétés de Stiefel pour ODNMF.

Remerciement : *Cette recherche a été financée par le projet ANR CLasSel ANR-08-EMER-002*

Références

I Dhillon, S. Mallela, and D. S. Modha. Information-theoretic coclustering. In *Proceedings of KDD'03*, pages 89–98. KDD'03, September 2003.

C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of KDD'06*, pages 635–640, Philadelphia, PA, September 2006. KDD'06.

A. Edelman, T. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal of Matrix Analysis and Application*, 20(2):303–353, 1998.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems NIPS*, 13 MIT Press.:303–353, 2001.

B. Long, Z. Zhang, and Ph. S. Yu. Co-clustering by value decomposition. In *Procedings of KDD'05*, pages 635–640. KDD'05, September 2005.

Jiho Yoo and Seungjin Choi. Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. *Information Processing and Management*, 46-Issue 5:559–570, 2010.

Summary

We embed the co-clustering in the NMF framework and we derive from the double k-means objective function a new formulation of the criterion. To optimize it, we develop two algorithms based on two multiplicative update rules.

Critères robustes de sélection de variables pour le modèle linéaire *via* l'estimation de coût

Aurélie Boisbunon*, Stéphane Canu*
Dominique Fourdrinier*,**

*Université de Rouen et INSA de Rouen
Avenue de l'Université - BP 12 - 76801 Saint-Étienne-du-Rouvray Cedex
aurelie.boisbunon}@univ-rouen.fr
stephane.canu@insa-rouen.fr
dominique.fourdrinier@univ-rouen.fr
**Cornell University, Department of Statistical Science
1176 Comstock Hall, Ithaca, New York 14853-9801

Résumé. Dans cette note [1], nous proposons d'adopter une approche décisionnelle de type estimation de coût en vue de la sélection de variables dans le modèle de régression linéaire. Cette procédure peut être appliquée dans un contexte distributionnel plus général que le modèle gaussien : le cadre des lois à symétrie sphérique, autorisant la dépendance entre les composantes du vecteur d'erreur et une robustesse théorique, que n'ont pas la plupart des méthodes classiques. Nous nous intéressons ici à l'estimateur de seuillage ferme. Outre un biais modéré, il a l'avantage de fournir un chemin de régularisation. Nous étudions les performances de sélection de notre critère au travers de simulations pour deux exemples de distribution des erreurs et comparons nos résultats à AIC et BIC.

1 Introduction

Nous considérons le modèle linéaire suivant :

$$Y = X\beta + \varepsilon, \tag{1}$$

où Y est un vecteur aléatoire dans $\mathbb{R}^n$, X est la matrice des données contenant p vecteurs de $\mathbb{R}^n$ orthogonaux, $p < n$, β est le vecteur inconnu des coefficients de la régression, et ε est le vecteur centré des erreurs dans $\mathbb{R}^n$. Lorsque les dimensions du modèle sont grandes, on fait l'hypothèse qu'un nombre réduit des variables de X ont une influence significative sur la variable d'étude Y. De manière équivalente, on suppose qu'il existe un sous-ensemble $I \subset \{1, \ldots, p\}$ tel que $\beta_i = 0$ pour tout $i \notin I$. L'estimateur classique des moindres carrés $\hat{\beta}^{MC} = (X^T X)^{-1} X^T Y$ est alors peu approprié de par son instabilité et son manque d'interprétabilité. Parmi les méthodes parcimonieuses proposées dans ce contexte, le *Least Absolute Shrinkage and Selection*

1. Ce travail a été réalisé dans le cadre du projet ANR ClasSel 08-EMER-002.
http://www.hds.utc.fr/classel/doku.php?id=fr:accueil

Operator (*lasso*), développé par Donoho et Johnstone (1994) et Tibshirani (1996), a rencontré un franc succès. Lorsque X est orthogonal, il est défini, $\forall i \in \{1, \ldots, p\}$, par

$$\hat{\beta}_i^{lasso} = (\hat{\beta}_i^{MC} - \lambda sgn(\hat{\beta}_i^{MC}))\mathbb{1}_{|\hat{\beta}_i^{MC}|>\lambda}, \tag{2}$$

où $\lambda > 0$ est une constante, $sgn(x) = x/|x|$ quand $x \neq 0$ et $sgn(0) = 0$, $\mathbb{1}_{x\in A} = 1$ sur $\{x \in A\}$ et 0 sinon, et $|.|$ représente la valeur absolue. Cet estimateur de β est linéaire par morceaux en Y, il est parcimonieux et propose un chemin de régularisation avec un nombre de sous-ensembles possibles de l'ordre de p. La sélection de variables se fait alors par le choix de l'hyperparamètre λ qui règle le niveau de parcimonie du modèle. Cependant, en raison de son grand biais, la consistance en sélection avec des critères de type erreur de prédiction n'est pas garantie (voir par exemple Leng et al. (2006)). C'est pourquoi nous lui préférons l'estimateur de seuillage ferme (*firm shrinkage*) proposé par Gao et Bruce (1997) et défini, $\forall i \in \{1, \ldots, p\}$, par

$$\hat{\beta}_i^{SF} = \mu(\hat{\beta}_i^{MC} - \lambda sgn(\hat{\beta}_i^{MC}))/(\mu - \lambda)\mathbb{1}_{\lambda<|\hat{\beta}_i^{MC}|\leq\mu} + \hat{\beta}_i^{MC}\mathbb{1}_{|\hat{\beta}_i^{MC}|>\mu}, \tag{3}$$

où $\mu > \lambda$ définit le passage du lasso aux moindres carrés. Le seuillage ferme réduit fortement le biais du lasso tout en gardant sa sélection. Pour éviter l'estimation d'un second hyperparamètre, nous considérerons le cas où $\mu = 2\lambda$ correspondant au seuillage moyen (*mid-threshold*) proposé par Walden et al. (1995).

Les critères les plus courants dans la littérature pour déterminer la valeur optimale de λ sont la validation croisée VC (Stone, 1974), le critère d'information d'Akaike AIC (Akaike, 1973), et le critère d'information bayésien BIC (Schwarz, 1978). Dans un contexte de grandes dimensions, VC est peu recommandé car trop coûteux. Par ailleurs, AIC a tendance à sélectionner des modèles trop complexes alors que BIC choisit des modèles trop simples.

Nous proposons d'utiliser une procédure basée sur l'estimation du coût quadratique de l'estimateur $\hat{\beta}^{SF}$ en (3). Cette procédure est applicable dans un cadre distributionnel beaucoup plus large que le cadre gaussien usuel : la famille des lois à symétrie sphérique (voir Kelker, 1970). Ces lois conservent la propriété de symétrie autour d'une moyenne présente dans le cas gaussien, permettant des calculs aisés, tout en ajoutant de la dépendance entre ses éléments. Par ailleurs, notre procédure ne dépend pas directement de la forme de la distribution des erreurs ε, qui n'est pas nécessairement spécifiée. Elle est donc robuste en ce sens, et nous souhaitons étudier l'influence de cette robustesse sur les résultats par rapport aux méthodes classiques.

2 Sélection via l'estimation du coût

Nos résultats sont développés pour la forme canonique du modèle linéaire. Nous exposons brièvement cette notation dans la section 2.1 avant de définir plus précisément notre procédure.

2.1 Forme canonique

La forme canonique du modèle linéaire (1) a été initiée par Scheffé (1959), développée par Zyskind (1967) et reprise par de nombreux auteurs comme Brandwein et Strawderman (1991). Elle consiste en une transformation orthogonale du vecteur Y, soit

$$Y \quad \xrightarrow{G} \quad \begin{pmatrix} G_1 Y \\ G_2 Y \end{pmatrix} = \begin{pmatrix} Z \\ U \end{pmatrix} = \begin{pmatrix} \theta \\ 0 \end{pmatrix} + G\varepsilon,$$

où la matrice $G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}$ est orthogonale et telle que les vecteurs lignes de G_1 engendrent le même sous-espace que les vecteurs colonnes de X, et où l'on pose $Z = G_1 Y$, $U = G_2 Y$, et $\theta = G_1 X\beta$. On cherche maintenant à estimer le vecteur $\theta \in \mathbb{R}^p$. En pratique, la matrice G peut être obtenue par la transposée Q^T, où Q est issu de la décomposition QR de X. Sous la forme canonique, l'estimateur des moindres carrés de θ se réduit à $\varphi_0(Z) = Z$, et l'estimateur de seuillage ferme devient, $\forall i \in \{1, \ldots, p\}$,

$$\varphi_i^{FS(Z)} = \frac{\mu}{\mu - \lambda}(Z_i - \lambda sgn(Z_i))\mathbb{1}_{\lambda < |Z_i| \leq \mu} + Z_i \mathbb{1}_{|Z_i| > \mu}. \tag{4}$$

2.2 Estimateur du coût quadratique pour le seuillage ferme

Le principe de notre procédure est le suivant. Étant donné φ_λ un estimateur de θ dont la parcimonie est réglée par λ, une fonction de coût appropriée $L(\varphi_\lambda, \theta)$ permet d'évaluer, outre la qualité de φ_λ, la proximité de la sélection au "bon" modèle, correspondant au sous-ensemble I des variables significatives, défini en introduction. Cependant, θ étant inconnu, ce coût est inaccessible. C'est pourquoi nous proposons de l'estimer à partir des observations et d'utiliser cette estimation pour l'évaluation du bon modèle : le minimum de cette estimation correspond à notre choix pour l'hyperparamètre λ.

Un choix courant pour sa simplicité est le coût quadratique $L(\varphi_\lambda, \theta) = ||\varphi_\lambda - \theta||^2$, où $||.||$ désigne la norme euclidienne. L'estimateur φ_λ est ici l'estimateur de seuillage ferme défini en (4). Un candidat naturel à l'estimation du coût $L(\varphi^{SF}, \theta)$ est l'estimateur sans biais donné par

$$\delta_0(Z, U) = \left(2k - p + 2\frac{\lambda}{\mu - \lambda} l\right) \frac{||U||^2}{n - p} + ||\varphi^{SF} - Z||^2, \tag{5}$$

où $k = \text{Card}\{i \setminus |Z_i| > \lambda\}$ est le nombre d'éléments sélectionnés, $l = \text{Card}\{i \setminus \lambda \leq |Z_i| < \mu\}$. Le caractère sans biais de $\delta_0(Z, U)$ provient de ce que son espérance est égale [2] au risque de φ^{SF}. Notons que $\delta_0(Z, U)$ est équivalent, en tant que sélecteur, à AIC dans le cas où la distribution des erreurs ε est gaussienne.

3 Simulations

Pour les simulations présentées ici, les données consistent en $p = 250$ variables pour $n = 6000$ observations. Le vecteur β est composé de $(p - k)$ coefficients nuls et de k coefficients non nuls générés suivant une loi uniforme $\mathcal{U}_{[7.5;10]}$ sur l'intervalle $[7.5; 10]$, k variant de 10 à 240. Ce vecteur remplit les conditions pour lesquelles le vrai sous-ensemble I appartient au chemin de régularisation du lasso. Enfin, on génère les erreurs ε à partir d'une loi normale $\mathcal{N}(0, I_n)$ dans un premier temps, hypothèse habituelle, puis à partir d'une loi de Student n-variée à 4 degrés de liberté $t_n(4)$, qui est une distribution à symétrie sphérique à queue plus lourde que la loi normale. Pour chaque exemple de distribution et chaque exemple de vecteur β, nous générons 100 répliques d'erreurs afin d'étudier les variations de notre critère face à l'aléa. Les résultats sont comparés avec AIC et BIC.

2. Le critère δ_0 est d'ailleurs issu du développement du risque de φ^{SF} avec des identités de type Stein (voir Fourdrinier et Wells, 1995)

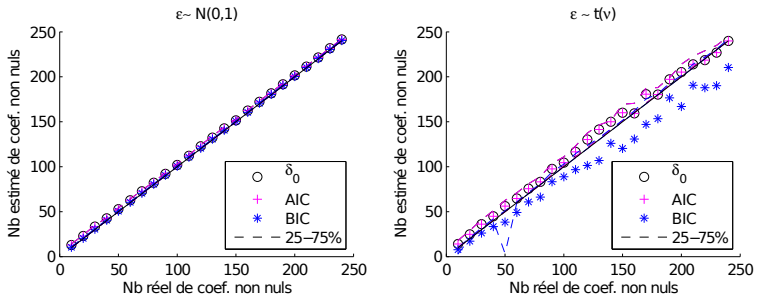


FIG. 1 – *Nombre de coefficients estimés non nuls en fonction du nombre de coefficients réels non nuls. Erreurs gaussiennes (gauche) et erreurs Student* $t(4)$ *(droite).*

La figure 1 présente le nombre moyen de coefficients estimés non nuls sur les 100 répliques en fonction du nombre de coefficients réels non nuls k. Les lignes pointillées correspondent aux premier et troisième quartiles. Dans le cas gaussien, correspondant aux hypothèses d'AIC et BIC, on remarque que les trois estimateurs se comportent aussi bien et sélectionnent le bon modèle presque à chaque fois. Dans le cas Student, on constate à nouveau l'équivalence entre AIC et δ_0 énoncée en 2.2, bien que nous ayions conservé la formule d'AIC sous l'hypothèse gaussienne tout au long de nos expériences. Ceci confirme la robustesse distributionnelle connue d'AIC dans le contexte de la régression linéaire où n est grand devant p et le vrai modèle appartient à l'ensemble sélectionné (voir Burnham et Anderson, 2002). Si notre approche s'avère équivalente pour d'autres lois sphériques, l'améliorer impliquera la considération de nouveaux estimateurs de coût (voir partie 4). Par ailleurs, les performances des deux estimateurs sont légèrement moins bonnes que pour le cas gaussien, tendant à sélectionner un modèle un peu trop grand. BIC, quant à lui, n'est très proche du bon modèle que dans les contextes très parcimonieux ($k \leq 40$). Ailleurs, il passe à côté d'une partie des variables significatives.

4 Discussion

Dans cette note, nous avons exposé une première approche de la mise en œuvre de l'estimation de coût comme sélecteur de variables. Nos simulations ont montré que, dans ce contexte, nos résultats sont meilleurs que BIC et comparables à AIC, même pour le cas sphérique non gaussien, alors que nous espérions une amélioration pour ce cas. Une étude plus approfondie des raisons de cette équivalence est en cours. Il semblerait que AIC puisse être lui aussi vu comme estimateur de coût, et serait donc valide pour la classe entière des lois à symétrie sphérique. Une perspective d'amélioration de notre procédure est l'utilisation de meilleurs estimateurs de coût. En effet, l'estimateur sans biais $\delta_0(Z, U)$ n'est sans doute pas le meilleur estimateur possible dans la mesure où une évaluation en terme de risque quadratique peut être mise en œuvre et montre que $\delta_0(Z, U)$ peut être amélioré. Des travaux en cours confirment cette appréciation. Nous souhaitons souligner que notre procédure est générale et applicable à d'autres fonctions de coût, ce qui permet d'intégrer des coûts de type 0-1 ou charnière dans un cadre de discrimination.

Références

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Second international symposium on information theory*, Volume 1, pp. 267–281. Akademiai Kiado.

Brandwein, A. et W. Strawderman (1991). Generalizations of james-stein estimators under spherical symmetry. *The Annals of Statistics 19*(3), 1639–1650.

Burnham, K. et D. Anderson (2002). *Model selection and multimodel inference : a practical information-theoretic approach*. Springer Verlag.

Donoho, D. et J. Johnstone (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika 81*(3), 425.

Fourdrinier, D. et M. Wells (1995). Estimation of a loss function for spherically symmetric distributions in the general linear model. *The Annals of Statistics 23*(2), 571–592.

Gao, H. et A. Bruce (1997). WaveShrink with firm shrinkage. *Statistica Sinica 7*, 855–874.

Kelker, D. (1970). Distribution theory of spherical distributions and a location-scale parameter generalization. *Sankhyā : The Indian Journal of Statistics, Series A 32*(4), 419–430.

Leng, C., Y. Lin, et G. Wahba (2006). A note on the lasso and related procedures in model selection. *Statistica Sinica 16*(4), 1273.

Scheffé, H. (1959). The analysis of variance, 1959. *New York*, 331–367.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics 6*(2), 461–464.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological) 36*(2), 111–147.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Walden, A., D. Percival, et E. Mccoy (1995). Spectrum estimation by wavelet thresholding of multitaper estimators.

Zyskind, G. (1967). On canonical forms, non-negative covariance matrices and best and simple least squares linear estimators in linear models. *The Annals of Mathematical Statistics 38*(4), 1092–1109.

Summary

In this note, we propose to adopt a loss estimation approach to variable selection in the linear regression model. This procedure can be applied to a distributional context more general than the usual Gaussian model: the spherically symmetric distribution framework, which implies both dependency between the error vector's componants and a theoretical robustness property at the same time, not shared by most of classical methods. Here we focus our interest on the Firm shrinkage estimator. In addition to a moderate bias, it has the advantage of providing a regularization path. We study the behaviour of our procedure for selecting the right model through simulations for two examples of the error distribution in comparison to AIC and BIC.

Un cadre général pour les mesures de co-similarité.

Clément Grimal *, Gilles Bisson **

*UJF-Grenoble 1/CNRS - Université de Grenoble
**CNRS - Université de Grenoble
LIG UMR 5217 / AMA team
{clement.grimal, gilles.bisson@imag.fr}

Résumé. Nous proposons ici une formulation générale de la notion de co-similarité qui permet d'effectuer des co-classifications conjointes d'objets et de descripteurs – par exemple des matrices documents/termes – en utilisant les algorithmes classiques de classification.

1 Contexte

En classification, lorsque ces caractéristiques décrivant une collection d'instances correspondent à un type de donnée homogène et qu'elles sont sémantiquement comparables, il est possible de les classifier au même titre que les instances. L'objectif de la co-classification (*co-clustering*) est de prendre en compte cette dualité afin de faire émerger automatiquement des regroupements plus pertinents : ainsi, dans le contexte des données textuelles, il est clair que la ressemblance entre documents dépend de la ressemblance entre les termes qui les composent, et non de leur seule identité, et réciproquement pour les termes.

Cette approche est largement étudiée dans le contexte de la *bioinformatique* et de la *fouille de textes*. Dans ce domaine, elle permet de surmonter le double problème de la faible densité des données (*sparsity*) et de la taille élevée du nombre des caractéristiques (*curse of dimensionality*). Parmi les nombreuses approches proposées, l'une des plus connue est l'analyse sémantique latente (LSA) introduite par Deerwester et al. (1990). Ces travaux repose sur le fait qu'un être humain utilise un vocabulaire varié pour décrire un même thème. Par exemple, si l'on considère un premier corpus contenant plusieurs co-occurrences des termes *océan* et *vagues* et un second contenant les termes *mer* et *vagues*, on peut inférer par transitivité que les termes *océan* et *mer* sont possiblement sémantiquement reliés. Cette association correspond à une co-occurrence du second ordre (une seule indirection) et peut être généralisée à des ordres supérieurs.

Nous avons développé une mesure nommée χ-`Sim` (Bisson et Hussain (2008); Hussain et al. (2010)) qui capture ces régularités par l'intermédiaire de la notion de *co-similarité*. L'idée est de construire de manière conjointe les deux matrices de similarité entre documents et mots, chacune prenant en compte durant le calcul les informations fournies par les autres. Ces matrices permettent ensuite de faire de la co-classification en utilisant des algorithme de classification standards tels les k-means, la CAH, ...

2 Notations utilisées

Les matrices (capitales) et les vecteurs (minuscules), sont en gras alors que les variables sont en italique.

- *Matrice de données :* $\mathbf{M}$ représente la matrice de données de n_a lignes et de n_b colonnes avec m_{ij} correspondant à l'intensité du lien entre l'objet représenté par la ligne i (a_i) et l'objet représenté par la colonne j (b_j). Nous utiliserons également une notation vectorielle pour les lignes $\mathbf{m_{i:}} = [m_{i1} \cdots m_{in_b}]$ et pour les colonnes $\mathbf{m_{:j}} = [m_{1j} \cdots m_{n_aj}]$. Dans la suite, nous nous réfèrerons à a_i quand nous nous intéresseront à l'objet i de type A, et nous utiliserons $\mathbf{m_{i:}}$ pour dénoter son vecteur.
- *Matrices de similarité :* $\mathbf{SA}$ et $\mathbf{SB}$ représentent respectivement les matrices (carrées et symétriques) de similarité pour les objets de type A (de taille $n_a \times n_a$) et les objets de type B (de taille $n_b \times n_b$). Ainsi $\forall i, j = 1..n_a, sa_{ij} \in [0, 1]$ et $\forall i, j = 1..n_b, sb_{ij} \in [0, 1]$.
- *Fonction de similarité :* la fonction générique $\mathrm{F_s}(\cdot, \cdot)$ prend en argument deux éléments de $\mathbf{M}$, m_{il} et m_{jn} et retourne une mesure de similarité entre ces deux éléments $\mathrm{F_s}(m_{il}, m_{jn})$.

3 Fondements de la mesure χ-Sim

Classiquement, la mesure de similarité (ou de distance) entre deux objets a_i et a_j est définie comme une fonction – notée ici $\mathrm{Sim}(a_i, a_j)$ – qui ne dépend que des « éléments » communs entre objets. On peut ajouter d'autres éléments, par exemple pour normaliser la valeur, mais fondamentalement l'idée reste la même, la conséquence étant que la similarité entre deux objets ne partageant aucune information est nulle :

$$\mathrm{Sim}(a_i, a_j) = \mathrm{F_s}(m_{i1}, m_{j1}) + \cdots + \mathrm{F_s}(m_{ic}, m_{jc}) \tag{1}$$

Maintenant, supposons que l'on dispose d'un matrice $\mathbf{SB}$ dont les éléments sont les mesures de similarité entre les objets représentés par les colonnes de la matrice de données (ici les mots des documents). Simultanément, introduisons, par analogie à la norme L_k (distance de Minkowski), la notion de *pseudo-norme k*. Ainsi, si $\mathbf{SB} = \mathbf{I}$ et $k = 1$, l'équation (1) peut être réécrite comme :

$$\mathrm{Sim}^k(a_i, a_j) = \sqrt[k]{\sum_{l=1}^{n_b} \left(\mathrm{F_s}(m_{il}, m_{jl})\right)^k \times sb_{ll}} \tag{2}$$

Maintenant, nous allons généraliser (2) afin de prendre en compte, non plus uniquement les éléments communs aux deux objets, mais également l'ensemble des paires d'éléments. De la sorte, nous devenons capable de « capturer » non seulement la similarité entre les éléments identiques mais aussi celle provenant d'éléments différents : dans le cas de corpus, on devient ainsi potentiellement capable de comparer des documents contenant des termes différents. Bien sûr, pour chaque paire d'éléments b_l et b_n qui ne sont pas directement partagés par a_i et a_j, nous pondérons leur contribution à la similarité $\mathrm{Sim}(a_i, a_j)$ par leur propre similarité normalisée sb_{ln}. Cette nouvelle similarité entre les deux objets a_i et a_j est définie dans l'équation (3) dans laquelle les termes dont $l = n$ sont ceux qui apparaissaient dans (2) :

$$\mathrm{Sim}^k(a_i, a_j) = \sqrt[k]{\sum_{l=1}^{n_b} \sum_{n=1}^{n_b} \left(\mathrm{F_s}\left(m_{il}, m_{jn}\right)\right)^k \times sb_{ln}} \tag{3}$$

En supposant, comme pour le cosinus, que $\mathrm{F_s}(m_{ij}, m_{jn})$ correspond au produit de ses deux arguments, i.e. $\mathrm{F_s}(m_{ij}, m_{jn}) = m_{il} \times m_{jn}$, nous pouvons réécrire (3) sous la forme du produit matriciel suivant :

$$\mathrm{Sim}^k(a_i, a_j) = \mathrm{Sim}^k(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \sqrt[k]{\left(\mathbf{m}_{i:}\right)^k \times \mathbf{SB} \times \left(\mathbf{m}_{j:}^{\mathrm{T}}\right)^k} \tag{4}$$

avec $\left(\mathbf{m}_{i:}\right)^k = \left[\left(m_{ij}\right)^k \cdots \left(m_{ic}\right)^k\right]$ et $\mathbf{m}_{j:}^{\mathrm{T}}$ représentant le vecteur transposé de $\mathbf{m}_{j:}$.

Finalement, nous pouvons introduire un terme de normalisation – noté $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:})$ – afin que toutes les mesures de similarité soient comprises dans l'intervalle $[0, 1]$. Nous obtenons alors l'équation ci-dessous (5), dans laquelle sr_{ij} représente un élément de la matrice $\mathbf{SA}$:

$$sa_{ij} = \frac{\sqrt[k]{\left(\mathbf{m}_{i:}\right)^k \times \mathbf{SB} \times \left(\mathbf{m}_{j:}^{\mathrm{T}}\right)^k}}{\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:})} \tag{5}$$

On peut remarquer que l'équation (5) généralise différentes mesures de similarité classiques :

- *L'indice de Jaccard* peut être obtenu pour les valeurs de paramètres suivantes : $k = 1$, $\mathbf{SB} = \mathbf{I}$ (la matrice identité), et $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \|\mathbf{m}_{i:}\|_1 + \|\mathbf{m}_{j:}\|_1 - \mathbf{m}_{i:}\mathbf{m}_{j:}^{\mathrm{T}}$
- *Le coefficient de Dice* correspond à $k = 1$, $\mathbf{SB} = 2\mathbf{I}$, et $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \|\mathbf{m}_{i:}\|_1 + \|\mathbf{m}_{j:}\|_1$
- *La mesure de similarité du cosinus généralisé (Qamar et Gaussier, 2009)* est obtenu lorsque $\mathbf{SB}$ est une matrice *semi-définie positive (SDP)* notée $\mathbf{A}$. Sous cette hypothèse, on peut donc définir le produit scalaire $< \mathbf{m}_{i:}, \mathbf{m}_{j:} >_{\mathbf{A}} = \mathbf{m}_{i:} \times A \times \mathbf{m}_{j:}^{\mathrm{T}}$, ainsi que la norme associée notée $\|\mathbf{m}_{i:}\|_{\mathbf{A}} = < \mathbf{m}_{i:}, \mathbf{m}_{i:} >_{\mathbf{A}}$. On définit alors $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \sqrt{\|\mathbf{m}_{i:}\|_{\mathbf{A}}} \times \sqrt{\|\mathbf{m}_{j:}\|_{\mathbf{A}}}$.

3.1 Fonction de normalisation de χ-Sim

Nous allons introduire un nouveau schéma de normalisation, que nous nommerons *pseudo-normalisation*, et qui s'inspire de la mesure de similarité du cosinus généralisé, en relaxant la contrainte sur la propriété SDP de la matrice $\mathbf{A}$ et en ajoutant le paramètre k de pseudo-norme. En utilisant le produit matriciel (4) introduit dans la section 3 nous définissons symétriquement les éléments des matrices $\mathbf{SA}$ et $\mathbf{SB}$ ainsi :

$$sa_{ij} = \frac{\mathrm{Sim}^k(\mathbf{m}_{i:}, \mathbf{m}_{j:})}{\sqrt{\mathrm{Sim}^k(\mathbf{m}_{i:}, \mathbf{m}_{i:})} \times \sqrt{\mathrm{Sim}^k(\mathbf{m}_{j:}, \mathbf{m}_{j:})}} \tag{6a}$$

$$sb_{ij} = \frac{\mathrm{Sim}^k(\mathbf{m}_{:i}, \mathbf{m}_{:j})}{\sqrt{\mathrm{Sim}^k(\mathbf{m}_{:i}, \mathbf{m}_{:i})} \times \sqrt{\mathrm{Sim}^k(\mathbf{m}_{:j}, \mathbf{m}_{:j})}} \tag{6b}$$

Les équations (6a) et (6b) définissent donc *un système d'équations linéaires*, dont les solutions correspondent aux (co-)similarité entre les deux types d'objets dont la relation est décrite par la matrice de données $\mathbf{M}$. Par conséquent, l'algorithme χ-Sim est basé sur une approche itérative, i.e. un calcul alterné des valeurs des matrices $\mathbf{SA}$ et $\mathbf{SB}$. La normalisation assure que $sa_{ii} = 1$ et que $sb_{ii} = 1$ sans garantir toutefois que les indices de similarité soient toujours $\leqslant 1$. Dans le cas de données textuelles, cela correspond à des problèmes de polysémies de termes. Considérons ainsi un corpus contenant, parmi plusieurs autres documents, les documents d_1 contenant le mot *orange* et d_2 contenant les mots *rouge* et *banane*. Supposons qu'à une itération quelconque la matrice $\mathbf{SB}$ indique que que la valeur de similarité entre *orange* et *rouge* est 1, celle entre *orange* et *banane* est 1 et celle entre *rouge* et *banane* est 0. Dès lors, en appliquant les formules précédentes, $\mathrm{Sim}^1(\mathrm{d}_1, \mathrm{d}_1) = 1$, $\mathrm{Sim}^1(\mathrm{d}_2, \mathrm{d}_2) = 2$ and $\mathrm{Sim}^1(\mathrm{d}_1, \mathrm{d}_2) = 2$ et donc $sr_{12} = \frac{2}{\sqrt{1 \times 2}} > 1$. Ici, la similarité entre ces deux documents est surestimée à cause de la nature polysémique du mot *orange* qui présenter une double analogie avec la couleur *red* et le fruit *banane*. Expérimentalement, on observe que les valeurs sr_{ij} and sc_{ij} reste la plupart du temps inférieures ou égales à 1.

Parallèlement, lorsque l'on affecte à k des valeurs inférieures à 1 comme suggéré par Aggarwal et al. (2001) pour la norme L_k dans le contexte d'espaces de grandes dimensions, on observe une amélioration des résultats sur des test de classification (cf. Hussain et al. (2010)). Il faut cependant noter que nous sommes dans une situation différente de la norme L_k car notre méthode ne définit pas un espace vectoriel normé. Si l'on examine le cas simple où $k = 1$, on a alors $\mathrm{Sim}^1(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \mathbf{m}_{i:} \times \mathbf{SB} \times \mathbf{m}_{i:}^{\mathrm{T}}$, ce qui correspond à la forme générale d'un produit scalaire, à la condition que $\mathbf{SB}$ soit symétrique et semi-défini positive (SDP). Malheureusement, cette condition n'est pas nécessairement vérifié du fait de notre schéma de normalisation [1]. Aussi, notre mesure de similarité est simplement une forme bilinéaire dans un espace préhilbertien « dégénéré », dans lequel notre mesure correspond au Cosinus.

Pour résoudre complètement ces problèmes, il est possible de projeter les matrices $\mathbf{SB}$ et $\mathbf{SA}$ à chaque itération sur l'espace des matrices SDP comme le propose (Qamar et Gaussier, 2009). Ainsi, nous garantirions que le nouvel espace engendré soit bien un espace préhilbertien. Cependant, on constate expérimentalement que l'ajout d'une telle étape ne permet pas d'améliorer significativement les résultats de notre approche, car les matrices de similarités sont déjà très proches de l'espace des matrices SDP.

3.2 Traiter le 'bruit' dans les matrices de similarité

Si l'on considère le graphe biparti associé à une matrice documents/termes, on peut aisément montrer (cf. Hussain et al. (2010)) que, à l'itération n de l'algorithme, l'élément sa_{ij} de la matrice de similarité des lignes est fonction du nombre de *chemins d'ordre* n qui existent entre les objets i et j. Cependant, dans les jeux de données textuelles, certains termes sont rares et/ou ne sont pas spécifiques d'une classe d'objets. Dans le cas de données textuelles cela peut correspondre à des mots soit mal-orthographié, soit qui apparaissent de manière fortuite dans une classe de documents ; par exemple, le fait de trouver dans un document qu'une personne est « ... une nouvelle étoile au firmament ... » ne rattache a priori en rien ce document à la catégorie « astronomie ». On peut alors considérer ces termes comme une sorte de *bruit*

1. Une condition nécessaire pour que $\mathbf{SB}$ soit SDP serait que $\forall i, j \in 1..c, |sb_{ij}| \leqslant \sqrt{sb_{ii} \times sb_{jj}} = 1$.

dans les données car itérations après itérations, ces termes permettent à l'algorithme d'établir de nouveaux chemins erronés entre les différentes familles d'objets. Ces chemins induisent des similarités très faibles mais ils sont nombreux, et nous pouvons faire l'hypothèse qu'ils peuvent brouiller les « vraies » similarités. En se basant sur cette observation, nous introduisons dans l'algorithme χ-Sim une étape de *seuillage* associée à une paramètre p qui a pour objectif de supprimer ces infomations. Concrètement il va s'agir, à chaque itération, de remettre à 0 les p % des plus petites valeurs des matrices de similarité $\mathbf{SA}$ et $\mathbf{SB}$.

3.3 Un algorithme générique pour χ-$\mathtt{Sim}_p^k$

Les équations (6a) et (6b) nous permettent de calculer les similarités entre deux lignes et entre deux colonnes. L'extension à toutes les paires de lignes et toutes les paires de colonnes peut être formulée sous la forme d'une simple multiplication matricielle. Nous avons besoin de définir ici $\mathbf{M}^{\circ_k} = \left((m_{ij})^k\right)_{i,j}$ qui représentent la mise à la puissance k de la matrice $\mathbf{M}$. L'algorithme générique est le suivant :

1. On initialise les matrices de similarité $\mathbf{SA}$ et $\mathbf{SB}$ avec la matrice identité $\mathbf{I}$. En effet, on considère que l'on a pas de connaissance *a priori*, et que donc seul la similarité entre un objet et lui-même vaut 1. On note ces matrices $\mathbf{SA}^{(0)}$ et $\mathbf{SB}^{(0)}$.
2. À l'itération t, on calcule la nouvelle matrice de similarité $\mathbf{SA}^{(t)}$ en utilisant la matrice $\mathbf{SB}^{(t-1)}$:

$$\mathbf{SA}^{(t)} = \mathbf{M}^{\circ_k} \times \mathbf{SB}^{(t-1)} \times \left(\mathbf{M}^{\circ_k}\right)^{\mathrm{T}} \text{ and } sa_{ij}^{(t)} \leftarrow \frac{\sqrt[k]{sa_{ij}^{(t)}}}{\sqrt[2k]{sa_{ii}^{(t)} \times sa_{jj}^{(t)}}} \tag{7}$$

   On fait la même chose pour la matrice de similarité $\mathbf{SB}^{(t)}$ en utilisant $\mathbf{SA}^{(t-1)}$:

$$\mathbf{SB}^{(t)} = \left(\mathbf{M}^{\circ_k}\right)^{\mathrm{T}} \times \mathbf{SA}^{(t-1)} \times \mathbf{M}^{\circ_k} \text{ and } sb_{ij}^{(t)} \leftarrow \frac{\sqrt[k]{sb_{ij}^{(t)}}}{\sqrt[2k]{sb_{ii}^{(t)} \times sb_{jj}^{(t)}}} \tag{8}$$

3. On fixe à 0 les p % des plus petites valeurs des matrices de similarité $\mathbf{SA}$ et $\mathbf{SB}$.
4. Les étapes 2 et 3 sont répétées t fois (un nombre d'itérations de 4 est une valeur raisonnable pour des données textuelles) pour mettre à jour itérativement $\mathbf{SA}^{(t)}$ et $\mathbf{SB}^{(t)}$.

Il est important de remarquer que même si χ-$\mathtt{Sim}_p^k$ calcule une mesure de similarité entre chaque paire d'objets en utilisant toutes les paires de composantes des vecteurs les représentant, la complexité de l'algorithme reste comparable aux mesures de similarité classiques comme celle du Cosinus. En supposant que, pour une matrice générale de taille $n \times n$, la complexité de la multiplication matricielle est de $\mathcal{O}(n^3)$ et que la complexité pour calculer $\mathbf{M}^{\circ_k}$ est de $\mathcal{O}(n^2)$, la complexité totale de χ-$\mathtt{Sim}_p^k$ est donnée par $\mathcal{O}(tn^3)$.

Références

Aggarwal, C. C., A. Hinneburg, et D. A. Keim (2001). On the surprising behavior of distance metrics in high dimensional space. In *Lecture Notes in Computer Science*, pp. 420–434. Springer.

Bisson, G. et F. Hussain (2008). Chi-sim : A new similarity measure for the co-clustering task. In *Proceedings of the Seventh ICMLA*, pp. 211–217. IEEE Computer Society.

Deerwester, S., S. T. Dumais, G. W. Furnas, Thomas, et R. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, 391–407.

Hussain, S. F., C. Grimal, et G. Bisson (2010). An improved co-similarity measure for document clustering. In *International Conference on Machine Learning and Applications*.

Qamar, A. M. et E. Gaussier (2009). Online and batch learning of generalized cosine similarities. In *Proceedings of the Ninth IEEE ICDM*, Washington, DC, USA, pp. 926–931. IEEE Computer Society.

Classification de grands ensembles de données par un algorithme stochastique moyennisé des k-noyaux

Jean-Marie Monnez*

*Institut Elie Cartan, UMR 7502, Nancy Université, CNRS, INRIA
BP 239, 54506 Vandoeuvre-lès-Nancy Cedex, France
Jean-Marie.Monnez@iecn.u-nancy.fr
http://www.iecn.u-nancy.fr

Résumé. Soit à classifier un grand ensemble de données pouvant être de grande dimension. On suppose que les vecteurs de données sont des observations indépendantes et identiquement distribuées d'un vecteur aléatoire Z défini sur un espace probabilisé que l'on veut partitionner en un nombre fixé k de classes. Après avoir défini un représentant d'une classe, appelé noyau, et un critère de classification, on définit un algorithme stochastique moyennisé des k-noyaux, dont on établit la convergence.

1 Introduction

Actuellement, les dimensions des ensembles de données augmentent plus vite que la vitesse de calcul des ordinateurs et il est important de disposer d'algorithmes d'exécution rapide (Bottou, 2010), prenant en compte dans le même temps plus d'observations que d'autres et donnant alors de meilleures estimations même s'ils ne sont pas les plus efficaces.

La méthode de classification en un nombre fixé k de classes peut-être la plus utilisée est celle des k-means (Forgy, 1965). Lorsque l'on dispose de beaucoup de données, on peut utiliser la méthode des k-means séquentielle de MacQueen (1967) : une observation est introduite à chaque itération et affectée à une classe du barycentre de laquelle elle est la plus proche et ce barycentre est alors actualisé. Cette méthode est basée sur un modèle probabiliste de population : on suppose que l'ensemble des données est un échantillon i.i.d. d'un vecteur aléatoire défini sur un espace probabilisé. L'algorithme de MacQueen donne une solution locale au problème de la recherche d'une partition de cet espace d'inertie intra-classes minimale. On peut le présenter comme un algorithme de gradient stochastique.

Nous étudions dans cet article la convergence d'un algorithme des k-noyaux séquentiel, basé sur le modèle probabiliste de population, mais pour lequel le barycentre est remplacé de façon plus générale par un représentant d'une classe appelé noyau et le critère L_2 par un critère plus général ; nous utilisons un algorithme de gradient stochastique pour déterminer une solution locale. Nous en établissons la convergence en utilisant un théorème de convergence presque sûre d'un processus de gradient stochastique à pas matriciels aléatoires établi dans (Monnez, 2006). Pour en accélérer la convergence, nous proposons d'utiliser une moyennisation classique en approximation stochastique (Polyak et Juditsky, 1992), comme dans (Cardot, Cénac, Monnez, 2011).

2 Critère de classification

Soit une suite d'observations (z_1 , ...,z_n , ...) qui constituent un échantillon i.i.d. d'un vecteur aléatoire Z dans R^d , que l'on suppose absolument continu (hypothèse H1a), défini sur un espace probabilisé (Ω, A, P) que l'on veut partitionner en k classes à partir de ces observations. En suivant Diday (1979), on définit :

1) *Un représentant θ^r d'une classe Ω_r*, appelé noyau. Ce peut être par exemple : un point de R^d ; un sous-espace affine de dimension fixée de R^d caractérisé par ses paramètres ; un paramètre de la loi de probabilité conditionnelle de Z dans Ω_r. On suppose de façon générale que θ^r est un paramètre réel de dimension q.

2) *Une mesure de dissimilarité entre un point z de R^d et un représentant θ^r, $D_r(z;\theta^r)$*. Par exemple, pour θ^r point de R^d : $D_r(z;\theta^r) = \| z-\theta^r \|^2$ *ou* $\| z-\theta^r \|$.

3) *Un critère de classification* : déterminer une partition $(\Omega_1, ..., \Omega_k)$ et un k-uplet de représentants $(\theta^1, ..., \theta^k)$ qui rendent minimale la fonction

$$\phi(A_1,...,A_k;x^1,...,x^k) = \sum_{r=1}^{k} E\left[1_{A_r} D_r(Z;x^r)\right] \geq E\left[\min_r D_r(Z;x^r)\right].$$

En supposant les x^r tous distincts, l'égalité est réalisée pour

$$A_r = \left\{ \omega : D_r(Z(\omega);x^r) = \min_j D_j(Z(\omega);x^j) \right\},$$

à condition que l'hypothèse H1b soit vérifiée :

(H1b) *Pour* $x^r \neq x^{r'}$, $P(D_r(Z;x^r) = D_{r'}(Z;x^{r'})) = 0$.

Le problème revient alors à déterminer $(\theta^1, ..., \theta^k)$ qui rendent minimale la fonction

$$g(x^1,...,x^k) = E\left[\min_r D_r(Z;x^r)\right] = E\left[\sum_{r=1}^{k} I_r(Z;x)) D_r(Z;x^r)\right],$$

avec $I_r(Z;x) = 1_{\{D_r(Z;x^r) = \min_j D_j(Z;x^j)\}}$.

3 Définition d'un processus de gradient stochastique

En supposant vérifiée l'hypothèse

(H1c) $\nabla_{x^r} g(x^1,...,x^k) = E\left[I_r(Z;x)\nabla_{x^r} D_r(Z;x^r)\right], r = 1,...,k,$

θ^1, ..., θ^k forment une solution du système d'équations

$$E\left[I_r(Z;x)\nabla_{x^r} D_r(Z;x^r)\right] = 0, r = 1,...,k .$$

Pour le résoudre, on définit le processus de gradient stochastique $X_n = (X_n^1,...,X_n^k)$:

$$X_{n+1}^r = X_n^r - a_n^r I_r(Z_n;X_n)\nabla_{x^r} D_r(Z_n;X_n^r) = X_n^r - a_n^r \nabla_{x^r} g(X_n) - a_n^r V_n^r,$$

$$V_n^r = a_n^r I_r(Z_n;X_n)\nabla_{x^r} D_r(Z_n;X_n^r) - \nabla_{x^r} g(X_n), r = 1,...,k;$$

a_n^r est une variable aléatoire positive mesurable par rapport à la tribu du passé T_n.

Soit ∇g le gradient de g et $V_n = (V_n^1,...,V_n^k)'$. Le processus récursif (X_n) s'écrit

$$X_{n+1} = X_n - A_n \nabla g(X_n) - A_n V_n ,$$

la matrice A_n étant diagonale de termes diagonaux $a_n^1,...,a_n^1,...,a_n^k,...,a_n^k$, chaque a_n^r étant répété d fois.

On applique le théorème 1 de convergence presque sûre d'un processus de gradient stochastique à pas matriciels aléatoires de Monnez (2006). On suppose :

(H2) a) *Il existe L >0 tel que, pour tout n ≥1,*

$$g(X_{n+1}) - g(X_n) \le \langle X_{n+1} - X_n, \nabla g(X_n) \rangle + L \|X_{n+1} - X_n\|^2 \text{ p.s.}$$

b) *La suite (X_n) est p.s. bornée et ∇g est continue presque partout dans le compact la contenant.*

(H 3) *Il existe deux suites de variables aléatoires positives (B_n) et (C_n), adaptées à la suite de sous-tribus (T_n) et telles que, presque sûrement,*

$$E\left[\|A_n V_n\|^2 | T_n\right] \le B_n g(X_n) + C_n, \sum_1^\infty (B_n + C_n) < \infty.$$

(H4) a) $\forall n \ge 1, \min_r a_n^r > 0 ; \max_r \sup_n a_n^r < \min\left(\frac{1}{2}, \frac{1}{4L}\right)$ p.s.

b) $\sum_1^\infty \max_r a_n^r = \infty$ p.s.

c) $\sup_n \frac{\max_r a_n^r}{\min_r a_n^r} < \infty$ p.s.

Théorème. *Sous les hypothèses H1a, b, c, H2a, H3, H4a,*

$$g(X_n) \text{ et } \sum_{r=1}^k \sum_{n=1}^\infty a_n^r \|\nabla_{x^r} g(X_n)\|^2$$

convergent p.s. Si les hypothèses H2b et H4 b, c sont aussi vérifiées, alors $\nabla g(X_n)$ et la distance entre X_n et l'ensemble des points stationnaires de g convergent p.s vers 0.

Soit $m_n^r = 1 + \sum_{l=1}^{n-1} I_l(Z_l ; X_l)$. Soit $\frac{1}{2} < \alpha \le 1$, $c_\alpha > 0$, $c_\gamma > 0$. On peut définir, comme dans (Cardot, Cénac, Monnez, 2011),

$$a_n^r = \begin{cases} a_{n-1}^r & \text{si } I_r(Z_n ; X_n) = 0 \\ \dfrac{c_\gamma}{(1 + c_\alpha m_n^r)^\alpha} & \text{si } I_r(Z_n ; X_n) = 1. \end{cases}$$

Les hypothèses H4, sauf H4c, sont alors vérifiées. Un choix usuel est $\alpha = \frac{3}{4}, c_\alpha = 1$.

On définit pour *r = 1,...,k,* le processus stochastique moyennisé $(\overline{X}_n^r)$ tel que

$$\overline{X}_1^r = X_1^r ; \overline{X}_{n+1}^r = \begin{cases} \overline{X}_n^r & \text{si } I_r(Z_n ; X_n) = 0 \\ \dfrac{m_n^r \overline{X}_n^r + X_{n+1}^r}{m_n^r + 1} & \text{si } I_r(Z_n ; X_n) = 1. \end{cases}$$

Si l'ensemble des points stationnaires de *g* est fini, alors, sous les hypothèses du théorème, la suite (X_n) converge presque sûrement vers un de ces points stationnaires, car $X_{n+1} - X_n$ converge p.s. vers 0 ; on en déduit que la suite $\left(\overline{X}_n\right)$ converge p.s. vers le même point.

4 Conclusion

Cet algorithme des k-noyaux et le théorème de convergence admettent plusieurs cas particuliers dont celui de MacQueen (1967), ainsi que l'algorithme CCM des k-médianes séquentiel (Cardot, Cénac, Monnez, 2011), dont on montre sur des simulations et un exemple la rapidité d'exécution par rapport à des algorithmes classiques. Une extension possible est la définition d'un algorithme des k-noyaux en présence de co-variables.

Références

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. *Compstat 2010, Lechevallier,Y., Saporta, G. (eds)*: 177-186. Physica Verlag, Springer.

Cardot, H., Cénac, P., Monnez, J.M. (2011). Fast clustering of large datasets with sequential k-medians : a stochastic gradient approach. Soumis à *CSDA*. hal-00558145

Diday, E., et collaborateurs (1979). *Optimisation en classification automatique.* INRIA, Le Chesnay.

Forgy, E. (1965). Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. *Biometrics*, 21: 768-769.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability*, Vol. I: Statistics: 281-297. Univ. California Press, Berkeley, Calif.

Monnez, J.M. (2006). Almost sure convergence of stochastic gradient processes with matrix step sizes. *Statistics & Probability Letters*, 76 (5): 531-536.

Polyak, B., Juditsky, A. (1992). Acceleration of stochastic approximation. *SIAM J. Control and Optimization*, 30: 838-855.

Summary

Consider the problem of partitioning in a fixed number k of clusters with a fast algorithm a large set of high dimensional data, which can be taken sequentially. Suppose that vectors of data are i.i.d. observations of a random vector. After defining a representative (kernel) of each cluster, a dissimilarity measure and a classification criterion, we define a sequential k-kernel algorithm as a stochastic gradient algorithm and give a theorem of almost sure convergence. Particular cases are interalia the sequential k-means algorithm of MacQueen and the sequential k-medians algorithm of Cardot, Cénac, Monnez. We use a classical method of averaging to accelerate its convergence.

Rotation orthogonale dans PCAMIX

Marie Chavent [*,**], Vanessa Kuentz[***]
Jérôme Saracco[*,**]

[*]Université de Bordeaux, IMB, CNRS, UMR 5251, France
{marie.chavent,jerome.saracco}@math.u-bordeaux1.fr
[**]INRIA Bordeaux Sud-Ouest, CQFD team, France
[***]Cemagref, UR ADBX, F-33612 Cestas Cedex, France
vanessa.kuentz@cemagref.fr

Résumé. La rotation orthogonale dans PCAMIX a été initialement introduite par Kiers (1991). PCAMIX est une méthode d'analyse en composantes principales pour un mélange de variables quantitatives et qualitatives qui inclut comme cas particuliers l'Analyse en Composantes Principales (ACP) et l'Analyse des Correspondances Multiples (ACM). Dans ce papier, nous donnons une nouvelle présentation de PCAMIX où les composantes principales et les loadings sont obtenues à l'aide d'une Décomposition en Valeurs Singulières. Dans ce contexte, nous proposons une nouvelle expression analytique directe pour l'angle varimax de rotation dans PCAMIX. L'algorithme de rotation qui en résulte est simple et relativement peu coûteux en temps de calculs. Une application sur un jeu de données réel illustre l'intérêt pratique de la rotation. L'ensemble des codes sera prochainement disponible dans un package R nommé *PCAmixdata*.

Mots-clés: mélange de données quantitatives et qualitatives, analyse en composantes principales, analyse des correspondances multiples, rotation.

1 Introduction

Différents critères ont été proposés pour la rotation en ACP. Le plus célèbre est varimax, introduit par Kaiser en 1958. L'idée est de maximiser la variance des colonnes de la matrice des loadings qui contient les corrélations au carré des variables aux composantes principales. Des groupes de variables se forment, facilitant ainsi l'interprétation. En Analyse des Correspondances, différents travaux ont été récemment réalisés (voir par exemple van de Velden and Kiers, 2005). D'autre part, Kiers (1991) a considéré la rotation orthogonale dans la méthode PCAMIX. Cette méthode d'analyse factorielle pour un ensemble de données qualitatives et quantitatives inclut comme cas particuliers l'ACP et l'ACM. Dans cet article, nous proposons une écriture de PCAMIX sous forme de Décomposition en Valeurs Singulières qui facilite l'utilisation de la rotation.

L'idée de la rotation orthogonale dans PCAMIX utilise la définition des loadings au carré (voir Chavent et al., 2011 pour plus de détails). Pour une variable quantitative, il s'agit du carré de sa corrélation avec la composante principale. Pour une variable qualitative, c'est le rapport de corrélation qui est utilisé. La fonction varimax est alors appliquée à la matrice des

loadings au carré, conduisant à un nouveau problème d'optimisation. En deux dimensions, la définition de la matrice de rotation orthogonale optimale s'obtient en résolvant un problème non contraint. Nous utilisons différentes astuces et formules trigonométriques pour annuler la dérivée et obtenir une nouvelle écriture explicite de l'angle optimal de rotation (planaire). Cette écriture est plus simple que celle proposée par Kiers et moins coûteuse en temps de calcul. Dans le cas de plus de deux dimensions, nous utilisons l'algorithme proposé par Kaiser (1958) qui consiste à réaliser des rotations successives de paires de facteurs jusqu'à convergence.

2 Un exemple illustratif

Nous appliquons l'approche de rotation sur le jeu de données "Prostate" utilisé entre autres par Hunt et al. (2003). Il concerne 506 patients atteints du cancer de la prostate ayant suivi un essai clinique aléatoire visant à comparer quatre traitements. Ces données sont disponibles dans le package R "Hmisc" développé par Harrell (2010). Les données sont mixtes : 8 variables sont quantitatives ("age", "poids", "pression sanguine systolique", "pression sanguine diastolique", "sérum hémoglobine", "taille de la tumeur", "indice de niveau de la tumeur", "sérum prostatique") et 4 variables sont qualitatites ("niveau de performance", "historique cardiovasculaire", "électrocardiogramme", "métastases"). Voici un exemple de code R suivi de quelques sorties graphiques et numériques obtenues avec le package "PCAmixdata" (prochainement disponible).

Les sorties numériques fournies par le package comprennent les scores des composantes principales avant et après rotation (standardisées dans ce cas), les coordonnées des modalités des variables qualitatives avant et après rotation, le cercle des corrélations pour les variables quantitatives avant et après rotation ainsi que les loadings au carré de chaque variable (carré de sa corrélation linéaire si elle est quantitative, rapport de corrélation si elle est qualitative) avec la composante principale.
Les lignes de code,

```
require(Hmisc)
getHdata(prostate)
require(PCAmixdata)
res<-PCAmix(X.quanti=prostate.quanti,X.quali=prostate.quali,ndim=4)
```

permettent de lancer la méthode PCAMIX et de conserver 4 composantes principales. Les graphiques obtenus sont présentés dans la Figure 1. Pour ne pas surcharger le résumé, nous nous sommes limités aux deux premières composantes principales.

La ligne suivante,

```
rot<-PCArot(res,dim=4)
```

permet d'effectuer une rotation sur les 4 composantes principales obtenues avec PCAMIX (voir Figure 1).

La rotation des composantes principales conduit à une meilleure association entre les variables (voir Tableau 2 et Figure 1) : le poids est associé avec les pressions systolique et diastolique, le sérum hémoglobine avec la taille et le niveau de la tumeur, l'âge ne s'associe pas avec les autres variables et le sérum prostatique se distingue également.

	Before rotation				*After rotation*			
	1	2	3	4	1	2	3	4
age	-0.06	0.10	-0.58	0.15	-0.03	-0.06	**-0.61**	-0.09
wt	-0.44	0.20	0.29	0.01	-0.26	**0.46**	0.18	-0.08
sbp	-0.36	0.77	0.12	0.06	0.05	**0.84**	-0.15	-0.01
dbp	-0.43	0.65	0.30	0.00	-0.05	**0.83**	0.06	-0.04
hg	-0.51	-0.09	0.32	0.02	**-0.46**	0.25	0.28	-0.13
sz	0.42	0.32	0.00	-0.40	**0.65**	0.06	0.06	-0.08
sg	0.57	0.27	0.15	-0.38	**0.72**	0.00	0.21	0.04
ap	0.53	0.14	0.28	0.56	0.18	-0.02	0.03	**0.82**
pf	0.23	0.16	0.22	0.51	0.18	0.02	0.17	**0.76**
hx	0.06	0.05	0.26	0.09	0.04	0.02	**0.40**	0.00
ekg	0.04	0.20	0.31	0.13	0.09	0.15	**0.41**	0.04
bm	0.48	0.09	0.00	0.00	**0.46**	0.00	0.00	0.11

TAB. 1 – *Loadings au carré avec les 4 premières composantes avant et après rotation*

References

Chavent, M., Kuentz, V., Saracco, J., (2011), Orthogonal rotation in PCAMIX, *Under review*.

Harrell F. E., (2010), The Hmisc package, CRAN R Project.

Hunt, L.A. and Jorgensen, M.A., (2003), Mixture model clustering for mixed data with missing information, *Computational Statistics and Data Analysis*, **41**, 429-440.

Kaiser, H.F., (1958), The varimax criterion for analytic rotation in factor analysis, *Psychometrika*, **23**(3), 187-200.

Kiers, H.A.L., (1991), Simple structure in Component Analysis Techniques for mixtures of qualitative and quantitative variables, *Psychometrika*, **56**, 197-212.

van de Velden, M., and Kiers, H. A. L., (2005), Rotation in correspondence analysis, *Journal of Classification*, **22**, 251-271.

Summary

Orthogonal rotation in PCAMIX has been initially introduced by Kiers (1991). PCAMIX is a factorial method for a mixture of quantitative and qualitative variables. It includes as ordinary Principal Component Analysis (PCA) and Multiple Correspondence Analysis (MCA) as special cases. In this paper, we give a new presentation of PCAMIX where the principal components and the squared loadings are obtained from a Singular Value Decomposition. In this context we give a new analytic expression of the varimax angle for rotation in PCAMIX. The resulting rotation algorithm is simple and computationnaly efficient. An application on real data shows the benefits of using rotation. All source codes will be soon available in the R package *PCAmixdata*.
Mots-clés: mixture of qualitative and quantitative data, principal component analysis, multiple correspondance analysis, rotation.

Rotation orthogonale dans PCAMIX

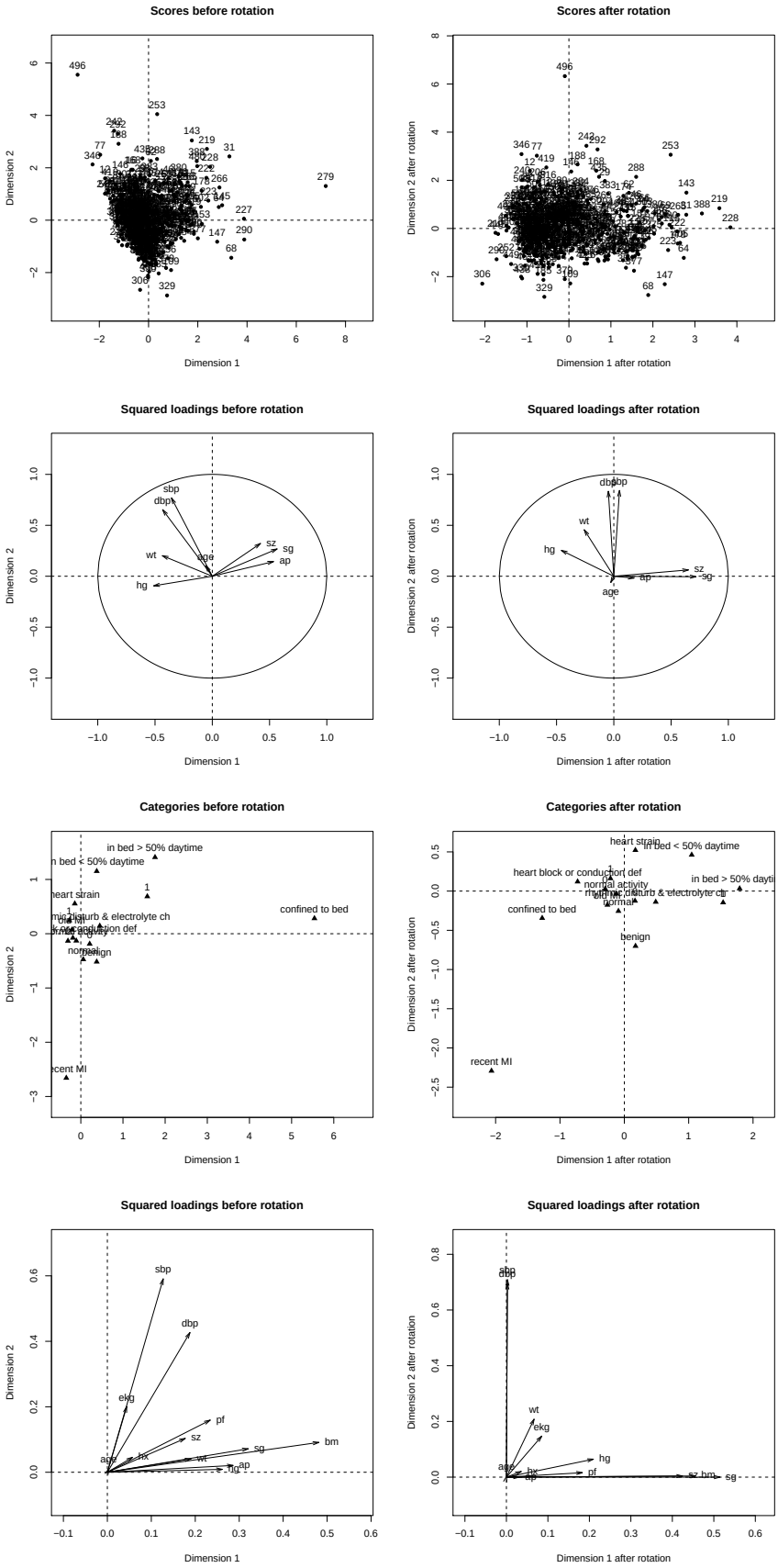


FIG. 1 – *Scores, cercles de corrélation, modalités et loadings au carré avant et après rotation.*

Intégration de contraintes *must-link* et *cannot-link* pour la classification : une approche indépendante de l'algorithme

Jacques-Henri Sublemontier*, Lionel Martin*, Guillaume Cleuziou* Matthieu Exbrayat*

*LIFO – Université d'Orléans – Orléans, FRANCE
nom.prenom@univ-orleans.fr

Résumé. Nous proposons une nouvelle formalisation de la problématique de classification semi-supervisée, en présence de contraintes *must-link* (ML) et *cannot-link* (CL). Il s'agit d'utiliser les contraintes pour apprendre de manière itérative un espace de représentation des données afin de favoriser leur satisfaction par l'algorithme de classification quel qu'il soit.

1 Introduction

La classification (ou clustering) semi-supervisée est l'objet d'intenses recherches ces dernières années, motivées par la possibilité qu'elle offre d'intégrer au processus - a priori non-supervisé - des connaissances en faible quantité, faciles à obtenir et susceptibles d'améliorer significativement la qualité des clusters obtenus. Les premières propositions visaient à interagir sur le processus même de clustering en modifiant quelques algorithmes bien connus (k-moyennes, COBWEB, DBSCAN) (Wagstaff et al., 2001; Ruiz et al., 2010) de manière à empêcher toute contradiction avec des connaissances extérieures exprimées sous-formes de contraintes sur des paires d'objets. Deux types de contraintes sont usuellement considérées : *must-link* (ML) (resp. *cannot-link* (CL)) indiquant que deux objets doivent (resp. ne doivent pas) figurer dans le même cluster in fine. Assez rapidement, des formalisations plus souples du problème ont été suggérées en introduisant une pénalité (e.g. au critère objectif utilisé) liée à la violation des contraintes (Dhillon et al., 2004). Une autre manière de procéder consiste à utiliser les contraintes fournies pour adapter/corriger la métrique ou plus généralement l'espace de représentation des données (Bilenko et al., 2004) ; cette approche est indépendante de la méthode de clustering envisagée ce qui constitue à la fois un avantage : la généricité et un inconvénient : l'absence de prise en compte de l'algorithme dans l'adaptation de la métrique [1].

L'orientation que nous choisissons vise à apprendre un espace de représentation des données de manière itérative en utilisant à chaque itération les résultats de l'algorithme de classification (vu comme une boîte noire) pour améliorer les chances pour l'algorithme de satisfaire les contraintes. La formalisation que nous proposons est une alternative à l'approche de type boosting (BOOSTCLUSTER) présentée par Liu et al. (2007) à laquelle nous nous comparons expérimentalement.

1. Les contraintes ont généralement toutes le même poids.

2 Présentation du modèle

Dans la suite, nous noterons : n le nombre d'individus et p la dimensionalité des individus. X désigne la matrice des données $X \in \mathcal{M}_{n\times p}$. u est un vecteur de projection. $d_u(x_i, x_j)$ (resp. $d(x_i, x_j)$) désigne la distance euclidienne entre x_i et x_j dans u (resp. dans l'espace d'origine).

L'approche que nous proposons doit conduire l'algorithme de classification, appelé A par la suite, à satisfaire les contraintes ML et CL. Nous proposons d'apprendre un nouvel espace de représentation satisfaisant ces contraintes et préservant au mieux la représentation initiale des données. Nous utilisons pour ce faire la technique bien connue de l'analyse en composantes principales. L'idée est de trouver un sous-espace de l'espace d'origine telle que la variance du nuage des individus soit maximale *i.e* le maximum d'information soit conservé. Cet aspect n'a pas du tout été pris en compte dans BOOSTCLUSTER. Partant de l'optimisation du critère de l'ACP, le problème qui nous motive consiste à s'assurer que les individus impliqués dans une contrainte ML (resp. CL) soient proches (resp. éloignés) dans le sous-espace de représentation. Nous proposons de modéliser cette volonté par les conditions suivantes qui devront être respectées :

- si $(x_i, x_j) \in ML$, il existe une constante $\xi_{i,j}$ telle que si $d_u^2(x_i, x_j) \leq \xi_{i,j}$ alors l'algorithme A groupera x_i et x_j.
- si $(x_i, x_j) \in CL$, il existe une constante $\xi_{i,j}$ telle que si $d_u^2(x_i, x_j) \geq \xi_{i,j}$ alors l'algorithme A séparera x_i et x_j.

$(x_i, x_j) \in ML \Longrightarrow x_i$ et x_j doivent être dans un même groupe
$(x_i, x_j) \in CL \Longrightarrow x_i$ et x_j doivent être dans des groupes différents

Nous définissons alors le problème $\mathcal{P}$ d'optimisation permettant d'obtenir le meilleur vecteur u^* de projection :

$$\mathcal{P} \left| \begin{array}{lrcll} \max\limits_{u} & u^T X^T X u & & & \\ s.c. & u^T u & = & 1 & \\ & d_u^2(x_i, x_j) & \leq & \xi_{i,j} & \forall (x_i, x_j) \in ML \quad (cs1) \\ & d_u^2(x_i, x_j) & \geq & \xi_{i,j} & \forall (x_i, x_j) \in CL \quad (cs2) \end{array} \right.$$

Pour résoudre ce problème d'optimisation, nous considérons le Lagrangien associé $\mathcal{L}(u, \lambda, \mu)$:

$$\mathcal{L}(u, \lambda, \mu) = u^T X^T X u - \sum_{(x_i,x_j)\in ML} \big(\lambda_{i,j}(d_u^2(x_i, x_j) - \xi_{i,j})\big) + \sum_{(x_i,x_j)\in CL} \big(\lambda_{i,j}(d_u^2(x_i, x_j) - \xi_{i,j})\big) - \mu(u^T u - 1)$$

que l'on peut réécrire :

$$\mathcal{L}(u, \lambda, \mu) = u^T \tilde{X} u + \sum_{(x_i,x_j)\in ML} \lambda_{i,j}\xi_{i,j} - \sum_{(x_i,x_j)\in CL} \lambda_{i,j}\xi_{i,j} - \mu(u^T u - 1)$$

où $\tilde{X}$ dépend des $\lambda_{i,j}$ et de μ. Le vecteur u^* optimal correspond exactement au vecteur propre associé à la plus grande valeur propre de la matrice $\tilde{X}$ définie positive. Nous obtenons le sous espace de projection U^* en sélectionnant les prochains vecteurs propres associés aux plus grandes valeurs propres restantes. Le problème est que la diagonalisation de $\tilde{X}$ nécessite

Algorithm 1 Uzawa BoC

Input : Données X, Nombre de groupes K, Algorithme de clustering A, $\{(x_i, x_j) \in ML\}$, $\{(x_i, x_j) \in CL\}$
1 Initialisation des $\lambda_{i,j}$ et des $\xi_{i,j}$
repeat
  2 Calcul du sous-espace U *via* $\mathcal{P}$
  3 Calcul de $X' = XU$ et application de A sur X'
  4 Mise à jour des $\xi_{i,j}$ par (2)
  5 Mise à jour des $\lambda_{i,j}$ par (1)
until convergence vers un U^* optimal

de connaître la valeur des multiplicateurs de Lagrange $\lambda_{i,j}$. Nous allons approcher itérativement la valeur de ces multiplicateurs pour résoudre notre problème d'optimisation en adaptant l'algorithme Uzawa (**Algorithm 1**) *i.e* partant d'une initialisation des $\lambda_{i,j} = 0$ pour tous les couples (x_i, x_j) impliqués dans une contrainte ML ou CL, les nouveaux $\lambda_{i,j}$ s'obtiennent de la manière suivante :

$$\lambda_{i,j}^{t+1} = max(0, \lambda_{i,j}^{t} + \rho \times (d_u^2(x_i, x_j) - \xi_{i,j})) \tag{1}$$

où ρ est un pas que l'on peut définir de manière constante. Notons que l'obtention de la solution optimale u^* du Lagrangien est indépendante des valeurs de $\xi_{i,j}$, nous déterminons ces derniers de manière purement algorithmique. La résolution de $\mathcal{P}$ permet *in fine* de trouver le sous espace U^* optimal favorisant l'algorithme A à respecter les contraintes de semi-supervision. L'algorithme A intervient de manière implicite dans le programme d'optimisation, et il guide l'apprentissage du sous espace en influençant la mise à jour des paramètres du Lagrangien $\mathcal{L}(u, \lambda)$ et des valeurs de $\xi_{i,j}$.

Parmi les données du problème $\mathcal{P}$, nous avons introduit les constantes $\xi_{i,j}$ servant à garantir que les individus $(x_i, x_j) \in ML$ (resp. CL) soient proches (resp. éloignés) dans le sous-espace de projection. Initialement nous fixons $\xi_{i,j} = d^2(x_i, x_j)$ si $(x_i, x_j) \in ML$ et $\xi_{i,j} = 0$ si $(x_i, x_j) \in CL$. Nous mettons ensuite à jour de manière heuristique les $\xi_{i,j}$ si les contraintes correspondantes $(cs1)$ ou $(cs2)$ sont satisfaites, mais l'algorithme A ne parvient pas à respecter les contraintes ML et CL. Nous décidons alors dans ce cas de "durcir" les contraintes :

$$\xi_{i,j}^{t+1} = \begin{array}{c} \frac{d_u^2(x_i,x_j)+d^2(x_i,x_j)}{2} \; si \; (x_i, x_j) \in CL \; et \; d_u^2(x_i, x_j) < \xi_{i,j} \\ \frac{d_u^2(x_i,x_j)}{2} \; si \; (x_i, x_j) \in ML \; et \; d_u^2(x_i, x_j) > \xi_{i,j} \end{array} \tag{2}$$

ce qui correspond à diminuer (resp. augmenter) la valeur des $\xi_{i,j}$ pour les $(x_i, x_j) \in ML$ (resp. CL) non respectées par A. C'est de cette façon que l'algorithme A intervient dans la mise à jour des $\lambda_{i,j}$ (*via* $\xi_{i,j}$) et influence le nouvel espace de représentation optimal de $\mathcal{P}$.

3 Résultats expérimentaux

Nous avons réalisé une étude expérimentale comparative entre notre approche et BOOST-CLUSTER. Nous validons notre approche sur différents jeux de données artificiels (*2D2K*[2])

2. Le jeu de données *2D2K* contient $1,000$ individus générés par un mélange de deux gaussiennes bi-dimensionelle avec des matrices de variances diagonales.

et réels (Iris, WDBC, Vowel). Nous observons une amélioration de la qualité du clustering obtenu en prenant comme algorithme de clustering K-moyennes avec notre algorithme (Uz) comparé à BOOSTCLUSTER (BC). Les résultats de la figure (1) correspondent à une moyenne de 20 runs sur les différents jeux de données, et nous faisons varier le nombre de contraintes de 0 à 1600 en imposant un équilibre entre les contraintes ML et CL. L'évaluation que nous avons choisie est le F-score et enfin, le nombre de dimensions du sous espace de représentation calculé à chaque étape correspond au nombre de valeurs propres positives.

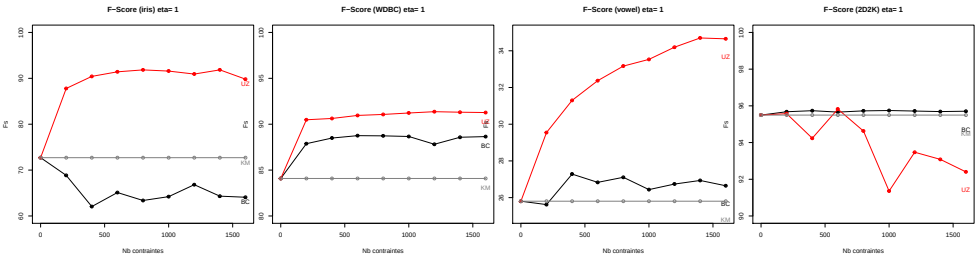


FIG. 1 – *comparatif entre l'approche Uzawa BoC (Uz) et* BOOSTCLUSTER *(BC) par rapport à K-moyennes de référence (KM)*

Références

Bilenko, M., S. Basu, et R. J. Mooney (2004). Integrating constraints and metric learning in semi-supervised clustering. In C. E. Brodley (Ed.), *ICML*, Volume 69 of *ACM International Conference Proceeding Series*. ACM.

Dhillon, I., Y. Guan, et B. Kulis (2004). Kernel k-means : spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 556. ACM.

Liu, Y., R. Jin, et A. K. Jain (2007). Boostcluster : boosting clustering by pairwise constraints. In P. Berkhin, R. Caruana, et X. Wu (Eds.), *KDD*, pp. 450–459. ACM.

Ruiz, C., M. Spiliopoulou, et E. M. Ruiz (2010). Density-based semi-supervised clustering. *Data Mining and Knowledge Discovery 21*, 345–370.

Wagstaff, K., C. Cardie, S. Rogers, et S. Schrödl (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, San Francisco, CA, USA, pp. 577–584. Morgan Kaufmann Publishers Inc.

Summary

We present a new formalisation for semi-supervised clustering with pairwise contraints. The aim of the approach is to use the constraints to learn iteratively a projection space for the data, such as the clustering algorithm satisfies the constraints at the best.

Clustering collaboratif : le challenge de regrouper conjointement

Germain Forestier

INRIA / INSERM / CNRS / Univ. Rennes 1, VISAGES U746, Rennes, France
germain.forestier@inria.fr

Résumé. Le clustering collaboratif consiste à faire collaborer conjointement plusieurs méthodes de clustering afin de parvenir à un résultat consensuel. Les différentes méthodes impliquées dans la collaboration vont partager des informations et vont remettre en cause leurs décisions en fonction des solutions proposées par les autres méthodes. Un enjeu important consiste à faire collaborer des méthodes différentes tout en assurant une prise en compte des décisions de chacune d'entre elles. Cet article présente les principales approches en clustering collaboratif et contextualise nos travaux dans ce domaine. Enfin, quelques enjeux émergents sont également exposés.

1 Introduction

L'idée de combiner les décisions de plusieurs méthodes de classification non supervisée a émergé du travail important mené dans le domaine de la combinaison de méthodes supervisées (Kuncheva, 2008). Dans le cas supervisé, le travail est simplifié par l'existence d'une référence commune (*i.e.* les classes) qui peut servir à faciliter la combinaison des décisions.

A contrario, dans le cadre de la combinaison de méthodes non supervisées, il n'existe pas de liens évident entre les clusters des différents résultats et les ceux-ci n'ont pas forcément le même nombre de clusters. D'autres approches ont donc été envisagées pour permettre la combinaison de ces résultats hétérogènes. De plus, d'autres contraintes comme la distribution des données sur plusieurs sites, ou le respect de la confidentialité des données, rendent parfois impossible un traitement centralisé et constituent une autre motivation de ces approches.

Afin d'adresser ce problème, nous nous sommes intéressé au clustering collaboratif qui consiste à faire collaborer plusieurs méthodes de clustering conjointement. Ces différentes méthodes vont partager des informations et vont remettre en cause leurs décisions en fonction des décisions proposées par les autres méthodes. Ainsi, une discussion est entreprise entre les méthodes afin de faire converger collectivement les différents résultats.

Dans cet article, nous présentons les principales approches en clustering collaboratif afin de contextualiser nos travaux et d'illustrer les résultats obtenus. Enfin, les enjeux émergents de ce domaine sont abordés.

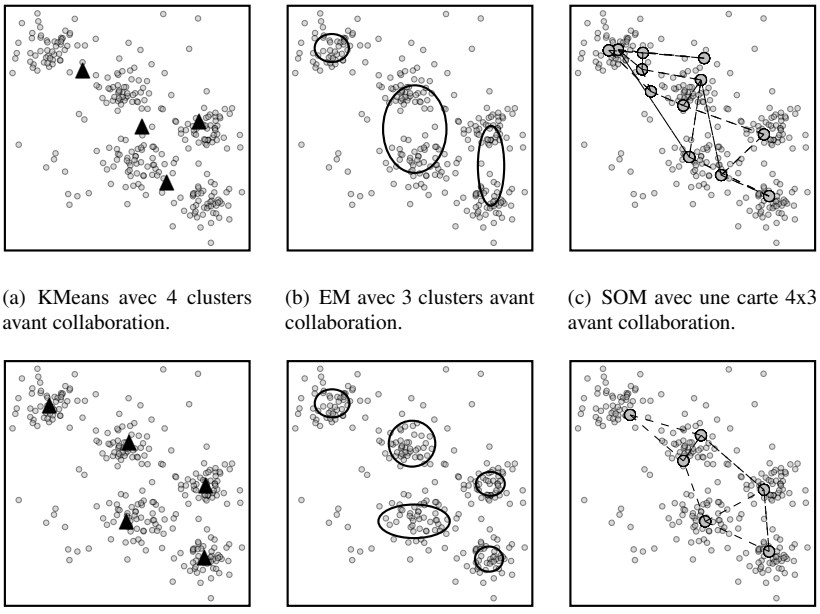
2 Le clustering collaboratif

Plusieurs approches existent en clustering collaboratif mettant en œvre la collaboration à différents niveaux. Les méthodes dîtes de clustering par ensemble (Strehl et Ghosh, 2002), s'intéressent à la combinaison de résultats de clustering en étudiant uniquement l'affectation des objets aux clusters des résultats. De ce fait, les données utilisées pour générer les résultats ne sont plus utilisées lors du processus collaboratif, qui consiste alors à trouver une partition consensuelle résumant l'ensemble des partitions de départ. Ces approches font ainsi l'hypothèse que les partitions générées initialement sont de qualité suffisante pour permettre au processus collaboratif de trouver un résultat pertinent. Les approches de clustering multi-objectifs (Handl et Knowles, 2007) tentent de réduire cette hypothèse, en créant de nouvelles partitions à partir des partitions initiales à l'aide d'opérateurs de croisement et de mutation. Cependant, ces opérateurs ne mettent pas en œuvre les méthodes de clustering utilisées pour produire les partitions initiales, introduisant de ce fait un nouveau biais.

Une autre approche en clustering collaboratif, consiste à se concentrer sur une méthode de clustering, en étudiant la manière dont peuvent collaborer plusieurs modèles construits par cette méthode. Dans ce cadre, les travaux de Pedrycz (2007) se sont intéressés à l'algorithme Fuzzy-c-means et proposent une méthode consistant à comparer les degrés d'appartenance des objets aux centroides des différents résultats. De manière similaire, Cleuziou et al. (2009) se sont également intéressés à l'algorithme Fuzzy-c-means et ont proposé une méthode de clustering collaboratif en introduisant un terme de pénalité permettant d'évaluer et de réduire itérativement les désaccords entre les résultats. Une autre approche, proposée par Grozavu et Bennani (2010), s'est intéressée à l'algorithme des cartes de Kohonen (SOM) et à la combinaison de plusieurs cartes.

Ces différentes approches permettent d'optimiser la collaboration pour une méthode donnée et ont obtenu de très bons résultats pour le clustering collaboratif. Cependant, elles restent spécifiques à l'algorithme de clustering utilisé et sont difficilement généralisables. Dans la quête d'une méthode plus générique, permettant l'utilisation conjointe de différents algorithmes de clustering, nous nous sommes intéressés dans nos travaux à la réduction des désaccords entre résultats de clustering provenant de méthodes de clustering différentes. Ce choix pose la question importante de la possibilité de partager des informations entre des résultats dont la structure sous-jacente des modèles n'est pas similaire. De ce fait, il est nécessaire de proposer un mécanisme générique, permettant d'une part d'identifier les désaccords, et d'autre part d'effectuer des actions permettant leur résolution.

Dans le cadre de nos travaux (Forestier, 2010), nous avons proposé une approche de clustering collaboratif permettant la collaboration de résultats produits par des méthodes différentes. Les désaccords entre les résultats proposés par les différentes méthodes sont appelés des *conflits*. Ces conflits sont identifiés en observant la répartition des objets dans les clusters des différents résultats. Cette étape d'identification est donc indépendante des méthodes de clustering utilisées. A l'issue de cette étape d'identification, une tentative de résolutions des conflits est engagée. Cette étape de résolution consiste à appliquer des opérateurs (*e.g.* fusion de clusters, scission de clusters) aux résultats impliqués dans le conflit. L'application de ces opérateurs est dépendante de la méthode de clustering utilisée, permettant ainsi de mener des modifications locales à partir d'une information obtenue de manière globale. La seule contrainte concernant la participation d'une méthode de clustering à la collaboration, est sa capacité à implémenter ces opérateurs. La résolution des conflits peut être itérative, ou faire intervenir des métaheuris-



(a) KMeans avec 4 clusters avant collaboration.

(b) EM avec 3 clusters avant collaboration.

(c) SOM avec une carte 4x3 avant collaboration.

(d) KMeans avec 5 clusters après collaboration.

(e) EM avec 5 clusters après collaboration.

(f) SOM avec 6 neurones après collaboration.

FIG. 1 – *Résultats de clustering avant (a,b,c) et après (d,e,f) collaboration pour trois résultats produits avec trois méthodes différentes (KMeans, EM et SOM).*

tiques (Forestier et al., 2010) plus complexes pour mieux parcourir l'espace de recherche des solutions. La figure 1 présente un exemple de collaboration entre trois résultats de clustering obtenus avec trois méthodes différentes : KMeans, EM et SOM. Elle illustre les états respectifs des résultats avant et après collaboration pour chacune des méthodes. La forte similarité des résultats obtenus après collaboration, malgré la diversité des méthodes mises en œuvre, atteste de la capacité de notre approche à faire collaborer des méthodes de clustering différentes.

3 Discussion

Le clustering collaboratif est un domaine récent dont les fondements commencent à être posés. La multiplication des travaux s'intéressant à la combinaison et à l'intégration de plusieurs résultats de clustering montrent l'intérêt de la communauté scientifique pour ces approches.

Dans le cadre de nos travaux, nous avons proposé une méthode collaborative permettant l'échange d'information entre des résultats de clustering produits par des méthodes différentes. Cependant, elle n'adresse pas directement le problème de l'apprentissage collaboratif. En effet, un des prochains challenges à résoudre consistera à échanger des informations directement pendant l'étape d'apprentissage des modèles (calcul des centroides pour KMeans, création de la carte pour SOM, etc.). Il conviendra alors de développer des processus génériques permettant l'échange d'informations entre les méthodes.

Enfin, un autre défi concerne l'utilisation de méthodes de clustering collaboratif sur plusieurs sources de données fortement hétérogènes. En effet, bien que certaines méthodes s'intéressent au traitement de données multivues (données représentées par différents ensembles d'attributs), ces données ont toujours le même révérenciel, c'est-à-dire le même nombre d'objets. Ainsi, un autre enjeu consistera à étudier comment différentes vues plongées dans des référentiels différents, pourraient être utilisées conjointement dans un processus collaboratif. Nos travaux (Wemmert et al., 2009) ont proposé des solutions dans le cadre du traitement collaboratif d'images de télédétection à différentes résolutions (*i.e.* taille des images différentes), nous poussant ainsi à étudier la définition de critères de *cohérence* entre résultats de clustering. Des critères génériques à tout type de données sont encore à proposer.

Références

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

Forestier, G. (2010). *Connaissances et clustering collaboratif d'objets complexes multisources*. Ph. D. thesis, Université de Strasbourg.

Forestier, G., C. Wemmert, et P. Gancarski (2010). Towards conflict resolution in collaborative clustering. In *IEEE International Conference on Intelligent Systems*, pp. 361–366.

Grozavu, N. et Y. Bennani (2010). Classification collaborative non supervisée. In *Conférence francophone sur l'apprentissage automatique (CAP)*.

Handl, J. et J. Knowles (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation 11*(1), 56–76.

Kuncheva, L. I. (2008). Classifier ensembles : Facts, fiction, faults and future. In *International Conference on Pattern Recognition, Plenary talk*.

Pedrycz, W. (2007). Collaborative and knowledge-based fuzzy clustering. *International Journal of Innovative, Computing, Information and Control 1*(3), 1–12.

Strehl, A. et J. Ghosh (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research 3*, 583–617.

Wemmert, C., A. Puissant, G. Forestier, et P. Gancarski (2009). Multiresolution remote sensing image clustering. *IEEE Geoscience and Remote Sensing Letters 6*, 533 – 537.

Summary

Collaborative clustering consists to make jointly collaborate several clustering methods in order to improve their results. The different methods involved in the collaboration share their informations and question their decisions according to the decisions proposed by the other methods. An important challenge is to make collaborate different clustering methods while insuring that each point of view is taken into consideration. This article present the main approaches in collaborative clustering and present our work in that field. Furthermore, some emerging challenges are also presented.

Index des auteurs

XVIIIèmes Rencontres de la Société Francophone de Classification

28-30 Septembre 2011, Orléans

La SFC organise, chaque année, les Rencontres de la Société Francophone de Classification, qui ont pour objectifs de présenter des résultats récents, ou des applications originales, en classification ou dans des domaines connexes, de favoriser les échanges scientifiques à l'intérieur de la société et de faire connaitre à divers partenaires extérieurs les travaux de ses membres. Durant ces rencontres est attribué le prix Simon Régnier, consacrant une contribution originale d'un jeune chercheur à la classification.

En 2011, l'Université d'Orléans par le biais des laboratoires LIFO et MAPMO organise ces dix-huitièmes Rencontres de la Société Francophone de Classification qui se dérouleront du 28 au 30 septembre 2011 sur le campus de la Source à Orléans.