XIVes Rencontres de la Société francophone de classification

SFC'2007

Recueil des résumés

Paris, 5, 6 et 7 septembre 2007

Comité scientifique

Olivier Hudry (président), Irène Charon, Georges Hébrail (co-présidents)

Farid Béninel
Patrice Bertrand
Paula Brito
François Brucker
Bernard Burtschy
Marie Chavent
Guy Cucumel
Christian Derquenne
Jean Diatta
Pierre Gançarski
Alain Guénoche
André Hardy
François-Xavier Jollois
Bruno Leclerc
Monique Noirhomme
Gilbert Ritschard
Rosanna Verde
François Yvon

Coheris Spad

www.enst.fr/SFC2007

TELECOM PARIS
école nationale supérieure des télécommunications

RESUMES DES CONFERENCES INVITEES

Stabilité et validité d'un partitionnement
Patrice Bertrand 1

Probabilistic approaches to symbolic data analysis
Hans Bock 3

Learning from data streams: an overview
Joao Gama 10

Sur quelques approches de classification non supervisée de graphes
Pascale Kuntz 11

Classification et axes principaux : revue, passerelles, propositions
Ludovic Lebart 12

Classification de séquences génomiques sur base d'occurrences de motifs de régulation
Jacques Van Helden 14

RESUMES DES COMMUNICATIONS

(les communications sont triées par ordre alphabétique du nom du premier auteur)

Un indicateur de qualité pour les règles d'association extraites à partir d'une matrice de données symboliques
Filipe Afonso 16

Amélioration du Boosting par combinaison des hypothèses antérieures
Emna Bahri, Nicolas Nicoloyannis, Mondher Maddouri 20

La discrimination logistique étendue à un cas de mélange de deux sous populations
Farid Beninel 24

Application de méthodes de classification sur des vitesses métrologiques de dégradation de compteurs d'eau
Frédéric Bertrand, Myriam Maumy 27

Classification basée sur l'agrégation d'opinions par la méthode de recuit simulé
Mounzer Boubou, Ahmed Bounekkar, Michel Lamure 31

Classification avec des Multigraphes. Application à la classification de données décrites par des variables de différents types
Helena Brás Silva, Paula Brito, Joaquim Pinto da Costa 35

Méthode de stabilisation par rééchantillonnage dans les nœuds pour construire des arbres de classification
Bénédicte Briand, Catherine Mercat-Rommens, Gilles Ducharme 39

Sélection non supervisée d'attributs - Application à l'indexation d'images satellitaires
Marine Campedel, Yvan Kyrgyzov, Henri Maître 43

Une approche divisive de classification hiérarchique de variables quantitatives
Marie Chavent, Vanessa Kuentz, Jérôme Saracco 47

Mesures de tendance centrale et de dispersion d'une série d'intervalles
Marie Chavent, Jérôme Saracco 51

Exploitation des données d'enquêtes avec des méthodes d'analyse de données symboliques
Mohamed Cherif Rahal, Filipe Afonso, Myriam Touati, Edwin Diday, Anne Peradotto, Yasmina Quatrain, Sylvaine Nugier, Martine Hurault-Plantet 54

Classification recouvrante avec pondération locale des attributs
Guillaume Cleuziou 58

Indices de similarité structurelle sur des données arborescentes
Noël Conruyt, David Grosser, Henri Ralambondrainy 62

Une méthode de partition des images de lésions mélanocytes cutanées
Valentina Cozza, Mario Rosario Guarracino, Rosanna Verde 66

CrossStream : résumé de flux relationnels
Baptiste Csernel, Fabrice Clérot, Georges Hébrail 70

Différences, distances entre textes sanskrits, élaboration d'édition critique
Marc Csernel, Patrice Bertrand 74

La morphologie mathématique : un outil pour la classification des données multidimensionnelles et des hyper rectangles
F. A. T. De Carvalho, Jean-Paul Rasson 78

Une méthode de partitionnement sur un ensemble de tableaux de distances
F.A.T De Carvalho, Yves Lechevallier 79

Quelques remarques sur la méthode d'ajustement de Mayer
Antoine de Falguerolles 83

Partitionnement des arcs d'un graphe pour la planification de réseaux optiques
Lucile Denoeud, Nicolas Puech 87

La dissimilarité de bipartition et son utilisation pour détecter les transferts horizontaux de gènes
Alpha Boubacar Diallo, Alix Boc, Vladimir Makarenkov 90

Règles d'association dans un contexte de descriptions ordonnées
Jean Diatta, Henri Ralambondrainy, André Totohasina 94

Nonnegative matrix factorization algorithms: Tweedie quasi-likelihoods approach
Simplice Dossou-Gbété 97

Génération de bases pour les règles d'association M_GK-valides
Daniel Feno, Jean Diatta, André Totohasina 101

Risque structurel et sélection d'instances pour la règle du plus proche voisin
Sylvain Ferrandiz 105

Une nouvelle méthode de classification pour des données intervalle
André Hardy, Nathanaël Kasoro 109

Intégration d'information biologique dans le traitement de données Xomiques
François Husson, Marie de Tayrac, Marc Aubry, Jean Mosser, Sébastien Lê 113

Une approche incrémentale d'une méthode de classification non supervisée par nuages d'insectes volants
Julien Lavergne, Hanane Azzag, Christiane Guinot, Gilles Venturini 117

Construction d'arbres à partir de relations d'intermédiarité. Application au stemma codicum
Marc Le Pouliquen, Jean-Pierre Barthélémy 121

FactoMineR, une librairie de fonctions R en analyse des données pour l'enseignement et la recherche
Sébastien Lê, Julie Josse, François Husson....133

Médianes et unanimité dans les treillis
Bruno Leclerc....137

Sur les différentes expressions formelles d'une hiérarchie binaire symétrique ou implicative
Israël César Lerman....139

Visualisation de clusters dans les espaces de grande dimension
Sylvain Lespinats, Bernard Fertil, Jeanny Hérault....143

Clustering de nuages de points stéréoscopiques : une comparaison de différents paradigmes
Nicolas Loménie, François-Xavier Jollois....147

Classification et analyse textuelle : l'approche topologique
Sylvie Mellet, Xuan Luong, Dominique Longree, Jean-Pierre Barthélémy....149

Sélection de modèles prévisionnels par analyse de données symboliques
Omar Merroun, Edwin Diday, Alain Dessertaine, Philippe Rigaux, Estelle-Sarah Eliezer....153

Classification de parcours de vie à l'aide de l'optimal matching
Nicolas S. Müller, Matthias Studer, Gilbert Ritschard....157

Etude de la classification des bactériophages
Dung Nguyen, Alix Boc, Abdoulaye Banire Diallo, Vladimir Makarenkov....161

Un algorithme non itératif pour la classification d'observations bidimensionnelles
Nicolas Paul, Michel Terre, Luc Fety....165

Extraction de concepts et de relations en analyse relationnelle de concepts (ARC)
M.H. Rouanne, M. Huchard, A. Napoli, P. Valtchev....169

Classification automatique pour la segmentation de signaux unidimensionnels
Allou Samé, Patrice Aknin, Gérard Govaert....174

Reconnaissance de dissimilarités de Robinson
Morgan Seston....178

Partitionnement par colonies de fourmis et essaims particulaires
Javier Trejos, Eduardo Piza, Alex Murillo, Mario Villalobos....182

Une approche basée sur les treillis de Galois pour la construction des réseaux de neurones
Norbert Tsopze, Engelbert Mephu Nguifo, Gilbert Tindo....186

INDEX

Index par auteurs....190

Stabilité et validité d'un partionnement

P. Bertrand

ENST Bretagne, Dpt Lussi, 2 rue de la Châtaigneraie, 35576 Cesson Sévigné
patrice.bertrand@enst-bretagne.fr

Mots clés : stabilité, validité, rééchantillonnage, partitionnement.

La classification non supervisée constitue l'un des principaux outils d'exploration en analyse des données, en particulier dans le cas où aucune hypothèse a priori n'est faite sur le jeu de données étudié. Cependant, une difficulté majeure est que la plupart des méthodes de classification non supervisée proposent une structure en classes bien que l'existence même des classes ne soit pas toujours garantie. De plus, malgré l'usage intensif de ces méthodes et l'abondante litérature qui leur est consacrée, relativement peu de résultats théoriques concernent la validation des classes obtenues. Pour valider une classification, une approche empirique fréquemment appliquée et qui connaît actuellement un regain d'intérêt, consiste à se baser sur l'évaluation de la stabilité des classes obtenues. Différent scores de stabilité d'une partition ont ainsi été récemment proposés. Ces scores diffèrent entre eux selon les choix effectués pour perturber les données, et pour estimer la stabilité des partitions obtenues sur les données perturbées. Plusieurs auteurs ont proposé d'utiliser ces scores pour déterminer un nombre optimal de classes d'un partitionnement (voir par exemple [3], [4], [5], [7] et [9]). Par ailleurs, differents papiers ont montré que le comportement asymptotique d'un tel score n'est pas en relation avec le degré de validité du partionnement considéré lorsque la taille des données tend vers l'infini (cf. [6] et [1]). Ici, nous nous référons plus précisément aux propriétés asymptotiques établies formellement, pour une vaste classe d'algorithmes de classification non supervisée, par Ben-David *et al.* (cf. [1], [2]). Dans cet exposé, nous discuterons l'intérêt de ces propriétés asymptotiques dans le cas de la validation d'un partionnement qui est obtenu sur un jeu de données réel. En particulier, nous présenterons quelques conclusions qui seront justifiées expérimentalement par des résultats obtenus sur des jeux de données de tailles finies, réels ou simulés, et par un résultat théorique récemment obtenu par O. Shamir et N. Tishby ([8]).

[1] S. Ben-David, U. von Luxburg, D. Pal, "A Sober Look at Clustering Stability", *Proceedings of the 19th Annual Conference on Learning Theory, Carnegie Mellon University, USA*, 2006, 5-19.
[2] S. Ben-David, D. Pal, H. U. Simon, "Stability of k-Means clustering", *preprint.*
[3] A. Ben-Hur, A. Elisseeff, I. Guyon, "A stability based method for discovering structure in clustered data", *Pacific Symposium on Biocomputing*, 7, 2002, 6-17.
[4] A. Bertoni, G. Valentini, "Model order selection for bio-molecular data clustering", *BMC Bioinformatics*, 8(Suppl 2), 2007.
[5] P. Bertrand, G. Bel Mufti, "Loevinger's measures for assessing cluster stability", *Computational Statistics and Data Analysis*, 50/4, 2006, 992-1015.
[6] A. Krieger, P. Green, "A cautionary note on using internal cross validation to select the number of clusters", *Psychometrika*, 64/3, 1999, 341-353.

[7] E. Levine, E. Domany, "Resampling method for unsupervised estimation of cluster validity", *Neural Computation*, 13/11, 2001, 2573-2593.
[8] O. Shamir, N. Tishby, "Cluster Stability for Finite Samples", *preprint*.
[9] R. Tibshirani, G. Walther, T. Hastie, "Estimating the Number of Clusters in a Data Set via the Gap Statistic", *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 63/2, 2001, 411-423.

Probabilistic approaches for symbolic data analysis

Hans H. Bock

Institute of Statistics, RWTH Aachen University, 52056 Aachen, Germany
(bock@stochastik.rwth-aachen.de)

Keywords : Symbolic data, probability models, statistics for set-valued data.

1. Introduction

Starting with the seminal paper by Diday (1988), there is a large number of publications, reports and software tools dealing with the analysis of what is called 'symbolic data', i.e. collections of data vectors whose components are not (only) real numbers or labels (categories) as in classical statistics, but may be intervals, sets of categories, or (empirical, theoretical) frequency distributions. Most articles dealing with such data, e.g., with their descriptive characterization, the measurement of similarity and dissimilarity, clustering and discrimination methods, etc. (see, e.g., Bock and Diday 2000, Noirhomme and Diday 2007) proceed in an often surprisingly empirical (even if: suggestive) way without referring to any underlying formal structure or general principles as they are provided, e.g., by probabilistic models in classical multivariate statistics where interesting analytical results (e.g., on the performance of methods or large sample behaviour of estimators) could be derived. Even the relationship to approaches such as 'statistics for set-valued data', 'fuzzy methods', 'imprecise data theory', etc. is often neglected.

In this paper we point to concepts and methods that rely on probabilistic models and may be useful or even better alternatives to corresponding 'symbolic' approaches. Throughout we assume that $x_1, ..., x_n$ are n p-dimensional symbolic data vectors $x_k = (x_{k1}, ..., x_{kp})'$ and concentrate on the case of interval-type data where the entries $x_{kj} = [a_{kj}, b_{kj}]$ are intervals from $\mathbb{R}$ with upper/lower boundaries $a_{kj} \leq b_{kj}$ $(j = 1, ..., p,\ k = 1, ..., n)$ such that the k-th recorded entity (object) is represented by a hyper-rectangle (hypercube, rectangle, interval) $Q_k = [a_{k1}, b_{k1}] \times \cdots \times [a_{kp}, b_{kp}]$ in $\mathbb{R}^p$.

2. Minimum volume hypercubes

Consideration of symbolic data is often motivated by the need or intention to characterize the properties of a *group* G of g individuals for which we have sampled g single-valued classical data vectors $y_1, ..., y_g \in \mathbb{R}^p$, by an interval-type vector x_k (e.g., in order to maintain anonymity). So the first question of SDA might be how to find the characteristic boundaries a_{kj}, b_{kj} within x_k from the individual vectors $y_s = (y_{s1}, ..., y_{sp})'$, $s = 1, ..., g$. A range of different methods have been be proposed or implemented:

• the *'min/max option'* with $a_{kj} = \min_{s \in G} y_{sj}$, $b_{kj} = \max_{s \in G} y_{sj}$ such that x_k (i.e., Q_k) is the minimum hypercube in $\mathbb{R}^p$ that contains all individual vectors $y_1, ..., y_g$

• the *'confidence interval option'* where the component-specific intervals are given by $x_{kj} = [m_j - u \cdot s_j, m_j + u \cdot s_j]$ where $m_j := (\sum_{s \in G} y_{sj})/g$ and $s_j^2 := (\sum_{s \in G} (y_{sj} - m_j)^2)/(g-1)$ are the mean and empirical variance of the g data values of variable j and u an appropriate quantile of the t_{n-1} distribution

• the *'quantile option'* where a_{kj}, b_{kj} are the lower and upper empirical β-quantile of the g data values $y_{1j}, ..., y_{gj}$ (typically, the quartiles with $\beta = 1/4$); then x_k contains at least

$100 \cdot (1 - 2p\beta)$ % of the data points of G

• the *'$(1-\alpha)$-coverage option'* where x_k is the smallest hypercube in $\mathbb{R}^p$ that contains $100 \cdot (1-\alpha)$ % of the data points $y_1, ..., y_g$.

Whereas the min/max option is quite sensitive to outliers and will typically result in excessively large hypercubes for a large number g of individuals, the $(1-\alpha)$-coverage option is robust in this respect and insofar much more adequate when defining 'symbolic data vectors' (but with a high computational complexity). The following definitions show that it is intimately related to various concepts of classical probability where the empirical distribution of the values $y_1, ..., y_g$ is replaced by the distribution P of a p-dimensional random vector Y with density f in $\mathbb{R}^p$ (w.r.t. the Lebesgue measure λ_p).

Def. 1: A *minimum volume set* (*modal set*) for Y is a measurable subset $S \subset \mathbb{R}^p$ that resolves the minimization problem:

$$\lambda_p(S) \rightarrow \min_{S \subset \mathbb{R}^p} \quad \text{constrained by} \quad P(Y \in S) \geq 1 - \alpha \tag{1}$$

where $\alpha > 0$ a given threshold (e.g., $\alpha = 0.05$).

It was shown by Nuñez-Garcia et al. (2003) that all level sets $A_c := \{y \in \mathbb{R}^p \mid f(y) \geq c\}$ are minimum volume sets (for the threshold $\alpha := P(Y \in A_c)$). They also determine conditions under which the inverse statement holds. Note that modal sets are related to possibility theory as well as to random set theory (see Nuñez-Garcia et al., 2003), and also to the 'high-density' or 'contour clusters' of Bock (1974, 1996a) and Hartigan (1975). Estimation methods are described in Scott and Nowak (2006).

In the context of symbolic data analysis, the following modification might be considered:

Def. 2: A *minimum volume hypercube* for Y is an interval $S = [a, b] \subset \mathbb{R}^p$ that resolves the minimization problem:

$$\lambda_p(S) = \lambda_p([a,b]) \rightarrow \min_{a \leq b} \quad \text{constrained by} \quad P(a \leq Y \leq b) \geq 1 - \alpha. \tag{2}$$

A third definition of an optimum or 'typical hypercube' for Y (with distribution P) has been considered by Käärik and Pärna (2007). Starting from a distance measure $d(y, x)$ between points $x, y \in \mathbb{R}^p$ (typically the Euclidean distance), they define the minimum distance between $x \in R^p$ and a set $Q \subset \mathbb{R}^p$ by $D(x, Q) := \min_{y \in Q} d(x, y)$ $(= 0$ for $x \in Q)$ and look for a solution of the optimisation problem

$$E[D(X,Q)] = \int_{\mathbb{R}^p} min_{y \in Q} \{d(x,y)\} \, dP(x) \quad \rightarrow \quad \min_{Q \in \mathcal{Q}} \tag{3}$$

under the constraint that Q has a given coverage $P(Y \in Q) = \beta$ (as an alternative: a given volume $\lambda_p(Q) = v$) and belongs to a given (sufficiently large) family $\mathcal{Q}$ of subsets Q from $\mathbb{R}^p$ (typically: balls, hypercubes, unions of such sets,...).

Def. 3: A *prototype hypercube* Q of level β for Y is any hypercube $Q = [a, b] \subset \mathbb{R}^p$ that resolves (3) with $\mathcal{Q}$ the set of all intervals in $\mathbb{R}^p$.

In our lecture we will illustrate the structure of the optimum intervals for the simple case of a normally distributed (unimodal) random vector $Y = (Y_1, Y_2)$ with two independent components $Y_1, Y_2 \sim \mathcal{N}(0, 1)$ and comment on the implications for SDA.

3. Average intervals and class prototypes

A related problem concerns the definition of an 'average' and a 'variance' of n data intervals $Q_k = [a_k, b_k]$ $(k = 1, ..., n)$. There exist various approaches in mathematics, statistics, and geometric probability.

a) Centrocubes as optimum class representatives
A basic approach starts from a dissimilarity measure $d(Q, G)$ between two intervals Q, G and defines the '*average interval*' ('*class prototype*, '*centrocube*') as an interval $G = [u, v] \subset I\!R^p$ with minimum average deviation in the sense

$$g(C, G) := \sum_{k \in C} d(Q_k, G) \;\rightarrow\; \min_G \tag{4}$$

(or with some modification therof). Typically there will be no explicit solution of this problem, but for some special choices of d an exact solution can be easily obtained (see Chavent & Lechevallier 2002, Chavent 2004, Bock 2005). The minimum value in (4) can be used as a variance measure.

b) Approaches starting from geometric probability
In the framework of geometric probability there exist various proposals to define the average of a random set, based on a measure-theoretical definition of a 'random (closed) set Q' in $I\!R^p$ and its distribution P (see, e.g., Mathéron 1975)[1]. The 'expectation' $E[Q]$ of Q is then defined in a way that retains some useful properties of the classical concept of the 'expectation of a random variable' or maintains their validity at least in an extended sense. A wealth of definitions and properties are presented and surveyed, e.g., in Molchanov (1997), Baddeley and Molchanov (1997, 1998), and Nordhoff (2003), e.g.:

Def. 4: The '*Aumann expectation*' of Q is defined by

$$E_{Au}[Q] := \{\; E[Y] \mid Y \text{ is a selection of } Q \;\text{ with } E[||Y||] < \infty] \;\} \;\subset\; I\!R^p \tag{5}$$

where 'a selection of Q' is any random vector Y in $I\!R^p$ with $Y \in Q$ almost surely.

A related variance definition for one-dimensional intervals is provided by Kruse (1987).

c) A parametric approach for defining an average interval
Basically, a hypercube $Q = [a, b]$ is characterized by its 'lower' and 'upper' vertices $a, b \in I\!R^p$, and equivalently by its midpoint $m = (a + b)/2$ and the vector of (half) side lengths $\ell = (b - a)/2$. Therefore a *random hypercube* Q is specified by its random midpoint $M \in I\!R^p$ and its random side length vector $L \in R^p_+$ with a joint distribution $P^{M,L}_\vartheta$ (eventually parametrized by a parameter ϑ). The expected midpoint and side length vectors $\mu := E[M] = (\widetilde{\mu}_1, ..., \widetilde{\mu}_p)'$ and $\lambda := E[L] = (\widetilde{\lambda}_1, ..., \widetilde{\lambda}_p)'$ are used in

Def. 5: The '*parametric average interval*' of Q is given by

$$\widetilde{E}[Q] \;:=\; [\; E[M] - u \cdot E[L],\; E[M] + u \cdot E[L] \;] \;=\; [\; \mu - u \cdot \lambda, \mu + u \cdot \lambda \;] \tag{6}$$

where $u > 0$ is a specified constant (typically: $u = 1$).

We illustrate Def. 5 by assuming that all $2p$ components in $M = (\widetilde{M}_1, ..., \widetilde{M}_p)'$ and $L = (\widetilde{L}_1, ..., \widetilde{L}_p)'$ are stochastically independent with $\widetilde{M}_j \sim \mathcal{N}(\widetilde{\mu}_j, \sigma^2)$, $\widetilde{L}_j \sim \Gamma(a_j, b_j)$ and $\widetilde{\lambda}_j = E[\widetilde{L}_j] = a_j/b_j$ for $j = 1, ..., p$ such that (M, L) has a distribution density of the

[1] In the case of SDA, the empirical probability measure on $\{Q_1, ..., Q_n\}$ may be considered, assigning mass $1/n$ to each hypercube.

form $f(m,\ell;\mu,\sigma^2,\alpha,\beta) = h_1(m;\mu,\sigma^2)\cdot h_2(\ell;\alpha,\beta)$ with $\alpha=(a_1,...,a_p), \beta=(b_1,...,b_p)$ (distribution type $\mathcal{I}(\mu,\sigma^2;\alpha,\beta)$, say). Then the parametric average interval of Q is given by

$$\widetilde{E}[Q] := [\ \widetilde{\mu}_1 - u\frac{a_1}{b_1},\ \widetilde{\mu}_1 + u\frac{a_1}{b_1}\ ] \times \cdots [\ \widetilde{\mu}_p - u\frac{a_p}{b_p},\ \widetilde{\mu}_p + u\frac{a_p}{b_p}\ ]. \tag{7}$$

In practice, we have to estimate the unknown parameters $\mu,\sigma^2,\alpha,\beta$ from n independent samples $Q_1,...,Q_n$ of Q. If $m_1,...,m_n \in \mathbb{R}^p$ are the observed midpoints and $\ell_1,...,\ell_n$ the observed midrange vectors of $Q_1,...,Q_n$ with $\ell_k = (\widetilde{\ell}_{k1},...,\widetilde{\ell}_{kp})'$, the m.l. estimates are given by

$$\hat{\mu} = \bar{m} := \frac{1}{n}\sum_{k=1}^{n} m_k, \quad \hat{\sigma}^2 = \frac{1}{np}\sum_{k=1}^{n} ||m_k - \bar{m}||^2, \quad \widehat{\widetilde{\lambda}}_j = \bar{\ell}_j := \frac{1}{n}\sum_{k=1}^{n} \widetilde{\ell}_{kj} \tag{8}$$

whereas the estimates $\hat{a}_j$ and $\hat{b}_j$ are the solutions of the m.l. equations

$$\ln \hat{a}_j - \psi(\hat{a}_j) = ln\left(\bar{\ell}_j/\ell_j^*\right) \qquad \hat{b}_j = \hat{a}_j/\bar{\ell}_j \tag{9}$$

where $\ell_j^* := (\Pi_k \widetilde{\ell}_{kj})^{1/n}$ is the geometric mean of $\widetilde{\ell}_{1j},...,\widetilde{\ell}_{nj}$ and $\psi(z) := \Gamma'(z)/\Gamma(z)$ the digamma function (for details see, e.g., Johnson et al. (1994), pp. 360, or Kotz et al. (2006), p. 2625).

If we replace the Gamma distribution for the side lengths $\widetilde{L}_j$ by a uniform distribution in an interval $[0,\Delta_j]$ from $\mathbb{R}^p_+$, the m.l. estimate of the boundary Δ_j is given by $\widehat{\Delta}_j := \max_k\{\widetilde{\ell}_{kj}\}$.

4. Parametric probabilistic clustering models for interval data

Parametric clustering models for n hypercubes $Q_1,...,Q_n$ can be formulated along the lines illustrated in section 3, either as a 'fixed-partition' model, a 'random partition' model, or a mixture model (see Bock 1996a, 1996b, 1996c). As an example let us consider the following *symbolic fixed-partition model* for the random intervals $Q_1,...,Q_n$ characterizing the n objects in $\mathcal{O} := \{1,...,n\}$:

(1) There exists an unknown partition $\mathcal{C} = (C_1,...,C_m)$ of $\mathcal{O}$ with a known number m of classes $C_i \subset \mathcal{O}$;
(2) For each class C_i there exist class-specific parameters μ_i,α_i,β_i with $\mu_i \in \mathbb{R}^p$ and $\alpha_i = (a_{i1},...,a_{ip})', \beta_i = (b_{i1},...,b_{ip})'$ in $\mathbb{R}^p_+$, and $\sigma^2 > 0$ such that
(3) all intervals Q_k from the same class C_i have the same distribution:

$$Q_k \sim \mathcal{I}(\mu_i,\sigma^2;\alpha_i,\beta_i) \qquad \text{for } k \in C_i.$$

Maximizing the likelihood with respect to the system $\vartheta = (\mu_1,...,\mu_m,\sigma^2,\alpha_1,...,\alpha_p,\beta_1,...,\beta_m)$ of all parameters and to the m-partition $\mathcal{C}$ yields the *m.l. clustering criterion*:

$$G(\mathcal{C},\vartheta) \ := \ \Pi_{i=1}^m \Pi_{k\in C_i}\ h_1(m_k;\mu_i,\sigma^2)\cdot h_2(\ell_k;\alpha_i,\beta_i) \ \rightarrow \ min_{\mathcal{C},\vartheta} \tag{10}$$

A solution, in particular an optimum m-partition, can be approximated by the classical *k-means-type algorithm*:

Starting from an initial partition $\mathcal{C} = (C_1,...,C_m)$
(a) we determine, in each class C_i, the m.l. estimates $\hat{\mu}_i$, $\hat{\alpha}_i$, $\hat{\beta}_i$, $\hat{\sigma}^2$ as in (8) and (9),

(b) then build m new classes $C_1^{(1)}, ..., C_m^{(1)}$ by assigning each object k (data interval Q_k) to the class with maximum likelihood such that for $i = 1, ..., m$:

$$C_i^{(1)} := \{\ k \in \mathcal{O} \mid f(m_k, \ell_k; \hat{\mu}_i, \hat{\sigma}^2, \hat{\alpha}_i, \hat{\beta}_i) = \max_{j=1,...,m} f(m_k, \ell_k; \hat{\mu}_j, \hat{\sigma}^2, \hat{\alpha}_j, \hat{\beta}_j)\ \}$$

(c) and iterate (a) and (b) until stationarity.

5. Probabilistic regression models for interval data

When extending classical regression methods to the symbolic interval data situation, we face the basic problem that it is not at all trivial to define a 'linear function of a hypercube', a 'linear structure' among n given hypercubes $Q_1, ..., Q_n$, a 'linear dependence among two hypercubes' etc. Therefore some of the proposed 'symbolic' regression methods proceed mainly in a heuristic, empirical way without an underlying general principle (see also De Carvalho et al. 2004, Neto et al. 2005).

In contrast, Billard and Diday (2000, 2002) propose a mathematical model for the two-dimensional case that mixes probabilistic aspects with empirical considerations as follows: They consider, within each rectangle Q_k, a uniform distribution with density f_k, say, and introduce a (virtual) two-dimensional random vector $Z = (X, Y)$ with the mixture distribution density $f(z) = (\sum_{i=1}^{n} \cdot f_k(z))/n$. Then the classical regression tools are applied the linear prediction model '$Y = a + bX + e$' for the random variables X, Y (under f), and the resulting parameters (expectations, variances, slope, correlation,...) are interpreted with a view to $Q_1, ...Q_n$ (see also Billard 2004). - Obviously, this model does not really describe a linear dependence between intervals or rectangles.

In contrast, Gil et al. (2001, 2007), González-Rodríguez et al. (2007) have proposed such a model in the framework of random set theory, with reference to the system $\mathcal{K}$ of all random convex compact sets in $I\!R^p$. They consider two alternative regression models for two random sets X, Y in $I\!R^p$:

Affine model 1:
There exists a fixed set $A \in \mathcal{K}$ and a scalar $b \in I\!R^p$ such that

$$Y = A \oplus bX := \{\ y = a + bx \mid a \in A, x \in X\ \} \tag{11}$$

Regression model 2:
There exists a fixed set $A \in \mathcal{K}$ and a scalar $b \in I\!R^p$ such that

$$Y|X = x \ \sim\ \epsilon_x \oplus bx \ =\ \{\ \eta + b\xi \mid \eta \in \epsilon_x, \xi \in x\ \} \tag{12}$$

where ϵ_x is a random set from $\mathcal{K}$ with Aumann expectation

$$E_{Au}[\ \epsilon_x \mid X = x\ ] \ =\ A \qquad \text{for all } x \in \mathcal{K}. \tag{13}$$

For both models the statistical problem consists in determining the set $A \in \mathcal{K}$ and the scalar $b \in I\!R$ such that the totality of predicted pairs $\{(x_k, A \oplus bx_k)\}_{k=1,...,n}$ is close to the totality of all observed pairs $\{(x_k, y_k)\}_{k=1,...,n}$ in the sense of

$$C(A, b) \ :=\ \sum_{k=1}^{n} D_W^2(y_k, A \oplus bx_k) \quad \to \min_{A,b} \tag{14}$$

where for model 2 the data (hypercube) pairs (x_k, y_k) are supposed to fulfil $y_k = \epsilon_k \oplus bx_k$ with a convex set $\epsilon_k \in \mathcal{K}$. Note that D_W is a suitable distance between convex sets as

defined, e.g., in González-Rodríguez et al. (2007). - Naturally, in the SDA context, the set A will be a fixed hypercube and X a random one. It would be useful to modify the existing results and algorithms to the case where both X and Y are hypercubes.

References

Baddeley, A.J., Molchanov, I.S. (1997): "On the expected measure of a random set." In: D. Jeulin (ed.): *Advances in theory and applications of random sets.* World Scientific, Singapore, 3-20.

Baddeley, A.J., Molchanov, I.S. (1998): "Averaging of random sets based on their distance functions." *Journal of Mathematical Imaging and Vision* 8, 79-92.

Billard, L. (2004): "Dependencies in bivariate interval-valued symbolic data." In: D. Banks, L. House, F.R. McMorris, Ph. Arabie, W. Gaul (eds.): *Classification, clustering, and data mining applications.* Springer, Heidelberg, 319-324.

Billard, L., Diday, E. (2000): "Regression analysis for interval-valued data." In: Kiers, H.A.L., Rasson, J.-P., Groenen, P.J.F., Schader, M. (eds.): *Data analysis, classification, and related methods.* Springer, Heidelberg, 2000, 369-374.

Billard, L., Diday, E. (2002): "Symbolic regression analysis." In: K. Jajuga, A. Sokolowski, H.-H. Bock (eds.): *Classification, clustering, and data analysis.* Springer, Heidelberg, 2002, 281-288.

Bock, H.-H. (1974): *Automatische Klassifikation.* Vandenhoeck & Ruprecht, Göttingen.

Bock, H.-H. (1996a): "Probability models and hypotheses testing in partitioning cluster analysis." In: Ph. Arabie, L. Hubert, G. De Soete (eds.): *Clustering and classification.* World Science, Singapore, 1996, 377-453.

Bock, H.-H. (1996b): "Probabilistic models in cluster analysis." *Computational Statistics and Data Analysis* 23, 5-28.

Bock, H.-H. (1996c): "Probabilistic models in partitional cluster analysis." In: A. Ferligoj, A. Kramberger (eds.): *Developments in data analysis.* FDV, Metodoloski zvezki, 12, Ljubljana, Slovenia, 1996, 3-25.

Bock, H.-H., Diday, E. (eds.) (2000): *Analysis of symbolic data. Exploratory methods for extracting statistical information from complex data.* Springer, Heidelberg, 2000.

Bock, H.-H. (2003): "Clustering methods and Kohonen maps for symbolic data." *J. of the Japanese Society of Computational Statistics* 15 (2), 217-229.

Bock, H.-H. (2005): "Optimization in symbolic data analysis: dissimilarities, class centers, and clustering." In: D. Baier, R. Decker, L. Schmidt-Thieme (eds.): *Data analysis and decision support.* Springer, Heidelberg, 3-10.

Bock, H.-H (2007): "Analyzing symbolic data: problems, methods, and perspectives." In: A. Okada, T. Imaizumi, W. Gaul, H.-H. Bock (eds.): *Proc. of the German Japanese Workshops in Tokyo and Berlin 2005/2006.* Springer, Heidelberg, 2007 (to appear).

Bock, H.-H., Diday, E. (eds.): *Analysis of symbolic data.* Springer, Heidelberg.

Chavent, M. (2004): "A Hausdorff distance between hyperrectangles for clustering interval data." In: D. Banks, L. House, F.R. McMorris, P. Arabie and W. Gaul (eds.): *Classification, clustering, and data mining applications.* Springer, Heidelberg, 333-339.

Chavent, M., Lechevallier, Y. (2002): "Dynamical clustering of interval data: Optimization of an adequacy criterion based on Hausdorff distance." In: K. Jajuga, A. Sokołowski and H.-H. Bock (eds.): *Classification, clustering, and data analysis.* Springer, Heidelberg, 53-60.

De Carvalho, F., Neto, E., Tenorio, C.P. (2004): "A new method to fit a linear regression model to interval data." In: S. Biundo, Frühwirth, T.W., Palm, G. (eds.) *KI 2004: Advances in artificial intelligence.* Springer, Heidelberg, 295-306.

Diday, E., Noirhomme, M. (eds.) (2007): *Symbolic data analysis and the SODAS software.* Wiley, New York. (In print)

Gil, M.A., López-García, M.T., Lubiano, M.A., Montenegro, M. (2001): "Regression and correlation analysis of a linear relation between random intervals." *Test* 10, 183-201.

Gil, M.A., González-Rodríguez, G., Colubi, A., Montenegro, M. (2007): "Testing linear independence in linear models with interval-valued data." *Computational Statistics and Data Analysis* 51, 3002-3015.

González-Rodríguez, G., Blanco, A., Corral, N., Colubi, A. (2007): "Least squares estimation of linear regression models for convex compact random sets." *Advances in Data Analysis and Classification* 1, 67-81.

Hartigan, J. (1975): *Clustering algorithms.* Wiley, New York.

Johnson, N.L., Kotz, S., Balakrishnan, N (1994): *Continuous univariate distributions.* Vol. 1. Wiley, New York.

Käärik, M., Pärna, K. (2007): "Approximating distributions by bounded sets." *Acta Applicandae Mathematica* 97, 15-23.

Kotz, S., Balakrishnan, N., Read, C.B., Vidakovic, B. (2006): *Encyclopedia of statistical sciences.* Vol. 4. Wiley, New York.

Kruse, R. (1987): "On the variance of random sets." *J. of Mathematical Analysis and Applications* 122 (2), 469-473.

Mathéron, G. (1975): *Random sets and integral geometry.* Wiley, New York.

Molchanov, I. (1997): "Statistical problems for random sets. In: J. Goutsias (ed.): *Random sets: theory and applications.* Springer, Berlin, 27-45.

Nordhoff, O. (2003): *Expectation of random intervals.* Diploma thesis. Institute of Statistics, RWTH Aachen University.

Neto, E., de Carvalho, F., Freire, E.S. (2005): "Applying constrained linear regression models to predict interval-valued data." In: U. Furbach (ed.): *KI 2005: Advances in Artificial Intelligence.* Springer, Berlin Heidelberg, 92-106

Nuñez-Garcia, J., Kutalik, Z., Cho, K.-H., Wolkenhauer, O. (2003): "Level sets and minimum volume sets of probability density functions." *Intern. J. of Approximate Reasoning* 34, 25-47.

Scott, C.D., Nowak, R.D. (2006): "Learning minimum volume sets". *J. of Machine Learning Research* 7, 665-7044.

Learning From Data Streams: an Overview

Joao Gama

LIADD-INESC Porto, Univeristy of Porto
jgama@fep.up.pt

KeyWords : Data Streams, Machine Learning, Data Mining

Many sources produce data continuously. Examples include radio frequency identification, sensor networks, customer click streams, telephone records, scientific data, etc. Data is collected over time from dynamic and non stationary environments. These sources are called data streams. A data stream is an ordered sequence of instances that can be read only once or a small number of times using limited computing and storage capabilities. What distinguishes current data sets from earlier ones is automatic data feeds. We do not just have people who are entering information into a computer. Instead, we have computers entering data into each other. Nowadays there are applications in which the data is modeled best not as persistent tables but rather transient data streams. Examples of applications include network monitoring, web applications, sensor networks, telecommunications data management, financial applications, etc. In these applications it is not feasible to load the arriving data into a traditional DataBase Management System and traditional DBMS are not designed to directly support the required continuous queries. Data streams are increasingly important in the research community, as new algorithms, sampling strategies, and other approximate methods are needed to process this streaming data in reasonable time.

Desirable properties of Data Mining algorithms that learn from data streams involve: ability to incorporate new data, process examples at the rate they are available, constant time and memory to process each example. Moreover, the assumption that examples are generated at random according to a stationary probability distribution does not hold any more. In complex systems and for large time periods, we should expect changes in the distribution of the examples. Learning algorithms should be able to *forget* outdated data.

Learning from Data streams requires *adaptive learning algorithms*: incremental algorithms that take into account concept drift. In this talk we discuss the general setting of learning from data streams and present illustrative algorithms for hierarchical clustering and decision tree learning.

Sur quelques approches de classification non supervisée de graphes

P. Kuntz[1]

1. LINA-COD, Site Ecole Polytechnique de l'Université de Nantes, La Chantrerie, BP 50609, 44306 Nantes cedex 3
Pascale.Kuntz@polytech.univ-nantes.fr

Mots clés : graphes, classification, visualisation, Laplacien, noyau

On considère ici un ensemble fini de graphes $\{G_1, G_2, \ldots, G_n\}$ qui peuvent être selon les cas orientés ou non, et étiquetés ou non. Le problème général consiste à classer sans modèle préalable cet ensemble en regroupant les graphes qui se « ressemblent ». Rencontré initialement essentiellement en chimie ou en reconnaissance des formes pour améliorer les algorithmes de recherche d'information, ce problème se rencontre désormais dans des applications très variées (réseaux sociaux, structures de communication en réseaux, modélisation de documents, …). D'un point de vue combinatoire, la comparaison de graphes fait inévitablement référence aux délicats problèmes d'isomorphisme. Nous n'abordons pas cet aspect ici, et nous plaçons dans une démarche classique d'« analyse des données » où nous cherchons à décrire les graphes par des descripteurs.

Dans une première partie, nous évoquons l'intérêt –et les limites – des outils récents de visualisation de graphes dans une démarche d'analyse exploratoire [2]. Dans une deuxième partie, nous présentons trois familles d'approches de construction de descripteurs : *(1)* une approche issue de la physique statistique qui consiste à définir des indicateurs structurels et fonctionnels (efficacité, robustesse, « clustering », …) [1], *(2)* une approche en plein essor en fouille de données qui consiste à rechercher préalablement les sous-structures les plus fréquentes puis à décrire les graphes selon la présence/absence de ces structures [4,6], *(3)* une approche basée la décomposition spectrale du Laplacien discret [7]. Dans une troisième partie, nous rappelons le principe général des méthodes à noyau et présentons quelques résultats récents sur les noyaux définis sur des ensembles de graphes [3,5].

[1] R. Alpert, A.-L. Barabasi, "Statistical mechanics of complex networks", Rev. Mod. Phys. 47, 2002.
[2] B. Pinaud, P. Kuntz, F. Picarougne, "The Website for graph visualization software references (GVSR), 14th Int. Symp. on Graph Drawing, LNCS 4372, 2006.
[3] T. Gärdner, "A survey of kernels for structured data", SIGKDD Explorations, 2003.
[4] A. Inokuchi, T. Washio, H. Motoda, "An Apriori algorithm for mining frequent substructures from graph data", PKDD, 2000.
[5] H. Kashima, A. Inokuchi, "Kernels for graph classification", IEEE ICDM Workshop on Active Mining, 2002.
[6] K. Tsuda, T. Kudo, "Clustering graphs by weighted substructure mining", ICML, 2006.
[7] R. Wilson, E. Hancock, B. Luo, "Pattern vectors from algebraic graph theory", Preprint, 2005.

Classification et axes principaux : revue, passerelles, propositions

L. Lebart

CNRS - ENST, 46, rue Barrault, 75634 Paris cedex 13

Mots clés : Axes principaux, classification, graphes.

Les méthodes de classification et les principales techniques d'analyse en axes principaux (analyse en composantes principales, analyse des correspondances simples et multiples, analyses canoniques et discriminantes) se sont mutuellement enrichies et complétées au cours des 30 dernières années. On commence par esquisser un historique et un panorama des nombreux travaux situés à l'intersection des deux champs. On pourrait faire remonter les essais de conciliation des deux approches aux techniques de rotations de l'analyses factorielle classique qui cherchait à faciliter l'interprétation des axes par la recherche de conglomérats de variables. Des techniques de classification particulières comme l'analyse factorielle typologique [6] et la classification descendante de Reinert [11] sont fondées sur des algorithmes qui utilisent de façon itérative des calculs d'axes principaux, dans le but d'enrichir les résultats et leur interprétabilité. Des liens plus théoriques ont été tissés dans certains cas : pour les correspondances hiérarchiques [2], liens entre indices de classification hiérarchique et valeurs propres [5], [9]. D'autres méthodologies plus pragmatiques, souvent implémentées dans les logiciels du commerce, jouent sur la complémentarité des approches : classification à partir des coordonnées factorielles, projection *a posteriori* des classes sur les plans principaux, sur les axes canoniques de discrimination (pour la visualisation des classes), utilisation de la classification pour disséquer et décrire de façon automatique des sous-espaces continus. L'introduction de métriques locales ([1], [7]) permet de mettre en oeuvre des méthodes hybrides qui améliorent les visualisations des classifications et peuvent se rattacher aux procédures de projections révélatrices (*projection pursuit*) ([3], [4]). Les classifications par analyses spectrales des Laplaciens de graphes ont induit de nouvelles approches (*e.g.,* [2] ; [10]). Les graphes dérivés des calculs de plus proches voisins ou construits à partir de seuils de distances, par leurs propriétés spectrales et les métriques qu'ils peuvent définir, apportent un éclairage intéressant sur les liens existants entre ces deux grandes familles de méthodes.

[1] Art, D., Gnanadesikan, R., Kettenring, J.R., "Data based metrics for cluster analysis", *Utilitas Mathematica*, *21 a*, 1982, 75-99.

[2] Benzécri, J.P., *Analyse des Données: Correspondances*. Dunod, Paris, 1973.

[3] Burtschy B., and Lebart L., "Contiguity analysis and projection pursuit", in *Applied Stochastic Models and Data Analysis*, World Scientific , Singapore, 1991, 117-128.

[4] Caussinus H., "Projections Revelatrices" in *Modèles pour l'Analyse des Données Multidimensionnelles*, J.J. Droesbeke, B. Fichet, P.Tassi, eds, Economica, Paris, 1992.

[5] Cazes P., "Correspondance entre deux ensembles et partition de ces deux ensembles", *Cahiers de l'Analyse des Données*, vol.XI, no.3, 1986, 335-340.

[6] Diday E., "Introduction à l'analyse factorielle typologique". Revue de Statistique Appliquée, 22, no4, 1974, 29-38.

[7] Gnanadesikan R., Kettenring J.R., Landwehr J.M., "Projection Plots for Displaying Clusters", *in Statistics and Probability, Essays in Honor of C.R. Rao,* G. Kallianpur, P.R. Krishnaiah, J.K.Ghosh, eds, North-Holland, 1982.

[8] Lebart, L., "Assessing Self Organizing Maps via Contiguity Analysis". *Neural Networks*, *19*, 2006, 847-854.

[9] Lebart L., Mirkin B., "Correspondence Analysis and Classification". In*: Multivariate Analysis: Future Directions 2*. Cuadras C.M. and Rao C.R., (eds), North Holland, 1993, 341-357.

[10] Mohar B., "Some Applications of Laplace Eigenvalues of Graphs", *Graph Symmetry, Algebraic Methods and Application*, Hahn G., Sabidussi G., NATO Ser. C., 497, Kluwer, 1997, 225-275.

[11] Reinert M. "Une méthode de classification descendante hiérarchique: Application à l'analyse lexicale par contexte". *Cahiers de l'Analyse des Données*, 3, 1983, 187-198.

Classification de séquences génomiques sur base d'occurrences de motifs de régulation

Jacques van Helden[1]

1. Laboratoire de Bioinformatique des Génomes et des Réseaux. Université Libre de Bruxelles. Boulevard du Triomphe. Campus Plaine, CP 263. B-1050 Bruxelles. Belgique.
Jacques.van.Helden@ulb.ac.be

Mots clés : bio-informatique, analyse textuelle, discrimination, analyse multidimensionnelle.

Les régions génomiques non-codantes, souvent négligées dans les premières analyses de génomes, jouent cependant un rôle essentiel dans la fonction des gènes, en assurant la régulation de leur expression. La régulation transcriptionnelle repose sur la présence de signaux de courte taille (quelques nucléotides) reconnus de façon spécifiques par des protéines (les facteurs de transcription). Nous présenterons une revue une série d'approches bioinformatiques que nous avons développées afin de classifier les séquences d'ADN en fonction de la présence de signaux de régulation.

Nous disposons actuellement des génomes complets de plusieurs centaines d'organismes, mais nous sommes loin de comprendre le fonctionnement complet d'un seul de ces génomes. Le décryptage de l'information contenue dans l'ADN représente sans doute l'un des plus grand défis de la biologie contemporaine.

Les séquences d'ADN d'un génome portent l'essentiel de l'information génétique responsable du développement et du fonctionnement d'un organisme vivant. Le bon fonctionnement d'un génome exige une régulation extrêmement précise de l'expression des gènes. Cette régulation intervient notamment durant le développement embryonnaire, et est responsable de la différentiation morphologique des organes, tissus et cellules. Elle joue également un rôle essentiel dans la réponse métabolique des cellules, leur permettant d'exprimer des sous-ensembles spécifiques d'enzymes en fonction des molécules disponibles dans l'environnement.

La régulation transcriptionnelle (qui concerne la transcription de l'ADN en ARN) repose sur la capacité de certaines protéines (les facteurs transcriptionnels) à reconnaître de façon spécifique des signaux localisés dans les séquences génomiques non-codantes. Ces signaux peuvent être décrits sous forme de mots, d'expressions régulières, ou de modèles probabilistes plus fins (matrices score/position). Du fait du faible contenu informationnel de ces signaux de régulation, et de la taille énorme des génomes, les méthodes prédictives basées sur les occurrences d'un simple motif sont d'emblée vouées à retourner un taux élevé de faux positifs. La détection d'un site de liaison potentiel ne suffit donc pas à affirmer que les gènes avoisinants sont régulés par un facteur transcriptionnel. La régulation d'un bon nombre de gènes repose cependant sur la présence de sites multiples : répétition de sites pour un même facteur (modèles homotypiques), ou combinaisons de sites de liaison pour des facteurs distincts (modèles hétérotypiques). Ces régions enrichies en sites constituent des modules intégratives, permettant à plusieurs facteurs d'interagir de façon synergique ou antagoniste.

Une première étape consiste à identifier les motifs caractéristiques d'un processus donné, et qui correspondent généralement aux signaux spécifiquement reconnus par un ou plusieurs facteurs transcriptionnels. On peut ensuite utiliser des programmes de localisation de motifs séquentiels (pattern matching) pour détecter les occurrences de ces motifs dans les régions

régulatrices de chaque gène. La tâche suivante consiste à regrouper les gènes présentant des motifs similaires.

Dans certains cas, on dispose de groupes d'entraînement (constitué sur base de connaissances biologiques préalables), qui permettent d'appliquer des méthodes de classification supervisée. On se retrouve typiquement confronté aux problèmes de sur-dimensionnalité, car le nombre de gènes constituant les groupes d'entraînement est généralement faible (typiquement une dizaine de gènes), et le nombre de variables peut être élevé (quelques dizaines à quelques milliers de variables). Ce problème est contourné en recourant à des méthodes de sélection de variable. Pour l'évaluation de la qualité des classificateurs, la faible taille des groupes d'entraînement nous contraint à recourir au Leave-one-out.

Un autre type de problème est la découverte de classes de gènes sur base d'occurrences de motifs, qu'on réalise en appliquant des méthodes non-supervisées (clustering). La nature des données (comptages d'occurrences) pose un problème pour le choix de métriques de (dis)similarité. Dans nos conditions, les métriques classiques (distance binaire, euclidienne, de Mahalanobis, coefficient de corrélation, …) donnent de piètres résultats. Nous présenterons une série de métriques alternatives, basées sur les probabilités de Poisson, qui améliorent les résultats de clustering sur ce type de données.

Nous terminerons l'exposé en discutant de l'inférence de réseaux de co-régulation sur base de motifs découverts dans les séquences régulatrices.

[1] van Helden, J., Andre, B. & Collado-Vides, J. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. J. Mol. Biol. 281, 827-42.

[2] Simonis, N., Wodak, S. J., Cohen, G. N. & van Helden, J. (2004). Combining pattern discovery and discriminant analysis to predict gene co-regulation. Bioinformatics 20, 2370-9.

[3] van Helden, J. (2004). Metrics for comparing regulatory sequences on the basis of pattern counts. Bioinformatics 20, 399-406.

[4] Gonze, D., Pinloche, S., Gascuel, O. & van Helden, J. (2005). Discrimination of yeast genes involved in methionine and phosphate metabolism on the basis of upstream motifs. Bioinformatics 21, 3490-500.

[5] van Helden, J. (2005). The Analysis of Regulatory Sequences. In Multiple Aspects of DNA and RNA: from Biophysics to Bioinformatics (Monasson, R., ed.), Vol. SESSION LXXXII, pp. 00-24. Elsevier.

Un indicateur de qualité pour les règles d'association extraites à partir d'une matrice de données symboliques

F. Afonso

Ceremade, Paris Dauphine, Pce du Maréchal de Lattre de Tassigny, 75775 Paris cedex 16
afonso@ceremade.dauphine.fr

Mots clés : Analyse de Données Symboliques, Règles d'association

1 Introduction

Depuis (Agrawal et al., 1993), l'extraction de règles d'association à partir de grandes bases de données, notamment entre les produits présents dans le panier de la ménagère, a été un thème très étudié. Les articles du panier de la ménagère sont appelés items alors que les sous-ensembles d'items sont appelés itemsets. Une transaction est un itemset enregistré à la caisse d'un supermarché. Ainsi, en entrée de ces algorithmes, nous avons : un ensemble de n items $I = \{i_1, ..., i_n\}$; un ensemble de m transactions $T = \{t_1, ..., t_m\}$ avec $t_i \in P(I) - \emptyset$. Nous donnons un exemple de matrice de transactions table 1 avec 15 transactions qui sont des sous-ensembles de 5 items v = viande, p = poisson, c = céréales, f = fruits et légumes et l = produits laitiers. Une règle d'association est alors définie par deux itemsets X et Y tels que: $X \rightarrow Y$ avec $X \subset I$, $Y \subset I$ et $X \cap Y = \emptyset$. Par exemple, la règle d'association $c \wedge f \rightarrow p$ se lit "si les produits c = céréales et f = fruits et légumes se trouvent dans le panier de la ménagère alors il y a aussi le produit p = poisson".

Agrawal et Srikant suggèrent l'algorithme Apriori afin de générer les règles d'association ayant un support *sup* et une confiance *conf* supérieurs à deux seuils minimaux *minsup* et $minconf$ respectivement où (voir (Agrawal et Srikant, 1994)):
$sup(X \cup Y) = sup(X \rightarrow Y) = sup(Y \rightarrow X) = \frac{card(t \in T / X \cup Y \subseteq t)}{card(T)}$ = proportion des transactions contenant à la fois X et Y;
$conf(X \rightarrow Y) = \frac{card(t \in T / X \cup Y \subseteq t)}{card(t \in T / X \subseteq t)} = \frac{sup(X \cup Y)}{sup(X)}$ = proportion des transactions contenant Y parmi celles contenant X.

Dans cet article, nous allons étudier un indicateur de qualité des règles d'association découvertes à partir d'une matrice symbolique (matrice décrivant des concepts).

T	Client	X=items	T	Client	X=items	T	Client	X=items
t_1	1	v	t_6	1	v,f	t_{11}	3	v
t_2	1	v,p,c	t_7	2	v,p	t_{12}	4	p,c
t_3	1	v,p,c	t_8	2	v,p,c	t_{13}	4	p
t_4	1	v	t_9	2	v	t_{14}	4	p,l
t_5	1	v,f	t_{10}	3	v,p	t_{15}	4	p

Table 1: Matrice de transactions classique. Les colonnes "T=Transaction" et "X=items" constituent les données en entrée de l'algorithme Apriori standard

Concepts = Clients	X = items	Concepts = Clients	X = items
1	1/2v, 1/6p, 1/6c, 1/6f	3	2/3v, 1/3p
2	1/2v, 1/3p, 1/6c	4	2/3p, 1/6c, 1/6l

Table 2: Matrice de données symboliques composée d'une variable diagramme obtenue à partir de la matrice de transactions table 1

2 Règles d'association extraites à partir d'une matrice de concepts

Dans (Afonso, 2004 et 2005), nous étendons l'algorithme Apriori aux cas des données diagrammes. Concrètement, nous n'avons plus, dans notre matrice de données, une valeur unique par case ou bien un sous-ensemble d'items par transaction comme dans le cas classique. Nous avons un diagramme dans chaque case, i.e. des valeurs multiples pondérées telles que la somme des poids soit égale à un. Cet "Apriori symbolique" va nous permettre d'étudier des concepts. Nous nous appuierons sur l'exemple du panier de la ménagère afin d'en étudier les concepts clients. Nous constatons que les 15 transactions, répertoriées dans la matrice classique table 1, proviennent de 4 clients différents.

Pour appliquer l'analyse symbolique sur les concepts clients, nous supprimons la colonne des transactions (table 1) et nous créons ces concepts (table 2) qui seront les unités statistiques de notre étude. Pour chaque client, cette matrice agrège tous les items achetés sous forme d'un diagramme construit avec la proportion de chaque article par rapport aux achats totaux du client. Nous obtenons alors une matrice symbolique où chaque ligne définit la "description" d'un client et chaque colonne est associée à une variable symbolique. Pour plus d'information concernant l'analyse de données symboliques, nous pourrons nous référer à (Bock et Diday, 2000).

Dans (Afonso, 2004 et 2005), nous avons proposé un algorithme, appelé SAprori, permettant d'extraire des règles d'association à partir d'une matrice de concepts. Les règles d'association découvertes sont du type :

$$1/5 < Pv \leq 2/5 \wedge 0 < Pc \leq 1/5 \rightarrow 0 < Pp \leq 1/5,$$

qui se lit :"**Si pour un client donné**, la fréquence d'achat de viandes par rapport aux achats totaux est comprise entre 1/5 ouvert et 2/5; **et si** la fréquence d'achat de céréales est comprise entre 0 ouvert et 1/5; **alors** la fréquence d'achat de poissons est comprise entre 0 ouvert et 1/5".

Plus généralement, nous découvrons les règles du type $X \rightarrow Y$ avec: Ω un ensemble d'individus (concepts), $w \in \Omega$,

$$X(w) = \bigwedge_{i,u}\{\underline{x}_{i,u} < P_{x_{i,u}}(w) \leq \bar{x}_{i,u}\};\ Y(w) = \bigwedge_{j,v}\{\underline{y}_{j,v} < P_{y_{j,v}}(w) \leq \bar{y}_{j,v}\}$$

où $\forall i, u, j, v$, nous avons x_i et y_j des variables diagrammes non nécessairement distinctes; $x_{i,u}$ la modalité u de la variable x_i; $y_{j,v}$ la modalité v de la variable y_j; $x_{i,u} \neq y_{j,v}$; $P_{x_{i,u}}$ ($P_{y_{j,v}}$) la fréquence de la catégorie u (v) de la variable diagramme x_i (y_j); $\underline{x}_{i,u}$ et $\bar{x}_{i,u}$ ($\underline{y}_{j,v}$ et $\bar{y}_{j,v}$) les bornes des intervalles de fréquences.

Nous ne donnons pas ici la description de l'algorithme SApriori pour l'extraction des règles d'association à partir de matrices symboliques. On pourra se référer à (Afonso, 2004 et 2005). Dans la section suivante, nous étudions un indicateur de qualité de ces règles d'association.

N^o	Règles	SupS %	ConfS %	ConfD %
1	$1/3 < \mathrm{P}_v \leq 2/3 \rightarrow 0 < \mathrm{P}_p \leq 1/3$	75	100	75
2	$0 < \mathrm{P}_p \leq 1/3 \rightarrow 1/3 < \mathrm{P}_v \leq 2/3$	75	100	75
3	$0 < P_c \leq 1/3 \rightarrow 0 < \mathrm{P}_p \leq 2/3$	75	100	60
4	$0 < \mathrm{P}_p \leq 2/3 \rightarrow 1/3 < \mathrm{P}_v \leq 2/3$	75	75	56
5	$0 < \mathrm{P}_p \leq 2/3 \rightarrow 0 < \mathrm{P}_c \leq 1/3$	75	75	56

Table 3: Règles d'association symboliques découvertes à partir de la matrice symbolique table 2, $minsupS = 75\%$, $minconfS = 75\%$

3 Un indicateur de qualité pour les règles d'association extraites à partir d'une matrice de concepts

L'algorithme SApriori découvre des règles d'association pour un support minimum $minsupS$ et une confiance minimum $minconfS$ où le support noté $supS$ et la confiance notée $confS$ de la règle $X \rightarrow Y$ sont donnés par:

$SupS(X \rightarrow Y) = \frac{card(ext(X \wedge Y)=\{w \in \Omega / X(w)=vrai, Y(w)=vrai\}}{card(\Omega)}$
$ConfS(X \rightarrow Y) = \frac{card(ext(X \wedge Y)=\{w \in \Omega / X(w)=vrai, Y(w)=vrai\}}{card(ext(X)=\{w \in \Omega / X(w)=vrai\}} = \frac{supS(X \rightarrow Y)}{supS(X)}$

Exemple : A partir de la matrice des concepts table 2, nous pouvons calculer la confiance $ConfS(0 < \mathrm{P}_p \leq 2/3 \rightarrow 1/3 < \mathrm{P}_v \leq 2/3) = \frac{1+1+1+0}{1+1+1+1} = 75\%$

Nous donnons table 3 les règles d'association obtenues à partir de la matrice de données table 2 pour un support minimum de 75% et une confiance minimum de 75%.

Ces indicateurs se révèlent insuffisants pour les règles d'association extraites à partir d'une matrice de concepts décrits par des variables diagrammes. Nous constatons qu'il y a de l'imprécision plus ou moins grande sur la fréquence en conclusion de ces règles. Plus l'intervalle de fréquence est petit en conclusion, plus nous pouvons considérer que la règle d'association est précise et donc intéressante. Nous étendons alors la mesure de la confiance aux cas des règles d'association extraites à partir d'une matrice de concepts décrits par des variables diagrammes. Avec cette confiance "diagramme", notée $confD$, nous pénalisons les règles avec de grands intervalles de fréquences en conclusion:
$ConfD(X \rightarrow Y) = confS(X \rightarrow Y)/(1 + \frac{\sum_{j,v}(\bar{y}_{j,v} - \underline{y}_{j,v})}{n_v})$ où n_v est le nombre de propriétés en conclusion.
Exemple : A partir de la matrice des concepts table 2, nous pouvons calculer la confiance diagramme $ConfD(0 < \mathrm{P}_p \leq 2/3 \rightarrow 1/3 < \mathrm{P}_v \leq 2/3) = 75\%/(1+(2/3\text{-}1/3)) = 56\%$

Propriété : $\frac{1}{2} ConfS(A \rightarrow B) \leq ConfD(A \rightarrow B) < ConfS(A \rightarrow B)$

Preuve : $\forall (j,v)$, $\underline{y}_{j,v} < \bar{y}_{j,v}$ donc $\frac{\sum_{j,v}(\bar{y}_{j,v} - \underline{y}_{j,v})}{n_v} > 0$. D'autre part, au maximum, les longueurs des intervalles en conclusion sont toutes égales à 1 ($\forall (j,v)$, $\bar{y}_{j,v}$-$\underline{y}_{j,v}$=1) et donc $\frac{\sum_{j,v}(\bar{y}_{j,v} - \underline{y}_{j,v})}{n_v} = 1.\diamond$

Nous pouvons alors définir un $confD$ minimum $minconfD$ pour la génération des règles. Nous donnons une expérimentation de cet indicateur dans la section suivante.

	h=1,minsup=10%, minconf=70%	3,8%,70%	6,6%,70%	6,6%,25%
Règles minconf	630	1240	8220	14555
Règles minconfD	0	162	1671	10481

Table 4: Règles d'association obtenues en modifiant h, $minsup$, $minconfS$ ou $minconfD$

4 Application

Nous partons d'une base de données sur des consommateurs américains. Cette base répertorie 93850 enregistrements et distingue 77 catégories différentes de produits (items). Nous construisons les concepts clients, soient 799 concepts. Chaque client est alors décrit par un diagramme résumant sa consommation. Nous appliquons l'algorithme symbolique SApriori aux 799 concepts clients avec quatre triplets (h = précision, $minsupS$ = support minimum, $minconfS$ = confiance minimum ou $minconfD$ = confiance diagramme minimum) différents ((1,10%,70%), (3,8%,70%), (6,6%,70%), (6,6%,25%)).

h est un paramètre propre à l'algorithme Apriori symbolique qui autorise la recherche d'intervalles de fréquences plus ou moins précis dans la définition des règles d'association. Plus h est grand, plus les intervalles de fréquences sont précis. Les règles d'association découvertes sont alors du type: $\bigwedge_{i,u}\{\frac{\underline{x}_{i,u}}{h} < P_{x_{i,u}} \leq \frac{\bar{x}_{i,u}}{h}\} \rightarrow \bigwedge_{j,v}\{\frac{\underline{y}_{j,v}}{h} < P_{y_{j,v}} \leq \frac{\bar{y}_{j,v}}{h}\}$ où $\underline{x}_{i,u}$ = 0..h-1 et $\bar{x}_{i,u}$ = 1..h, $\underline{y}_{j,v}$ = 0..h-1 et $\bar{y}_{j,v}$ = 1..h.

Exemples: Si $h = 1$, nous trouvons des règles du type $0<P_v\leq 1 \rightarrow 0<P_p\leq 1$. Si $h = 6$, nous trouvons des règles du type $\{1/6<Pv\leq 2/6\}\wedge\{1/6<Pc\leq 3/6\}\rightarrow\{0<Pp \leq 1/6\}$.

Pour conclure, nous donnons table 4 le nombre de règles d'association respectant les seuils minimaux de confiance $minconfS$ et $minconfD$. Lorsque nous regardons les règles d'association respectant le seuil minimum de confiance diagramme ($minconfD$), nous remarquons que nous n'en avons aucune pour le test ($h = 1$, $minsup = 10\%$, $minconfD = 70\%$) alors que nous en avions 630 pour une confiance minimale également de 70%. En effet, la précision $h = 1$ correspond au cas où nous ne faisons aucune distinction entre les niveaux de fréquences d'une modalité dans les diagrammes (autre que la distinction entre fréquences nulles et non nulles). Par contre, pour le triplet ($h = 6$, $minsup = 6\%$, $minconfD = 70\%$), nous obtenons 1671 règles d'association. Nous constatons donc que la confiance diagramme pénalise fortement les règles d'association avec une forte imprécision en conclusion et par la même fait ressortir les règles d'association plus précises.

[1] F. Afonso, "Méthodes prédictives par extraction de règles en présence de données symboliques", *Thèse doctorale*, Paris Dauphine, France, 2005.

[2] F. Afonso, "Extension de l'algorithme Apriori et des règles d'association au cas des données symboliques diagrammes et sélection des meilleures règles par la régression linéaire symbolique", *RNTI, Classification et fouilles de données*, D.A. Zighed et G. Venturini eds, Cépadues 2004.

[3] R. Agrawal, T. Imielinski, A. Swami, "Mining association rules between sets of items in large databases", *ACM SIGMOD Records*, 1993.

[4] R. Agrawal, R. Srikant, "Fast algorithms for mining association rules", *Proc. of the 20th Int'l Conf. on Very Large Databases*, 1994.

[5] H-H. Bock, E. Diday, *Analysis of Symbolic Data, Exploratory methods for extracting statistical information from complex data*, Springer Verlag, 2000.

Amélioration du Boosting par combinaison des hypothèses antérieures

E. Bahri[1], N. Nicoloyannis[1] and M. Maddouri[2]

1. ERIC, 5 avenue Pierre Mendes France, 69676 Bron Cedex
2. INSAT, zone urbaine, 1002 la Charguia II Tunisie
(emna.bahri, nicolas.nicoloyannis)@univ-lyon2.fr
Mondher.Maddouri@fst.rnu.tn

Mots clés : Agrégation de classifieurs, apprentissage automatique, Boosting.

La réduction de l'erreur en généralisation est l'une des principales motivations de la recherche en apprentissage automatique. De ce fait, un grand nombre de travaux ont été menés sur les méthodes d'agrégation de classifieurs afin d'améliorer, par des techniques de vote [6] et [7], les performances d'un classifieur unique. Le boosting et son algorithme de base AdaBoost [8] font partie des méthodes ensemblistes les plus performantes aujourd'hui, grâce à la mise à jour adaptative de la distribution des exemples visant à augmenter de façon exponentielle le poids des exemples mal classés. De nombreux résultats théoriques sont venus étayer son comportement déjà très efficace en pratique. Notamment, les théorèmes montrant la décroissance exponentielle des erreurs empiriques et de généralisation ont apporté des réponses formelles au non-sur-apprentissage du boosting quand le nombre d'itérations augmente [3]. Un des points encore fortement étudié actuellement dans la communauté porte sur la tolérance du boosting face à des données bruitées [1], [4] et [2]. En effet, les résultats théoriques originels pré-supposent l'exploitation de données "pures et parfaites". Les bases de données modernes ont remis en cause ce postulat et nécessité le développement de nouveaux algorithmes de boosting. Les travaux présentés se positionnent à ce niveau là. En effet, on propose une amélioration de l'algorithme du boosting pour faire face aux problèmes liés au sur-apprentissage face au bruit et à la vitesse de convergence.

La nouvelle approche d'Adaboost

Soit X_0 à prévoir et $S = (x_1, y_1), ..., (x_n, y_n)$ un échantillon
- Pour i=1,n faire
- Initialiser les poids $p_0(x_i) = 1/n$;
- Fin pour
- $t \leftarrow 0$
- Tant que $t \leq T$ faire
- Tirer un échantillon d'apprentissage S_t dans S selon les probabilités p_t.
- Construire une hypothèse h_t sur S_t par un algorithme d'apprentissage A.
- Soit ϵ_t l'erreur apparente de h_t sur S avec $\epsilon_t = \sum poids \quad des \quad exemples$ ***tel que*** $argmax(\sum_{i=1}^{t} \alpha_i h_i(x_i) \neq y_i)$. Calculer $\alpha_t = 1/2ln((1-\epsilon_t)/\epsilon_t)$.
- Pour i=1, m faire
- $P_{t+1}(x_i) \leftarrow (p_t(x_i)/Z_t)e^{-\alpha_t}$ **si** $argmax(\sum_{i=1}^{t} \alpha_i h_i(x_i)) = y_i$ **(bien classé)**
  $P_{t+1}(xi) \leftarrow (p_t(x_i)/Z_t)e^{+\alpha_t}$ **si** $argmax(\ \sum_{i=1}^{t} \alpha_i h_i(x_i)) \neq y_i$ **(mal classé)**
  (Z_t est une valeur de normalisation telle que $\sum_{i=1}^{n} p_t(x_i) = 1$)
- Fin Pour
  $t \leftarrow t + 1$
- Fin tant que
- Fournir en sortie l'hypothèse finale :
  $H(x) = argmax \ \ y \in Y \sum_{t=1}^{T} \alpha_t$

La modification au sein de l'algorithme est faite :

- Lors de la modification des poids des exemples : En fait, on ne compare pas seulement la classe prédite par l'hypothèse à l'itération courante avec la classe réelle mais la somme des hypothèses pondérées depuis la première itération jusqu'à l'itération courante . Si cette somme vote pour une classe différente de la classe réelle alors une mise à jour exponentielle tel que le cas d'AdaBoost est appliquée à l'exemple mal classé. De ce fait des résultats portant sur l'amélioration de la vitesse de convergence sont attendus, de même pour la réduction de l'erreur de généralisation étant donnée la richesse de l'espace des hypothèses à chaque itération.
- Lors du calcul de l'erreur $\epsilon(t)$ de l'hypothèse à l'itération t : Cette méthode, à chaque itération, prend en considération les hypothèses précédentes à l'itération courante lors du calcul de $\epsilon(t)$. De ce fait, l'erreur apparente à chaque itération est le poids des exemples voté par les hypothèses pondérées des itérations antérieures comme mal classés par comparaison à la classe réelle. Cette modification, laisse l'algorithme à chaque itération très dépendant des autres itérations. Des résultats améliorant surtout l'erreur de généralisation sont attendus puisque le vote de chaque hypothèse (coefficient $\alpha(t)$) est calculé à partir des autres hypothèses.

Expérimentations

Dans cette section, on va essayer de comparer en termes d'erreur en généralisation, de rappel et de vitesse de convergence les deux approches, l'approche d'origine AdaBoost et l'approche Hybride AdaBoost*Hyb*. L'apprenant faible utilisé est l'algorithme C4.5 choisi suite à l'étude [1] qui a montré que C4.5 est trés sensible au bruit. On a travaillé sur 15 bases de l'UCI Irvine [5] tout en indiquant la valeur de l'erreur en généralisation et le rappel, choisis comme critère de performance. Ensuite, on a bruité aléatoirement ces bases de données avec un taux de bruit de 20% selon [1], pour savoir le comportement de différents algorithmes. Enfin, on a établi un diagnostic de convergence de ces différents algorithmes en se basant sur le nombre d'itérations effectuées.

On s'est basé sur le principe de la diversité dans le choix des bases de données (Valeurs manquantes, classe à K modalités, etc.).

Comparaison en terme d'erreur de généralisation et de Rappel

Le tableau Tab.1 présente les résultats obtenus pour cette partie en ayant choisi pour chacun des algorithmes d'effectuer 20 itérations. Le choix du nombre d'itérations sera expliqué dans la dernière partie d'éxperiences. On a indiqué le taux d'erreur en généralisation et le rappel pour chacun des algorithmes AdaBoost M1 et AdaBoost Hyb, puisque notre approche n'améliore pas effectivement le boosting si elle agit négativement sur le rappel. L'observation des résultats montre déjà les effets positifs de l'approche hybride. En effet, pour 14 bases sur 15, l'algorithme AdaBoost*Hyb* présente un taux d'erreur inférieur ou égale à AdaBoost M1. En effet, c'est seulement pour la base Lymph que notre approche donne une erreur de généralisation plus élevée que l'approche classique. Ce gain en faveur de AdaBoost Hyb nous montre bien qu'en exploitant des hypothèses générées aux itérations antérieures pour corriger les poids des exemples, il est possible d'améliorer les performances du Boosting. Ceci peut être expliqué par la précision du calcul de l'erreur apparente $\epsilon(t)$ et par conséquent le calcul du coefficient du classifieur $\alpha(t)$ ainsi que la richesse de l'espace des hypothèses à chaque itération puisqu'il s'agit de l'ensemble des hypothèses générées aux itérations précédentes et de l'itération courante. De même, l'AdaBoost Hyb augmente le rappel des bases de données ayant des taux d'erreur moins importants. Le rappel des deux algorithmes est le même dans le cas où les taux d'erreur des bases de données sont égaux. On constate aussi que notre approche améliore le rappel dans le cas de la base Lymph où l'erreur était plus importante. On note alors que la nouvelle approche n'agit pas négativement sur le rappel mais elle l'améliore même lorsque qu'on a une erreur de généralisation plus importante.

Bases de Données	Adabbost M1		Adaboost hyb	
-	Erreur	Rappel	Erreur	Rappel
Iris	6.00%	93.00%	**3.00%**	**96,00%**
Nhl	35.00%	65.00%	**28.00%**	**71.00%**
Vote	4.36%	95.00%	**4.13%**	**95.00%**
Weather	21.42%	63.00%	**21.00%**	**64.00%**
Credit-A	15.79%	84.00%	**13.91%**	**86.00%**
Titanic	**21.00 %**	**68.00 %**	**21.00%**	**68.00 %**
Diabètes	27.61%	65.00%	**25.56%**	**68.00%**
Hypothyroid	0.53%	72.00%	**0.42%**	**74.00%**
Hépatitis	15.62%	69.00%	**14.00%**	**73.00%**
Contact-lenses	25.21%	67.00%	**16.00%**	**85.00%**
Zoo	**7.00%**	**82.00%**	**7.00%**	**82.00%**
Straight	2.40%	95.00%	**2.00%**	**97.00%**
IDS	1,90%	97.00%	**0.37%**	**98.00%**
Lymph	**19.51%**	54.00%	20.97%	**76.00%**
Breast-Cancer	45.81%	53.00%	**30.41%**	**60.00%**

TAB. 1 – Performances en termes de taux d'erreurs et Rappel

Comparaison sur des données bruitées

Dans cette partie, on a ajouté un taux de bruit de 20% pour chacune de ces bases, en changeant aléatoirement la valeur de la classe prédite à l'aide d'un programme par une autre valeur possible de cette classe. Le tableau Tab.2 nous montre le comportement des algorithmes vis- à-vis du bruit. On remarque bien que l'approche hybride est sensible elle aussi au bruit puisque le taux d'erreurs en généralisation est augmenté pour toute les bases des données. Cependant cette augmentation reste toujours inferieur à celle de l'approche classique sauf pour les bases de données telles que Credit-A, Hépatitis et Hypotyroid. Donc, on a étudié de prés ces bases de données et on a noté un point : les valeurs manquantes. En fait, Crédit-A, Hépatitis et Hypothyroid possèdent respectivement 5%, 6%et 5,4% de valeurs manquantes. On constate alors que notre amélioration perd son effet avec l'accumulation de deux types de bruit : les valeurs manquantes et le bruit artificiel , bien que l'algorithme AdaBoost*Hyb* améliore les performances d'AdaBoost contre le bruit sur le reste des bases de données.

Bases de données	**AdaBoost M1**	**AdaBoostHyb**
Iris	33.00%	**28.00%**
Nhl	45.00%	**32.00%**
Vote	12.58%	**7.76%**
Weather	25.00%	**21%**
Credit-A	**22.56%**	24.00%
Titanic	34.67%	**26.98%**
Diabètes	36.43%	**31.20%**
Hypothyroid	**0.92%**	2.12%
Hépathitis	**31.00%**	41.00%
Contact-lenses	33%	**25%**
Zoo	18.84%	**11.20%**
Straight	3.45%	**2.81%**
IDS	2.40%	**0.50%**
Lymph	28.73%	**24.05%**
Breast-Cancer	68.00%	**48.52%**

TAB. 2 – Performances en termes de taux d'erreurs sur des données bruitées

Comparaison de la vitesse de convergence

Dans cette partie, on va s'intéresser au nombre d'itérations à partir duquel les algorithmes convergent, c'est à dire où le taux d'erreur se stabilise. Le tableau Tab.3 nous montre que l'approche hybride permet à AdaBoost de converger plus vite. En effet, le taux d'erreur d'AdaBoost M1 ne se stabilise pas même à la 100 ième itération, alors que AdaboostHyb converge à la 20 ième itération ou même avant. C'est pour cette raison qu'on a choisi pour la première partie 20 itérations pour effectuer la comparaison en termes d'erreur et de rappel. Ces résultats sont aussi valables pour la base de données Hépatitis. En fait, Cette base de données est riche par les valeurs manquantes (Taux 6%). Ces valeurs manquantes présentent toujours un problème de convergence pour les algorithmes

d'apprentissage. De plus, ces même résultats se manifestent sur des bases de données de différents types (plusieurs attributs, la classe à prédire à K modalités, tailles importantes). Ceci nous laisse penser que dû à la façon de calculer l'erreur apparente tenant compte des hypothèses antérieures, l'algorithme atteint plus rapidement la stabilité.

-	Adabbost M1				Adaboost hyb			
Nb. itérations	10	20	50	100	10	20	50	100
Iris	7,00	6,00	5,90	5,85	3,50	3,00	3,00	3,00
Nhl	37,00	35,00	34,87	34,55	31,00	28,00	28,00	28,00
Weather	21,50	21,42	21,40	14,40	21,03	21,00	21,00	21,00
Credit-A	15,85	15,79	15,75	14,71	14,00	13,91	13,91	13,91
Titanic	21,00	21,00	21,00	21,00	21,00	21,00	21,00	21,00
Diabetes	27,70	27,61	27,55	27,54	25,56	25,56	25,56	25,56
Hypothyroid	0,60	0,51	0,51	0,50	0,43	0,42	0,42	0,42
Hepatitis	16,12	15,60	14,83	14,19	14,03	14,00	14,00	14,00
Contact-Lenses	26,30	24,80	24,50	16,33	16,00	16,00	16,00	16,00
Zoo	7,06	7,00	7,00	7,00	7,00	6,98	7,00	7,00
Straight	2,50	2,46	2,45	2,42	0,42	0,42	0,42	0,42
IDS	2,00	1,90	1,88	1,85	0,37	0,37	0,37	0,37
Lymph	19,53	19,51	19,51	19,50	20,97	20,97	20,97	20,97
Breast-Cancer	45,89	45,81	45,81	45,79	30,50	30,41	30,41	30,41

TAB. 3 – Performances en termes de vitesse de convergence en pourcentage

Conclusion et perspectives

Les éxperimentations et les résultats trouvés montrent que l'approche proposée améliore les performances d'AdaBoost en taux d'erreur, en rappel et en vitesse de convergence. Cependant, il s'est avéré que cette même approche est sensible elle aussi au bruit. Donc, une étude théorique sur la convergence est envisagée pour confirmer les resultats experimentaux. Une perspective qui nous semble importante consiste à améliorer cette approche contre les données bruitées en se basant soit sur les graphes de voisinage soit sur des paramètres de mise à jour efficaces.

Références

[1] T G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees : bagging, boosting, and randomization. Machine Learning (1999) 1-22

[2] T G. Dietterich. Ensemble Methodes in Machine Learning. First International Workshop on Multiple ClassifierSystems (2000) 1-15

[3] R. E. Schapire and Y. Singer. Improved boosting algorithms using confedence rated predictions. Machine Learning (1999), vol (37) number 3 297-336

[4] G. Ratsch. Ensemble learning methods for classification. Master's thesis,Dep of computer science, University of Potsdam April (1998)

[5] D.J. Newman, S. Hettich, C.L. Blake and C.J. Merz. UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998) http ://www.ics.uci.edu/∼mlearn/MLRepository.html

[6] E. Bauer and R. Kohavi. PAn Empirical Comparison of Voting Classification Algorithms : Bagging, Boosting, and Variants. Machine Learning vol 24 (1999) 173-202

[7] N. Littlestone and M. K. Warmuth. The Weighted Majority Algorithm. Information and computation (1994)vol 24 212-261

[8] R. Shapire. The strength of weak learnability. Machine Learning vol 5 (1990) 197-227

La discrimination logistique dans un cas de mélange de deux sous populations

F. BENINEL

CREST-ENSAI, Campus de Ker Lann, 35170 Bruz, France
farid.beninel@ensai.fr

Mots clés : apprentissage sur une sous population et prédiction sur une autre; discrimination logistique généralisée

Habituellement en analyse discriminante on a à prédire le groupe d'appartenance à partir des variables de description ou covariables. La règle de prédiction est élaborée en utilisant un échantillon d'apprentissage soumis aux mêmes conditions externes que les individus à prédire.

Dans ce travail, on s'intéresse à la prédiction d'individus d'une certaine sous population utilisant un échantillon d'apprentissage d'une autre sous population.
Ce problème nous est apparu l'occasion d'une étude biologique de prédiction du sexe d'oiseaux d'une certaine éspèce, à partir de mesures biométriques (Van Franecker, Ter Brack, 1993). Dans le contexte, l'échantillon d'apprentissage est composé d'oiseaux *adultes* et l'échantillon de prédiction est composé d'oiseaux *juvniles*.
L'approche utilisée dans un premier temps, consiste à étendre à ce contexte l'analyse discriminante gaussienne (Biernacki et al., 2002).

Dans un autre contexte (credit scoring dans la banque, prédiction de classes de risque en assurance), il arrive fréquemment qu'échantillon de prédiction et échantillon d'apprentissage diffèrent quant aux conditions externes (BeninelBiernacki, 2007).

Ainsi, on propose d'étendre la discrimination logistique. Différents modèles d'extension sont étudiés. Ces modèles se fondent sur des relations acceptables entre les fonctions scores que l'on associerait à chacune des deux sous populations en présence.

Les données consistent en deux échantillons $(x_i^{(1)}, z_i^{(1)})_{i\in E_L^{(1)}}$ et $(x_i^{(2)}, z_i^{(2)})_{i\in E_L^{(2)}}$.
Les paires $(x_i^{(1)}, z_i^{(1)})$ sont des réalisations indépendantes du couple aléatoire (X, Z) restreint à la sous population Ω_1, les paires $(x_i^{(2)}, z_i^{(2)})$ sont, elles, des réalisations indépendantes du couple (X, Z) restreint a une autre sous population Ω_2.
Les individus dans chacune des deux sous populations sont observés quant au couple (X, Z) toutes choses égales par ailleurs; et d'une sous population à l'autre ces conditions diffèrent.

Le problème,ici, est de mettre en évidence une fonction d'affectation au groupes pour les individus de Ω_2. Le contexte est que la taille de l'échantillon d'apprentissage $E_L^{(2)}$ est faible pour suffire à construire une fonction score.
Toutefois, la taille de $E_L^{(1)}$ est, elle, suffisante pour construire une fonction qui score les individus de Ω_1. L'utilisation d'un lien acceptable entre les fonctions scores des 2 sous populations permet de se servir de l'information que recèlent les échantillons $E_L^{(1)}$ et $E_L^{(2)}$

pour affecter, aux groupes, des individus de Ω_2.

Fonction score de Ω_1: $ln(\frac{\pi(x^{(1)})}{1-\pi(x^{(1)})}) = \beta_0^{(1)} + \beta^{(1)\mathrm{T}} x^{(1)}$

Fonction score de Ω_2: $ln(\frac{\pi(x^{(2)})}{1-\pi(x^{(2)})}) = \beta_0^{(2)} + \beta^{(2)\mathrm{T}} x^{(2)}$

Les modèles de liaisons sont données par $c \in R$ et Λ une matrice diagonale d'ordre d tels que $\beta_0^{(2)} = \beta_0^{(1)} + c$, $\beta^{(2)} = \Lambda\beta^{(1)}$.
Les modèles de liaisons retenus sont ceux faisant intervenir les mêmes variables dans chacun des 2 modèles.

- $(\mathbf{M_1})$: $\beta_0^{(2)} = \beta_0^{(1)}$ et $\beta^{(2)} = \beta^{(1)}$; on reconnaît la discrimination logistique habituelle et il n'y a aucun nouveau paramètre à estimer;
- $(\mathbf{M_2})$: $\beta_0^{(2)} = \beta_0^{(1)}$ et $\beta^{(2)} = \lambda\beta^{(1)}$; seul le paramètre λ est à estimer;
- $(\mathbf{M_3})$: $\beta_0^{(2)}$ libre et $\beta^{(2)} = \beta^1$; seul le paramètre $\beta_0^{(2)}$ est à estimer;
- $(\mathbf{M_4})$: $\beta_0^{(2)}$ libre et $\beta^{(2)} = \lambda\beta^{(1)}$; deux paramètres, $\beta_0^{(2)}$ et λ, sont à estimer;
- $(\mathbf{M_5})$: $\beta_0^{(2)} = \beta_0^{(1)}$ et $\beta^{(2)} = \Lambda\beta^{(1)}$ avec $\Lambda = diag(\lambda_1, .., \lambda_d)$; seuls les paramètres $\lambda_1, .., \lambda_d$ sont à estimer;
- $(\mathbf{M_6})$: $\beta_0^{(2)}$ libre et $\beta^{(2)} = \Lambda\beta^{(1)}$ avec $\Lambda = diag(\lambda_1, .., \lambda_d)$; dans ce cas $(d+1)$ paramètres sont à estimer;

Pour les modèles à suivre, l'estimation des paramètres, de liaison entre les fonctions scores de Ω_1 et celle de Ω_2 se fait par maximisation de la vraisemblance conditionnelle. En notant θ ce paramètre (uni ou multidimensionnel), la log-vraisemblance conditionnelle est donnée par

$$l_{cond}(\theta) = \sum_{i \in E_L^{(2)}} z_i^{(2)} ln(\pi(x_i^{(2)}, \theta)) + (1 - z_i^{(2)}) ln(1 - \pi(x_i^{(2)}, \theta))).$$

On donne, pour l'ensemble des modèles, le système d'èquations (non linéaires).
Concernant l'unicité, pour chaque modèle une condition nécessaire et suffisante est mise en évidence.

(Le modèle M.2): Le paramètre λ est solution de l'équation non linéaire à l'inconnue λ

$$\frac{\partial l_{cond}(\lambda)}{\partial \lambda} = \sum_{i \in E_L^{(2)}} < X_i^{(2)}, \beta^{(1)} > (Z_i^{(2)} - \pi(X_i^{(2)}, \lambda)) = 0.$$

(Le modèle M.3): Le paramètre $\beta_0^{(2)}$ est solution de l'équation non linéaire à l'inconnue $\beta_0^{(2)}$

$$\frac{\partial l_{cond}(\beta_0^{(2)})}{\partial \beta_0^{(2)}} = \sum_{i \in E_L^{(2)}} (Z_i^{(2)} - \pi(X_i^{(2)}, \beta_0^{(2)})) = 0.$$

(Le modèle M.4): La détermination des paramètres $(\beta_0^{(2)}, \lambda)$ revient à résoudre le système non linéaire ci après

$$(S.4) : \begin{cases} \frac{\partial l_{cond}(\beta_0^{(2)},\lambda)}{\partial \beta_0^{(2)}} = \sum_{i \in E_L^{(2)}} (Z_i^{(2)} - \pi(X_i^{(2)}, \beta_0^{(2)}, \lambda)) = 0, \\ \\ \frac{l_{cond}(\beta_0^{(2)},\lambda)}{\partial \lambda} = \sum_{i \in E_L^{(2)}} < X_i^{(2)}, \beta^{(1)} > (Z_i^{(2)} - \pi(X_i^{(2)}, \beta_0^{(2)}, \lambda)) = 0. \end{cases}$$

(Le modèle M.5): Les termes $\lambda_1, .., \lambda_d$ sont solutions du système

$$(S.5) : \begin{cases} \frac{l_{cond}(\Lambda)}{\partial \lambda_k} = \sum_{i \in E_L^{(2)}} X_{i,k}^{(2)} (Z_i^{(2)} - \pi(X_i^{(2)}, \Lambda)) = 0, \\ \\ k = 1, .., d. \end{cases}$$

(Le modèle M.6): Les termes $\beta_0^{(2)}$ et $\lambda_1, .., \lambda_d$ sont solutions du système

$$(S.6) : \begin{cases} \frac{l_{cond}(\beta_0^{(2)},\Lambda)}{\partial \beta_0^{(2)}} = \sum_{i \in E_L^{(2)}} (Z_i^{(2)} - \pi(X_i^{(2)}, \beta_0^{(2)}, \Lambda)) = 0, \\ \\ \frac{l_{cond}(\beta_0^{(2)},\Lambda)}{\partial \lambda_k} = \sum_{i \in E_L^{(2)}} X_{i,k}^{(2)} (Z_i^{(2)} - \pi(X_i^{(2)}, \beta_0^{(2)}, \Lambda)) = 0, \\ \\ k = 1, .., d. \end{cases}$$

Pour l'ensemble des modèles l'unicité de solution est garantie sauf pour des cas singuliers de données, i.e., des données où le nombre d'observations est plus faible que la dimensionnalité du paramètre à estimer.

[1] J. Anderson, "Logistic discrimination", *P. Krishnaiah et L. Kanal(Eds), Handbook of statistics, North Holland* 2, 1982, 169-191.

[2] Beninel, F., Biernacki, C., " Relaxations de la régression logistique: modèles pour l'apprentissage sur une sous population et la prédiction sur une autre, *EGC2007, DMBAF* p:17-26, 2007

[3] C. Biernacki, F. Beninel, V. Bretagnolle, "A generalized discriminant rule when training population and test population differ on their descriptive parameters", *Biometrics* 58, 2002, 387-397.

[4] Mc Lachlan, G., "Discriminant Analysis and Statistical Pattern Recognition", *New York: Wiley*, 1992

[5] Tuffery, S., "Data mining et Statistique décisionnelle", *Paris: Editions Technip*, 2007

Application de méthodes de classification sur des vitesses métrologiques de dégradation de compteurs d'eau

F. Bertrand[1] et M. Maumy[1]

1. IRMA-ULP, 7, rue René Descartes, 67084 Strasbourg cedex
(fbertran,mmaumy)@math.u-strasbg.fr

Mots clés : Applications, discrimination.

Introduction

Les compteurs d'eau, en vieillissant, fournissent une mesure de plus en plus imprécise de la consommation d'eau. Cette dégradation se traduit généralement par un sous-comptage. Ce phénomène est source de problèmes pour les distributeurs d'eau qui ont mis en place des stratégies de gestion des parcs-compteurs ayant comme objectif la réduction des pertes économiques et le respect d'une politique de comptage équitable pour les différents usagers. Toute stratégie nécessite préalablement la compréhension du mécanisme de dégradation et la quantification du sous-comptage. Le vieillissement des compteurs est décrit à travers un modèle dynamique à états discrets, représentant chacun une certaine qualité métrologique. Ce modèle, couplé avec l'observation des erreurs de mesure à l'intérieur de chaque état, permet l'estimation notamment du taux de compteurs défaillants et de l'évolution de la précision de la mesure en fonction de la durée de service du dispositif. L'estimation des paramètres du modèle et la prédiction des valeurs des grandeurs d'intérêt pratique, ont été réalisées dans un cadre bayésien, avec l'utilisation de techniques de simulation MCMC.

Le but de cet article est de classifier en un certain nombre de groupes les compteurs d'eau d'un parc-compteurs d'un distributeur d'eau dans le but d'anticiper sur le changement de ces derniers et par conséquent d'atteindre les deux objectifs préalablement cités. Le papier s'articule en trois parties. La première partie présente le problème : introduction du vocabulaire, liens entre compteurs et qualité de mesure et présentation des données. La deuxième partie concerne la modélisation stochastique du vieillissement des compteurs. Nous introduisons une première ébauche de modèle de dégradation qu'il est nécessaire de compléter en introduisant d'autres facteurs explicatifs. Enfin, la troisième partie développe des techniques de classification pour proposer différents groupes de compteurs d'eau. L'originalité de cet article repose sur l'application de méthodes de classification classiques sur des données issues de modèles bayésiens.

1 Description du problème

1.1 Définitions et vocabulaire

Comme tous les instruments de mesure, les compteurs d'eau sont susceptibles d'erreur. Sur un branchement, le volume enregistré (v_{enr}), et donc facturé, est généralement

différent du volume effectivement consommé ($v_{\text{réel}}$).
L'étude statistique de ce papier, se limitera exclusivement aux données des compteurs dits volumétriques (largement majoritaires en France) caractérisés par la présence d'un organe de mesure qui se déplace sous l'effet de la poussée hydrodynamique, refoulant un volume déterminé d'eau à chaque tour.
L'erreur relative de mesure d'un compteur (e) est obtenue en divisant la différence entre le volume enregistré (v_{enr}) et le volume réellement écoulé ($v_{\text{réel}}$) par ce dernier.
La courbe métrologique est la représentation graphique de la relation entre l'erreur relative de mesure (e) et le débit circulant (q).
Le rendement d'un compteur (r) est le rapport entre le volume enregistré (v_{enr}) et le volume consommé ($v_{\text{réel}}$).

1.2 Classement des compteurs selon leur qualité de mesure

Le modèle de vieillissement prévoit le passage par quatre états métrologiques, de qualité décroissante $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, $\mathcal{E}_4$. La définition des quatre états est inspirée par la réglementation actuelle et ses développements [1]. Nous réservons l'état $\mathcal{E}_1$ aux compteurs qui respectent la conformité aux erreurs maximales tolérées sur toute la gamme de débits. L'état $\mathcal{E}_2$ est caractéristique des compteurs ayant une métrologie imparfaite mais encore acceptable. Les compteurs qui se trouvent dans l'état $\mathcal{E}_3$ présentent des rendements très mauvais et sont non-conformes. Enfin dans l'état $\mathcal{E}_4$ nous avons mis les compteurs bloqués à tout débit, c'est-à-dire ceux qui mesurent une valeur fixe de débit ne variant plus avec le débit réel.

1.3 Données

Deux sources différentes de données sont disponibles pour l'étude. D'une part nous utilisons les résultats d'étalonnage de compteurs en service, réalisés dans les laboratoires d'essai d'un distributeur d'eau (base métrologique) et d'autre part nous nous servons des informations issues d'un outil interne de gestion des parcs de compteurs (base des blocages) pour analyser les phénomènes de blocages. Le recours aux deux bases de données est nécessaire puisque l'état absorbant $\mathcal{E}_4$ est pratiquement inobservé dans la base métrologique.
À partir de la courbe métrologique de chaque compteur, nous pouvons déterminer son état et son rendement. Les individus sont en suite regroupés et dénombrés par état et par âge pour obtenir les $n_i(t)$, soit le nombre de compteurs qui se trouvent dans l'état $\mathcal{E}_i$ à l'âge t. Leurs rendements sont utilisés pour estimer les paramètres des lois de probabilité des rendements à l'intérieur de chaque état.

2 Modélisation stochastique

2.1 Premier modèle de dégradation

Un mécanisme markovien homogène de dégradation a été imaginé sur la base de considérations techniques. En effet des tests d'endurance réalisés par des fabricants sur banc d'essai montrent que le « vieillissement naturel » des compteurs est très lent et ne provoque de changement d'état, à partir de l'état $\mathcal{E}_1$, qu'après des périodes extrêmement longues. Les changements d'état sont donc essentiellement liés à des accidents de parcours. En outre, il est réaliste de faire l'hypothèse que la dégradation est irréversible et donc la

chaîne a pour état absorbant $\mathcal{E}_4$. Le modèle est à temps discret, et l'unité temporelle est une année. Les éléments Θ_{ij} de la matrice de transition Θ sont les probabilités de passage de l'état $\mathcal{E}_i$ (âge $t-1$) à l'état $\mathcal{E}_j$ (âge t).
Le vecteur ligne $\mathbb{P}(t) = \{P_1(t), P_2(t), P_3(t), P_4(t)\}$ des probabilités des quatre états, à l'âge t, est exprimé en fonction de $\mathbb{P}(t-1)$ et de la matrice de transition Θ par : $\mathbb{P}(t) = \mathbb{P}(t-1)\Theta$.
À l'intérieur de chaque état, le rendement suit une loi de probabilité caractéristique de l'état, dont les paramètres sont éventuellement fonction de l'âge. Le couplage entre le modèle de dégradation et les lois de probabilité des rendements permet de modéliser l'évolution de la métrologie d'un ensemble de compteurs par mélange (en proportions variables avec l'âge) d'individus appartenant aux différents états. Pour de plus amples renseignements sur le modèle de dégradation, nous renvoyons à la thèse de Alberto Pasanisi [4].

2.2 Amélioration du modèle de dégradation : à la recherche de nouveaux facteurs explicatifs

Le modèle exposé précédemment peut-être amélioré en introduisant des facteurs explicatifs pertinents. En effet, puisqu'un distributeur d'eau souhaite produire de l'information aussi précise que possible, il est souhaitable d'avoir alors recours à un découpage de son réseau suivant plusieurs critères comme les directions régionales, les centres opérationnels, les agences, etc. Mais un découpage plus fin et particulièrement intéressant en pratique peut être réalisé avec les contrats. Les contrats sont des unités géographiques forcément inégales en extension et nombre d'usagers. Cette répartition du territoire a une utilité opérationnelle. Le paramètre le plus significatif, dans le cadre de cette étude, pour décrire la taille d'un contrat est le nombre de compteurs, qui équivaut en pratique au nombre d'abonnés desservis.
Dans la pratique, il est bien reconnu par les experts et les exploitants que la dégradation des compteurs n'est pas la même sur tous les sites. La principale variable explicative typiquement évoquée comme responsable de cette différence de comportement est la « typologie » d'eau distribuée.
Pour résumer, d'une part la méconnaissance de l'effet de chaque facteur sur le mécanisme de dégradation des compteurs et d'autre part la complexité des conditions réelles d'exploitation, difficilement traduisibles en un nombre réduit de paramètres, poussent à aborder le problème différemment. En effet, l'approche choisie consiste à associer à chaque site d'exploitation une notion d'agressivité qui traduit l'effet global de tous les facteurs.
Les avis a priori des experts dans le domaine du vieillissement des compteurs volumétriques convergent sur un nombre de groupes d'agressivité égal à trois. Les méthodes développées dans [4], se déroulant en trois phases, permettent de constituer ces trois groupes de contrats caractérisés par différents intervalles d'agressivité délimités par trois valeurs particulières d'agressivité (A_1, A_2, A_3 en ordre croissant) :

1. Estimation d'un paramètre d'agressivité uniquement sur la base des résultats métrologiques. Ce paramètre est appelé la **vitesse de dégradation métrologique** λ. En s'appuyant sur la valeur de la vitesse λ, les contrats sont découpés en trois groupes d'agressivité croissante.

2. Pour chacun des trois groupes ainsi obtenus, nous calculons les taux de blocage des compteurs de référence, suivant les informations de la base des blocages. Les calculs dans [4] montrent une cohérence entre les deux indicateurs d'agressivité (vitesse de dégradation métrologique et taux de blocage) : les sites où les compteurs

se dégradent plus vite sont aussi ceux où nous observons, en proportion, plus de blocages.

3. Estimation de l'agressivité des contrats pour lesquels uniquement les informations de la base des blocages sont disponibles, grâce à l'utilisation de l'association entre le taux de blocage et l'agressivité. En pratique nous assignons un contrat à un des trois groupes précédemment découpés en fonction de la valeur observée des taux de blocage des compteurs de référence. Si, par exemple, les taux observés sur un site d'agressivité inconnue sont plus proches de ceux qui caractérisent le groupe A_1, alors nous assignons le contrat examiné au groupe A_1 etc.

3 Classification des contrats

L'objectif principal de cette étude est de proposer une classification des contrats à partir du calcul de l'estimation de la vitesse de dégradation métrologique λ.
Un premier problème se pose : savoir si nous devons faire des groupes et le cas échéant, déterminer le nombre de ces groupes. Il est à noter que, jusqu'à présent, à la fois la détermination du nombre de groupes et leur constitution, était réalisée grâce à la connaissance et aux avis des experts du vieillisement des compteurs d'eau. En utilisant ce nombre de groupes fixé a priori, nous avons utilisé des méthodes classiques de classification comme la classification hiérarchique ascendante, la méthode des $k-$moyennes et celle des cartes de Kohonen. Puis, nous avons comparé les résultats obtenus avec les groupes constitués par les experts, puisqu'une comparaison directe, c'est-à-dire sans référence extérieure, des résultats de ces trois méthodes n'a pas de sens.
La méthode, la plus compatible avec les avis des experts en la matière et dont nous nous sommes donc finalement servis, est la classification hiérarchique ascendante pour diverses distances associées à la méthode de liaison de Ward. Ce constat nous a naturellement amené à nous demander en combien de groupes il était le plus adéquat de couper le dendrogramme issu des classifications hiérarchiques ascendantes obtenues précédemment. Nous avons déterminé statistiquement le nombre de groupes d'agressivité à considérer en utilisant les règles de Mojena [3] particulièrement adaptées aux méthodes hiérarchiques, voir le livre de Everitt [2] pour de plus amples détails. Il s'est dégagé un nombre optimal de groupes égal à quatre et la répartition des contrats parmi ces derniers a été validée par les experts du domaine. L'exposé présentera cette démarche de manière détaillée et illustrée par des données réelles.

[1] A. Costes et Y. Pia, "Les compteurs d'eau en France : la règlementation et son évolution", *Techniques, Sciences et Méthodes* 7, 2000, 21–27.
[2] B. S. Everitt, *Cluster analysis*, 4^{th} edition, Arnold Publication, 2001.
[3] R. Mojena, "Hierarchical grouping methods and stopping rules: An evaluation", *Computer Journal* 20 (4), 1977, 359–363.
[4] A. Pasanisi, "Aide à la décision dans la gestion des parcs de compteurs d'eau potable", *Thèse présentée pour obtenir le grade de Docteur de l'ENGREF en Sciences de l'Eau (Option Statistique)*, 2004.

Classification basée sur l'agrégation d'opinions par la méthode de recuit simulé

M. Boubou[1] , A. Bounekkar[1] et M. Lamure[1]

Université Lyon I, LIRIS-MA2D
43 Boulevard du 11 Novembre 1918
69622 VILLEURBANNE CEDEX, France
(boubou,bounekkar,lamure)@univ-lyon1.fr

Résumé

Dans ce papier, nous allons présenter une méthode de classification basée sur l'agrégation d'opinions. La méthode proposée consiste à associer à chaque variable une fonction de classement qui va jouer le rôle de juge. Ce dernier va classer les individus selon ses propres critères. A partir de l'ensemble de classement de toutes les variables, on cherche à construire sur l'ensemble des individus un classement collectif qui soit la meilleure agrégation possible.
Afin de résoudre le problème d'optimisation rencontré, nous avons utilisé la méthode de recuit simulé.

Mots clés : classification, optimisation, recuit simulé, agrégation, partition.

1 Introduction

Dans ce papier, une méthode de classification basée sur l'agrégation d'opinions. Dans ce contexte, nous proposons une solution du problème d'agrégation des relations binaires en relation d'équivalence par l'utilisation de la méthode du recuit simulé.
Ce problème a été posé par *Régnier* [5] repris ensuite par *Marcotorchino et al.* [4], *Amorim et al.* [2] et *Barthélemy et al.* [1].

2 Présentation du problème

Nous considérons deux ensembles : L'ensemble des variables, $\mathcal{V} = \{V_1, V_2, \ldots, V_p\}$ et l'ensemble des individus, $\mathcal{I} = \{w_1, w_2, \ldots, w_n\}$. Nous associons à une variable V_k une application $A_k(.,.) : \mathcal{I}^2 \rightarrow \{0,1\}$ tel que $\forall (w_i, w_j) \in \mathcal{I}^2; w_i \neq w_j$

- $A_k(w_i, w_j) = 1$ Si pour V_k, w_i et w_j sont dans une même classe.
- $A_k(w_i, w_j) = 0$ Si pour V_k, w_i et w_j sont dans deux classes différentes.

Étant donné les classements des p variables $A_1, A_2, \ldots, A_p$ nous cherchons à construire sur $\mathcal{I}$, un classement collectif qui soit la meilleure agrégation possible qui va maximiser le nombre des concordances entre le classement collectif et les ensembles de classement des variables.
Une partition de l'ensemble $\mathcal{I}$ peut être assimilée à un vote défini sur $\mathcal{I}$ qui peut prendre autant des modalités qu'il y a d'éléments dans la partition.
Nous identifierons cette partition à une application $X(.,.)$ de $\mathcal{I}^2$ dans $\{0,1\}$.
Étant données, deux applications $X(.,.)$ et $Y(.,.)$ de $\mathcal{I}^2$ dans $\{0,1\}$, on appelle concordance de $X(.,.)$ et de $Y(.,.)$ notée $C(X,Y)$ le nombre entier défini par :

$$C(X,Y) = Card\{(w_i, w_j)\,;\, X(w_i, w_j) = Y(w_i, w_j)\}$$

Cette concordance peut se décomposer en deux catégories de concordances :

– Une concordance positive : $C^+(X,Y) = Card\{(w_i,w_j)\,;\, X(w_i,w_j) = Y(w_i,w_j) = 1\}$
– Une concordance négative : $C^-(X,Y) = Card\{(w_i,w_j)\,;\, X(w_i,w_j) = Y(w_i,w_j) = 0\}$

Étant donnée une partition caractérisée par l'application $X(.,.)$ de $\mathcal{I}^2$ dans $\{0,1\}$ et le classement d'une variable V_k, la concordance entre cette partition et le classement de la variable V_k est donnée par $C(X,A_k)$. Les différents classements de variables $A_k; k = 1,\ldots,p$ étant données, nous cherchons X qui maximise l'expression :

$$C(X,\mathcal{A}) = \sum_{k=1}^{p} C(X,A_k) \tag{1}$$

Ce problème de maximisation peut être formulé en terme de programmation linéaire en nombre entiers. En effet, pour tout $(w_i,w_j) \in \mathcal{I}^2$ on pose $x_{ij} = X(w_i,w_j)$ et pour tout $k = 1,\ldots,p$ on pose $a_{ij}^k = A_k(w_i,w_j)$

$$C(X,A_k) = Card\{(w_i,w_j)\,;\, X(w_i,w_j) = A_k(w_i,w_j)\} = Card\{(w_i,w_j)\,;\, x_{ij} = a_{ij}^k\}$$

Le couple (w_i,w_j) contribue donc à la cohérence dans tous les cas suivants :
– Cas 1 : $x_{ij} = a_{ij}^k = 1$ qui est équivalent à : $x_{ij}a_{ij}^k = 1$
– Cas 2 : $x_{ij} = a_{ij}^k = 0$ qui est équivalent à : $(1-x_{ij})(1-a_{ij}^k) = 1$

Par suite :

$$C(X,\mathcal{A}) = \frac{1}{n^2}\sum_{k=1}^{p}\sum_{(i,j)\in\mathcal{I}^2}(1-x_{ij})(1-a_{ij}^k) + x_{ij}a_{ij}^k$$

Etude du critère d'optimisation

Un algorithme de résolution de ce problème, connu sous le nom de "Clique partitioning", a été proposé dans [3, 2] en utilisant la programmation linéaire. Pour résoudre ce problème, nous proposons une approche heuristique : l'algorithme du recuit simulé.

Nous associons à une application X définie sur $\mathcal{I}^2$ deux vecteurs (x_{ij}) et $(\bar{x}_{ij})$ définis comme suit : $\forall (w_i,w_j) \in \mathcal{I}^2$:

$$\begin{cases} x_{ij} = 1 \text{ Si } X(w_i,w_j) = 1 \\ x_{ij} = 0 \text{ Sinon} \end{cases} \quad \text{et} \quad \begin{cases} \bar{x}_{ij} = 1 \text{ Si } X(w_i,w_j) = 0 \\ \bar{x}_{ij} = 0 \text{ Sinon} \end{cases}$$

Nous associons à la variable $V_k; (k = 1,\ldots,p)$ les deux vecteurs (a_{ij}^k) et $(\bar{a}_{ij}^k)$ définis comme suit $\forall (w_i,w_j) \in \mathcal{I}^2$:

$$\begin{cases} a_{ij}^k = 1 \text{ Si } A_k(w_i,w_j) = 1 \\ a_{ij}^k = 0 \text{ Sinon} \end{cases} \quad \text{et} \quad \begin{cases} \bar{a}_{ij}^k = 1 \text{ Si } A_k(w_i,w_j) = 0 \\ \bar{a}_{ij}^k = 0 \text{ Sinon} \end{cases}$$

Il y a une concordance entre la partition donnée par X et la partition donnée par A_k si : $(x_{ij} = 1 \text{ et } a_{ij}^k = 1)$ ou si $(\bar{x}_{ij} = 1 \text{ et } \bar{a}_{ij}^k = 1)$. Notons :

$$\begin{cases} q_{ij}^k = 1 \text{ s'il y a une concordance entre } X(w_i,w_j) \text{ et } A_k(w_i,w_j) \\ q_{ij}^k = 0 \text{ Sinon} \end{cases}$$

Le nombre de concordances $C(X,A_k) = \sum_{(i,j)\in\mathcal{I}^2} q_{ij}^k$ et le nombre de concordance entre X et les partitions données par $A_k; (k = 1,\ldots,p)$ peut s'écrire :

$$C(X,\mathcal{A}) = \sum_{k=1}^{p} C(X,A_k) = \sum_{k=1}^{p}\sum_{(i,j)\in\mathcal{I}^2}[x_{ij}a_{ij}^k + \bar{x}_{ij}\bar{a}_{ij}^k] = \sum_{k=1}^{p}\sum_{(i,j)\in\mathcal{I}^2}[x_{ij}a_{ij}^k + (1-x_{ij})\bar{a}_{ij}^k] = \sum_{k=1}^{p}\sum_{(i,j)\in\mathcal{I}^2}\bar{a}_{ij}^k + \sum_{k=1}^{p}\sum_{(i,j)\in\mathcal{I}^2}[a_{ij}^k - \bar{a}_{ij}^k]x_{ij}$$

$\forall (w_i,w_j) \in \mathcal{I}^2$ nous posons ; $(r_{ij} = \sum_{k=1}^{p} a_{ij}^k)$, $(\bar{r}_{ij} = \sum_{k=1}^{p} \bar{a}_{ij}^k)$ et $(s_{ij} = r_{ij} - \bar{r}_{ij})$. L'expression (1) devient alors

$$C(X,\mathcal{A}) = \sum_{(i,j)\in\mathcal{I}^2}\bar{r}_{ij} + \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij} \tag{2}$$

3 Formulation du problème

La solution du problème revient à maximiser l'expression $C(X, \mathcal{A})$ donné par (2). Ceci revient à maximiser la quantité linéaire : $\sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij}$. Pour poser la propriété de transitivité imposée à X par des contraintes linéaires sur les x_{ij}, notons : $T = \{(i,j,k) \in I^3 | 1 \leq i < j < k \leq n\}$. Donc la propriété de transitivité est équivalente aux trois conditions linéaires suivantes :

$$\forall (i,j,k) \in T \left\{ \begin{array}{c} x_{ij} + x_{jk} - x_{ik} \leq 1 \\ x_{ij} - x_{jk} + x_{ik} \leq 1 \\ -x_{ij} + x_{jk} + x_{ik} \leq 1 \end{array} \right. \quad (3)$$

Le problème devient un problème de programmation linéaire en nombres entiers suivant : $max \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij}$, sachant que $x_{ij} \in \{0,1\}$ avec les contraintes linéaires citées dans le formule (3)

Remarque :
Le nombre de variables est de l'ordre de n^2, et le nombre de contraintes est de l'ordre de n^3. C'est ce qu'on appelle "click partitioning problem". Il est un NP-difficile [1],[2]. Plusieurs algorithmes ont été proposés pour la résolution de ce problèmes [3],[2]. Dans ce sens, nous proposons une autre méthode de résolution basée sur l'algorithm du recuit simulé.

4 Résolution du problème

Nous avons trouvé que la donnée d'une partition X vérifiant la propriété de la transitivité est équivalente à celle d'un vecteur $(x_{ij})_{(i,j)\in\mathcal{I}^2}$, chaque élément x_{ij} étant à valeur dans $\{0,1\}$ et vérifiant le condition (3).
Remarquons que la donnée d'un tel vecteur x_{ij} est équivalente à celle d'une partition $\mathcal{P}$ sur $\mathcal{I}$, selon le schéma suivant : $x_{ij} = 1 \Leftrightarrow \exists G \in \mathcal{P}$ tel que $i \in G$ et $j \in G$. Conformément à (2), nous notons : $val(\mathcal{P}) = \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij}$
Si $\mathcal{P} = \{G_1, \ldots, G_m\}$, nous pouvons écrire :

$$val(\mathcal{P}) = \sum_{h=1}^{m} \sum_{\substack{(i,j)\in G_h \\ i<j}} s_{ij}$$

Notre problème prend alors la formulation suivante :

$$max\{val(\mathcal{P}) | \mathcal{P} \in \mathbb{P}\} \quad (4)$$

où $\mathbb{P}$ désigne l'ensemble des partitions de I.
Avant de résoudre ce problème (4), nous allons préciser la description d'un voisinage $\mathcal{V}(\mathcal{P})$ d'une partition $\mathcal{P} = \{G_1, \ldots, G_m\}$.

Définition 1 *Une partition $\mathcal{P}'$ appartient à $\mathcal{V}(\mathcal{P})$, si elle dérive de $\mathcal{P}$ en déplaçant un seul individu i_0 comme suit :*

- *Soit i_0 est transféré dans un groupe $G_l (l \neq h)$, de $\mathcal{V}(\mathcal{P})$,*
- *Soit i_0 constitue un nouveau groupe G_l à lui tout seul.*

Il s'en suit que : $\delta = val(\mathcal{P}') - val(\mathcal{P})$ et, en posant, pour simplifier l'écriture :

$$S_{ij} = \left\{ \begin{array}{rl} s_{ij} & Si \;\; i < j \\ s_{ji} & Si \;\; i > j \\ 0 & Si \;\; i = j \end{array} \right. \Longrightarrow \delta = \sum_{j\in G_l} S_{i_0 j} - \sum_{j\in G_h - \{i_0\}} S_{i_0 j}$$

Le schéma de la procédure est décrit par l'algorithme suivant :

```
Données : T_0, K_Max, r_0 , Résultat : P*
Begin
    Initialisation T := T_0
        choisir au hasard une partition P de I ; P* = P
    Répéter
        change :=FAUX
        Pour k := 1 à K_Max faire
            choisir au hasard P' dans V(P)
            calculer δ = val(P') − val(P)
            tirer au hasard une valeur R dans [0,1] (loi uniforme)
            si δ > 0 ou R < e^(δ/T) alors
                    P := P' ; change :=vrai ;
                si val(P) > val(P*) alors P* := P ;
                fin si
            fin Si
        Fin Pour
    T := T * r_o
    jusqu'à Not change
end
```

Remarques :

1. Remarquons que la définition de $\mathcal{V}(\mathcal{P})$ permet d'atteindre toutes les partitions possibles à partir d'une quelconque partition initiale en utilisant le procédé itératif.
2. Les paramètres T_0, K_{Max}, r_o sont respectivement la température initiale, le nombre d'itérations dans la boucle interne, et le coefficient de refroidissement. Nous avons choisi les valeurs de ces paramètres conformément aux recommandations de [3],[2], et plus précisément $K_{Max} = 10 \times card(I), r_o = 0.98$.
3. La valeur de T_0 doit garantir que $e^{(\delta/T_0)}$ soit proche de 1 en moyenne avec $\delta < 0$.

5 conclusion

L'algorithme du recuit simulé comporte un aspect aléatoire, ce qui explique que deux exécutions successives sur le même exemple ne donnent pas forcément le même résultat. Pour cela nous appellerons "Résultat" de l'algorithme, le meilleur des résultats obtenus lors de plusieurs exécutions successives.
La principale avantage de cette méthode de classification, est que les données ne sont pas forcement quantitatives.
Elle est basée sur l'opinion de l'expert du domaine, qui définit les fonctions de classement.

Références

[1] J.-P. Barthélemy et B. Leclerc, "The median procedure for partitions", *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **19**, (1995) 3–33.

[2] S.G. de Amorim, J.-P. Barthélemy et C.C. Ribeiro, "Clustering and clique partitioning : simulated annealing and tabu search approaches", *Journal of Classification*, **9**,(1992) 17–41

[3] M. Grötschel, Y. Wakabayashi, "A cutting plane algorithm for a clustering problem", *Mathematical Programming*, Series B, **45** (1989) 59–96

[4] J.F. Marcotorchino, P. Michaud, "Agrégation de similarités en classification automatique", *Revue de Statistique Appliquée*, Tom **30** no. 2 (1982)

[5] S. Regnier, "Sur quelques Aspects Mathématiques des Problèmes de Classification Automatique", *I.C.C. Bulletin*, No **4**, 175, (1965).

Classification avec des Multigraphes Application à la classification de données décrites par des variables de différents types

H. Brás Silva[1], P. Brito[2], J. Pinto da Costa[3]

1. Département de Mathématiques, Institut de Génie de Porto (ISEP), Portugal
2. LIAAD/Inesc Porto LA & Fac. Économie, Université de Porto, Portugal
3. Dep. Mathématiques Appliquées / Fac. Sciences, Université de Porto, Portugal
hbs@isep.ipp.pt, mpbrito@fep.up.pt, jpcosta@fc.up.pt

Mots clés : Classification, Partition, Partition consensus, Théorie des Graphes.

Résumé

On présente une extension d'une méthode de classification basée sur la coloration de graphes ([1]), où les variables peuvent être prises en compte individuellement ou en groupes, permettant ainsi d'obtenir des partitions dans des sous-espaces de variables. En particulier, cela permet de considérer des données décrites par des variables de différents types (quantitatives, qualitatives,...). Cet approche est basée sur des multigraphes, où les différentes arêtes entre sommets correspondent à différentes variables ou groupes de variables. On définit un multigraphe associant à chaque sommet un élément de l'ensemble à classifier ; on peut alors obtenir des partitions φ-projetées, qui résultent de la projection des multigraphes sur des graphes simples, ou des partitions consensus des partitions associées à chaque variable/groupe de variables.

1 Méthode de Classification basée sur la coloration de graphes : rappel

Soit $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ l'ensemble de n éléments à classifier et d une mesure de dissimilarité sur Ω. On définit un graphe $G(V, E)$ sur Ω, associant à chaque sommet de V un élément de Ω ; il y a une arête reliant deux sommets du graphe si la dissimilarité entre les correspondants éléments de Ω est supérieure à la valeur d'un paramètre de contrôle α.

Dans [1], un algorithme de classification non-hiérarchique basé sur la coloration de graphes, et un indice de classification qui permet d'identifier la partition qui s'ajuste au mieux à la structure de l'ensemble donné ont été proposés. Cet indice compte le nombre d'arêtes manquantes pour que chaque paire de sommets appartenant à des classes différentes soient adjacents, ce qui correspondrait à la situation où la dissimilarité entre éléments de classes différentes était toujours supérieure à α, et entre éléments d'une même classe toujours inférieure à α. Le nombre de classes et la partition qui s'ajuste au mieux aux données, parmi celles obtenues pour différentes valeurs de α, sont identifiés par un minimum local des valeurs de l'indice de classification.

2 Classification avec des Multigraphes

Soit $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ où chaque $\omega_i, i = 1, \ldots, n$, est caractérisé par p variables, $Y_1, \ldots, Y_p$ quantitatives ou qualitatives, et soit $x_{ij} = Y_j(\omega_i)$ la valeur de la variable j sur ω_i. Considérons les variables regroupées en T groupes, selon leur type, quantitative ou qualitative (ou selon un autre critère). Soit B_t l'ensemble des indices des variables appartenant au groupe t, $t \in \{1, \ldots, T\}$.

2.1 Mesures de dissimilarité

La mesure de dissimilarité globale entre deux individus ω_i et ω_ℓ est définie par $d(\omega_i, \omega_\ell) = \sum_{t=1}^{T} \beta_t d_t^{type(t)}(\omega_i, \omega_\ell)$, où, $type(t) \in \{QUAN, QUAL\}$ et β_t est un paramètre de pondération associé au groupe B_t (il sera fixé à 1 pour des groupes de variables quantitatives et à 30% de la moyenne des écarts-types des variables quantitatives pour des groupes correspondant à des variables qualitatives [3]).

Pour des variables quantitatives, $d_t^{QUAN}(\omega_i, \omega_\ell) = \left(\sum_{j \in B_t} |x_{ij} - x_{\ell j}|^2\right)^{\frac{1}{2}}$; pour des variables qualitatives, $d_t^{QUAL}(\omega_i, \omega_\ell) = \sum_{j \in B_t} \delta(x_{ij}, x_{\ell j})$ où $\delta(x_{ij}, x_{\ell j}) = 1$ si $x_{ij} \neq x_{\ell j}$ et $\delta(x_{ij}, x_{\ell j}) = 0$ si $x_{ij} = x_{\ell j}$. Les coordonnées du centre de gravité de Ω sont données par les moyennes des variables quantitatives et les modes des variables qualitatives.

2.2 Deux Méthodes basées sur un Multigraphe

On définit le multigraphe $MG(V, E)$ sur Ω avec n sommets et m arêtes associant à chaque sommet v_i de V un élément ω_i de Ω, et il y a une arête reliant v_i, v_ℓ dans la couche t (correspondant au groupe B_t), si $d_t^{type(t)}(\omega_i, \omega_\ell) \geq \alpha_t \in I\!R; t = 1, \ldots, T$. On fait varier la valeur de α_t entre $\ell_t^{min} \geq d_t^{min}$ et $\ell_t^{max} \leq d_t^{max}$, où d_t^{min} et d_t^{max} sont respectivement les dissimilarités minimale et maximale dans la couche t entre paires d'éléments, divisant les intervalles $[\ell_t^{min}, \ell_t^{max}]$ en H_t sous-intervalles de longueur identique.
On considère alors deux approches : des partitions φ-projetées, résultant de la projection de multigraphes sur des graphes simples, et des partitions consensus, obtenues à partir des partitions associées à chaque groupe de variables.

2.2.1 Partitions projetées

Soit φ un entier tel que $1 \leq \varphi \leq T$ et soit $\eta(v_i, v_\ell)$ le nombre d'arêtes reliant les sommets v_i et v_ℓ de $MG(V, E)$. Un graphe *φ-projeté* est tout simplement un graphe qui a le même ensemble de sommets du multigraphe et où il y a une arête reliant les sommets v_i et v_ℓ si $\eta(v_i, v_\ell) \geq \varphi$.
On applique alors la méthode décrite dans la partie 1 au graphe obtenu par la *φ-projection.* Parmi les $N = \Pi_{t=1}^{T} H_t$ partitions obtenues, une pour chaque combinaison de valeurs des $\alpha_t, t = 1, \ldots, T$, on séleccionne celle correspondant à la valeur maximale de l'*Indice d'Évaluation de Partitions (IAP).* Cet indice évalue une partition suivant le principe que le nombre de voisins communs entre éléments d'une même classe doit être supérieur au nombre de voisins communs entre éléments de classes distinctes.

2.2.2 Partitions Consensus

Une autre approche consiste à déterminer une *partition consensus* des partitions obtenues pour chaque groupe de variables, et où la mesure de dissimilarité est basée sur le

nombre de fois qu'une paire d'éléments appartient à la même classe pour les différentes partitions obtenues avec chaque groupe de variables. La méthode basée sur la coloration de graphes, décrite briévement dans la partie 1, est appliquée à chaque groupe de variables, d'où il résultent T partitions, $P_1, \ldots, P_T$. La *partition consensus*, P_c, est alors celle qui correspond à $min \sum_{t=1}^{T} \Delta(P_t, P_c)$, où P_c soit appartient à l'ensemble de toutes les partitions possibles de Ω, menant à une partiton consensus *médiane*, soit est une des partitions obtenues des groupes de variables, menant à une partition consensus *medoid*.

La fonction $\Delta(P_t, P_c)$ est une dissimilarité entre partitions et compte le nombre de paires d'éléments qui appartiennent à une même classe dans une des partitions et à des classes différentes dans l'autre. Considérons les T partitions (une par groupe de variables) représentées par la matrice tri-dimensionnelle $C \equiv (c_{i\ell t})$, où $c_{i\ell t} = 1$ (respectivement, 0) si ω_i, ω_ℓ appartiennent à une même classe dans P_t (respec. n'appartiennent pas à une même classe dans P_t) $(i, \ell = 1, \ldots, n;\ t = 1, \ldots, T)$. Alors, $\Delta(P_t, P_r) = \sum_{1 \leq i < \ell \leq n} (c_{i\ell t} - c_{i\ell r})^2$. Notons que si l'on normalise Δ, alors on obtient $1 - R$ où R est l'indice de Rand ([6]).

3 Applications

On présente ici les résultats obtenus par les deux méthodes présentées ci-dessus sur deux ensembles de donnéees : les données "Zoo" et les Iris. Dans chaque cas, l'indice de Rand corrigé (CRI) ([4]) est utilisé pour mesurer l'accord entre la partition de référence et la partition obtenue. Rappellons que $-1 \leq CRI \leq 1$, et que le plus est élevée la valeur de CRI, plus grand est l'accord entre les deux partitions. Pour comparaison, on a également appliquée aux deux ensembles de données une méthode de type nuées dynamiques, "K-Prototypes" ([3]), qui coïncide avec la méthode des nuées dynamiques, variante du centre de gravité, pour des donnéees purement quantitatives ; ainsi qu'une méthode basée sur le partitionnement de graphes, "METIS" ([5]).
Dans chaque cas, CMGC dénote la méthode basée sur la coloration d'un graphe (simple) (voir partie 1), φ-pp dénote la partition obtenue par une φ-projection, MédianeCP la partition consensus médiane et MedoidCP la partition consensus medoid ; K est le nombre de classes et T le nombre de groupes de variables considérés.

3.1 Les données "Zoo"

Les données "Zoo" (*UCI Machine Learning Repository*) consistent en un ensemble de 101 animaux caracterisés par 15 variables binaires plus une 1 variable quantitative et organisés en 7 classes : mammifères (41 animaux), oiseaux (20 animaux), reptiles (5 animaux), poissons (13 animaux), amphibiens (4 animaux), flyers (8 animaux) et invertebrés (10 animaux). Le Tableau 1 présente les résultats obtenus.
On constate que la partition la plus proche de la partition de référence est la partition consensus médiane, quand toutes les variables sont prises individuellement ($T = 16$), suivie de la partition consensus medoid, pour $T = 2$ (les variables binaires formant un groupe et la seule variable quantitative le 2ème groupe), et de la partition obtenue para la méthode basée sur la coloration d'un graphe simple (CMGC).

3.2 Les données Iris

Cet ensemble consiste en 150 fleurs, 50 de chacune de trois espèces, "Iris Setosa", "Iris Versicolor", et "Iris Virginica" caractérisées par 4 variables quantitatives décrivant

	K	CRI
CMGC	19	0,729
CMGC/φ-pp/$\varphi = 1/T = 2$	7	0,517
CMGC/φ-pp/$\varphi = 2/T = 2$	6	0,514
CMGC/MédianeCP/$T = 16$	11	0,887
CMGC/MédianeCP/$T = 2$	21	0,715
CMGC/MedoidCP/$T = 16$	2	0,585
CMGC/MedoidCP/$T = 2$	10	0,807
METIS (K = 7)	7	0,381
k-Prototype (K = 7)	7	0,426

TAB. 1 – Résultats pour les données "Zoo".

	K	CRI
CMGC	2	0,568
CMGC/φ-pp/$\varphi = 4/T = 4$	5	0,859
CMGC/φ-pp/$\varphi = 1/T = 2$	2	0,568
CMGC/MédianeCP/$T = 4$	3	0,560
CMGC/MédianeCP/$T = 2$	2	0,568
CMGC/MedoidCP/$T = 4$	4	0,837
CMGC/MedoidCP/$T = 2$	2	0,568
METIS (K = 3)	3	0,786
k-Prototype (K = 3)	3	0,554

TAB. 2 – Résultats pour les données Iris.

la longueur et la largeur de pétales et des sépales de chaque fleur. La classe correspondant à l'espèce "Iris Setosa" est bien séparée des deux autres, qui sont plus proches entre eux. Le Tableau 2 présente les résultats obtenus. La partition qui s'approche au mieux de celle définie par les trois espèces a été obtenue par la méthode basée sur les φ-projections, pour $T = 4$, et avec $\varphi = 4$, suivie de celle basée sur le consensus medoid, quand toutes les variables sont prises séparément ($T = 4$).

4 Conclusions et Perspectives

Dans ce travail, on propose une extension d'une méthode de classification basée sur la coloration de graphes, qui permet de considérer des données décrites par des variables de différents types. Les variables peuvent être prises en compte individuellement ou en groupes, permettant ainsi d'obtenir des partitions dans des sous-espaces, chaque variable ou groupe de variables ayant un rôle individuel dans l'identification des classes.
Cette méthodologie s'est montré efficace quand elle est appliquée à des données simulées aussi bien qu'à des données réelles, que ce soit par les partitions φ-projetées ou par des partitions consensus obtenues à partir des partitions déterminées dans des sous-espaces de variables.
Des perspectives de développement concernent la définition de critères pour la formation des groupes de variables, et la considération d'autres approches de consensus.

Références

[1] H. Brás Silva, P. Brito, J. Pinto da Costa, "A Partitional Clustering Algorithm Validated by a Clustering Tendency Index based on Graph Theory", *Pattern Recognition* 39(5), 2006, 766-788.

[2] A. D. Gordon et M. Vichi, "Partitions of Partitions", *Journal of Classification*, 15, 1998, 265-285.

[3] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", *Data Mining, Knowledge Discovery* 2(3), 1998, 283-304.

[4] L. Hubert et P. Arabie, "Comparing Partitions", *Journal of Classification,* 2, 1985, 193-218.

[5] G. Karypis et V. Kumar, "A fast et high quality multilevel scheme for partitioning irregular graphs", *SIAM Journal on Scientific Computing*, 20, 1, 1998, 359-392.

[6] W. M. Rand, "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical Association*, 66, 1971, 846-850.

Méthode de stabilisation par rééchantillonnage dans les nœuds pour construire des arbres de classification

B. Briand[1], C. Mercat-Rommens[1] and G. Ducharme[2]

1. IRSN/DEI/SESURE/LERCM, Centre de Cadarache, Bât 153 – BP 3, 13115 Saint-Paul-lez-Durance
2. Institut de mathématiques et de modélisation de Montpellier, cc 051, Université Montpellier II, Place Eugène Bataillon, 34095 Montpellier Cedex 5
benedicte.briand@irsn.fr

Mots clés : apprentissage statistique, discrimination, radioécologie, stabilisation.

1. Introduction

Le constat sur le terrain des conséquences radiologiques de rejets radioactifs, notamment dans le cas du retour d'expérience de l'accident de Tchernobyl, montre que les conséquences pour l'homme et l'environnement d'une pollution d'origine industrielle dépendent de l'importance et de la nature de celle-ci, mais également du territoire qui la reçoit. Ce constat a suscité le développement à l'IRSN (Institut de Radioprotection et de Sûreté Nucléaire) du projet de recherche SENSIB [9,10], acronyme pour sensibilité radioécologique, afin d'identifier les spécificités des territoires français qui influent fortement sur le devenir d'un contaminant radioactif dans l'environnement. La connaissance sur les caractéristiques des territoires pourra alors être utilisée, de façon anticipée par rapport aux situations accidentelles, pour émettre des recommandations en matière de gestion des territoires contaminés et hiérarchiser la prise de décision.

L'objectif de ce travail est de développer une méthodologie permettant d'identifier les différentes caractéristiques environnementales ou anthropiques qui vont influer sur les niveaux de contamination radioactive des végétaux. Les arbres de classification ont alors été identifiés comme un axe de recherche appliqué particulièrement intéressant en raison de :

- leur pouvoir explicatif : les règles de décision fournies par les différentes branches de l'arbre permettent de mettre en évidence les associations de variables explicatives qui conduisent aux différentes classes de la variable à expliquer,
- leur simplicité de représentation et d'interprétation qui en fait des outils facilement utilisable par les non-spécialistes.

La méthodologie développée consiste à construire des arbres de classification à partir de données simulées issues d'un modèle radioécologique de transfert. Pour parer le problème d'instabilité des arbres de classification et conserver leur structure, une méthode de stabilisation par rééchantillonnage dans les nœuds est utilisée. Tout au long de la présentation la méthodologie est illustrée à partir d'un exemple de contamination radioactive accidentelle de strontium 90 sur une production de laitue.

2. Génération d'échantillons artificiels de données

Pour construire de tels arbres, des données relatives à des contaminations radioactives accidentelles sur le territoire français sont alors nécessaires. A la suite de l'accident de Tchernobyl, de nombreuses mesures de radioactivité ont été effectuées notamment dans les milieux agricoles. Cependant, les caractéristiques de prélèvements associées à ces mesures sont souvent peu nombreuses et imprécises. Ce manque d'information nous a donc conduit à recourir à un modèle radioécologique de transfert des radionucléides dans l'environnement pour générer des échantillons artificiels de données. Les variables d'entrées du modèle sont

alors identifiées comme les différentes caractéristiques environnementales et anthropiques susceptibles de modifier les conséquences d'une pollution. La variable de sortie du modèle représente la contamination radioactive du végétal que l'on a choisi de coder selon certaines limites fixées par la radioprotection. La caractérisation des distributions et des éventuelles relations entre les variables d'entrées du modèle a donc été nécessaire pour évaluer la variable de sortie (contamination du végétal) et créer des échantillons artificiels de données.

3. Méthodes de construction d'arbre de classification

3.1 Méthode CART

Dans un premier temps, la méthode CART [1] a été utilisée pour le traitement des données simulées. L'arbre final obtenu par cette méthode repose sur l'application successive des trois étapes suivantes qui distinguent CART d'autres méthodes de segmentation comme CHAID [7] ou ID3 [11] :

1) *Construction de l'arbre maximal*. L'échantillon de données est divisé de façon successive de manière à construire un arbre très détaillé A_{max}. Le processus de division est stoppé quand le nœud est pur ou quand le nombre d'observations dans le nœud est inférieur à un effectif fixé. L'arbre obtenu est très dépendant de l'échantillon qui a permis sa construction.

2) *Procédure d'élagage*. Une séquence d'arbres est construite entre A_{max} et sa racine. Chaque arbre A_{i+1} de la séquence est obtenu par suppression des branches de A_i les moins informatives. Un paramètre de complexité α est calculé, et au fur et à mesure qu'il augmente de plus en plus de branches sont élaguées conduisant à des arbres de plus en plus petits.

3) *Choix de l'arbre optimal*. Parmi cette séquence d'arbres se trouve l'arbre optimal. La sélection est basée sur l'estimation du taux de mauvais classement à l'aide d'un échantillon de validation (qui n'a pas participé à la construction de l'arbre) ou d'une validation croisée.

Appliquée à notre exemple de contamination, la méthodologie présentée précédemment nous permet de construire différents arbres de classification. Le critère d'hétérogénéité[1] choisi est l'entropie et dans chaque cas trois échantillons de taille 5000 sont générés pour construire et valider ces arbres : un échantillon d'apprentissage (pour construire l'arbre maximal et la séquence de sous arbres), un échantillon de validation (pour sélectionner le sous arbre optimal de la séquence selon la règle de l'écart-type [1]) et un échantillon test (pour estimer le taux de mauvais classement de l'arbre sélectionné).

D'un arbre à l'autre les résultats obtenus sont assez instables. Cette instabilité se marque à différents niveaux : nombre de feuilles, prédictions ainsi que sur le choix des divisions. Ce type d'instabilité est assez connu dans le domaine des arbres de décision [2,6]. La variabilité observée sur les divisions et donc sur les règles de décision est assez gênante dans le cadre de notre application car nous voulons utiliser ces règles pour proposer des recommandations dans un contexte post-accidentel. Il est donc nécessaire de disposer de règles robustes.

3.2 Méthode de rééchantillonnage dans les nœuds

Pour parer l'instabilité des arbres de classification, des méthodes basées sur l'agrégation d'arbres sont proposées comme les techniques d'agrégation par bootstrap Bagging [3] ou les forêts aléatoires Random Forest [4]. Elles permettent d'améliorer nettement les prédictions mais ne rendent plus disponible la structure de l'arbre et donc l'ensemble des règles de décision qui en découlent. Dans le but de préserver la structure de l'arbre et pour obtenir des règles de classification plus stables, une méthode de stabilisation par rééchantillonnage dans les nœuds a été utilisée [5]. Cette approche consiste à construire l'arbre selon l'algorithme suivant :

[1] Ce critère va mesurer le degré de mélange des classes dans un nœud : il est maximal lorsque le mélange est le plus important et nul lorsque le nœud contient des observations d'une seule classe de la variable à expliquer.

Pour chaque nœud t de taille L_t *Faire* :
Pour b=1 à B *Faire* :
Générer 1 échantillon bootstrap $L_t(b)$
Rechercher la division optimale sur chacune des variables explicatives
Fin Pour
Sélection de la variable qui sera utilisée pour effectuer la division (vote à la majorité)
Détermination de la division pour la variable choisie :
division=médiane(réplications bootstrap)
Fin Pour

Cette procédure permet de stabiliser l'arbre de classification tout en conservant sa structure, les règles de décision ainsi obtenues sont plus « robustes ». En lien avec le contexte de l'étude, nous ne voulons pas obtenir des arbres trop détaillés, car ils seront difficilement utilisables en contexte post-accidentel, c'est pourquoi nous choisissons de limiter notre arbre maximal à 5 niveaux (le premier niveau est représenté par le nœud racine, le deuxième niveau par les deux nœuds enfants issus du nœud racine,…).
Pour élaguer cet arbre nous proposons d'utiliser la méthode suivante, plus adaptée à notre problématique environnementale :
1. Les branches apportant aucune information sont supprimées. On définit alors un critère qui va mesurer la réduction du taux de mauvais classement entre un nœud et les feuilles issues de ce nœud :

$$R_{mc}(t) = \frac{mc(t) - mc(F_t)}{N(t)}$$

où *mc(t)* représente le nombre d'observations mal classées au nœud t ; $mc(F_t)$ représente le nombre d'observations mal classées dans les feuilles issues du nœud t ; *N(t)* représente le nombre d'observations au nœud t.
La procédure consiste alors à choisir les nœuds intermédiaires pour lesquels $R_{mc}(t)$ est nul et à élaguer l'arbre aux nœuds sélectionnés.
2. L'expert peut intervenir dans la procédure d'élagage en supprimant les branches qui lui paraissent peu pertinentes.

4. Résultats

Appliquée à notre exemple de contamination radioactive de la laitue par du strontium 90, cette méthodologie nous permet d'obtenir l'arbre de classification présenté en figure 1. Pour chaque nœud terminal, l'effectif dans le nœud, le taux de mal classés, ainsi que le pourcentage d'observations mal classées sont explicités (estimés par un échantillon test de taille n_{test}=5000). Le pourcentage de mauvais classement estimé par l'échantillon test est de l'ordre de 5,44%. A partir du même échantillon d'apprentissage un arbre de classification a été construit par la méthode CART et le taux de mauvais classement a été estimé par l'échantillon test utilisé précédemment. Les méthodes de type Bagging et Random Forest ont aussi été appliquées. Les résultats relatifs à l'estimation du taux de mauvais classement sont présentés sur la figure 2 sous forme de pourcentages.

CART	Rééchantillonnage dans les nœuds	Bagging	Random Forest
5,46	5,44	3,96	3,94

Figure 2 : Comparaison du pourcentage de mauvais classement pour les quatre méthodes

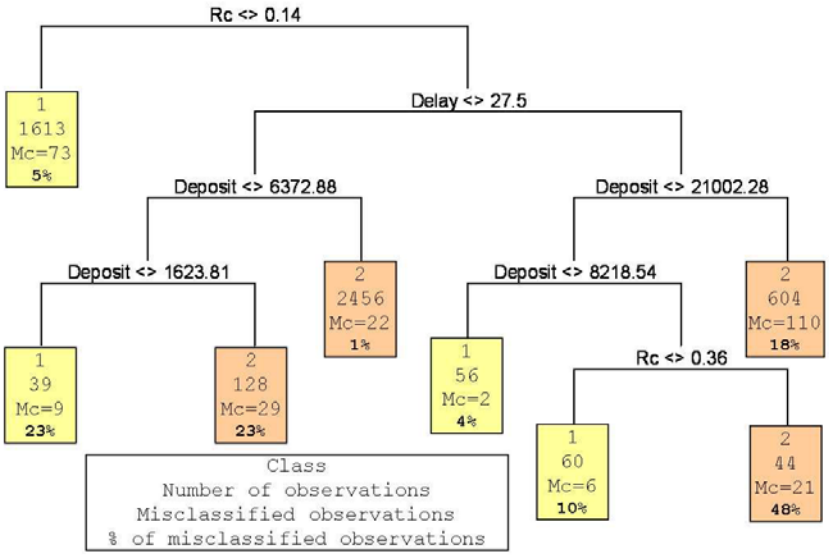


Figure 1 : Arbre de classification obtenu par la méthode de rééchantillonnage dans les nœuds (Rc : Rapport de captation le jour de l'accident ; Delay : Délai entre le dépôt et la récolte de la production de laitue ; Deposit : Dépôt de radioactivité le jour de l'accident)

Comme l'on pouvait s'y attendre les deux méthodes basées sur l'agrégation d'arbres sont plus performantes. Globalement, la méthode CART et la méthode de rééchantillonnage dans les nœuds sont équivalentes. Cependant l'arbre construit selon notre méthodologie présente des règles de décision plus robustes et l'élagage manuel nous permet de mieux appréhender certains mécanismes de contamination des végétaux (ici la laitue).

Références

[1] Breiman L., Friedman J.H., Olshen R., and Stone C.J., Classification and Regression Trees, Wadsworth, Belmont CA, 1984.
[2] Breiman L., Heuristics of instability and stabilization in model selection. The Annals of Statistics, 1996, 24(6):2350—2383.
[3] Breiman L., Bagging predictors. Machine Learning, 1996, 24, pp.123-140.
[4] Breiman L., Random Forest. Machine Learning, 2001, 45(1), 5-32.
[5] Dannegger F., Tree stability diagnostics and some remedies for instability. Statistics in Medicine, 2000, 19:475-491.
[6] Ghattas B., Agrégation d'arbre de classification. Revue de Statistique Appliquée, 2000, tome 48 n°2, pp. 85-98.
[7] Kass G. V., An exploratory technique for investigating large quantities of categorical data, Applied Statistics, 1980, 29(2), 119-127.
[8] Limites indicatives pour les radionucléides dans les aliments, applicables dans le commerce international à la suite d'une contamination nucléaire accidentelle (CAC/GL 5-1989), Genève, 1989.
[9] Mercat-Rommens, C. and Renaud, P., From radioecological sensitivity to risk management: the SENSIB Project. Second International Conference Radioactivity in the Environment, Nice, October 2005.
[10] Mercat-Rommens, C., Roussel-Debet, S., Briand, B., Durand, V., Besson, B. et Renaud, P., La sensibilité radioécologique des territoires : vers un outil opérationnel – Le projet SENSIB. Radioprotection, 2007.
[11] Quinlan R., Discovering rules by induction from large collections of examples, D. Michie ed., Expert Systems in the Microelectronic age, 1979, pp. 168-201.

Sélection non supervisée d'attributs - Application à l'indexation d'images satellitaires.

M. Campedel, I. Kyrgyzov et H. Maître

ENST, Département TSI,
46, rue Barrault, 75634 Paris cedex 13
(Marine.Campedel, Ivan.Kyrgyzov, Henri.Maitre)@enst.fr

Mots clés : Analyse multidimensionnelle, graphes et classification, application à l'imagerie satellitaire.
Remerciements : Ce travail est effectué dans le cadre du Centre CNES/DLR/ENST "Competence Center on Information extraction and Image Understanding for Earth Observation".

Introduction

Les images satellitaires, nombreuses et au contenu complexe, sont encore souvent exploitées manuellement par des experts du domaine d'application qui les requiert. Ces images sont particulièrement nombreuses (100Go de données journalières pour le seul satellite SPOT) et finalement faiblement exploitées. En outre, de nouveaux satellites, tels les satellites Pleiades, produisant des images à très haute résolution (THR) seront lancés prochainement, donnant accès chacun à 450 images par jour[1], avec une résolution de 70cm par pixel. L'exploitation manuelle de l'ensemble de ces données n'est pas concevable, il est urgent de développer des algorithmes performants, automatiques, facilitant leur accès. Selon les applications visées, les objets d'intérêt dans ces images ne sont pas identiques ; l'indexation des images satellitaires ne peut donc se faire qu'à partir d'analyses non supervisées (sans spécification a priori de classes sémantiques d'intérêt).

Cet article s'appuie sur nos travaux précédents [1] ; ceux-ci présentent une méthodologie permettant de choisir les caractéristiques les mieux adaptées à l'indexation des images satellitaires. L'idée majeure est d'exploiter l'ensemble des résultats exposés dans la littérature (i.e. concaténer tous les attributs a priori pertinents) et de parvenir à identifier le sous-ensemble d'attributs le moins redondant (pour une question de coût de calcul et de stockage) et le plus "informatif" possible, à l'aide d'algorithmes de sélection automatique.

Nos expériences passées, reposant sur une comparaison de méthodes de sélection supervisées et non supervisées, ont démontré l'efficacité de méthodes simples, rapides et non supervisées, pour réduire la redondance introduite volontairement par la concaténation des attributs. Ces méthodes non supervisées fonctionnent en deux étapes : la première permet de grouper les attributs similaires et la seconde identifie des représentants pour chacun des groupes obtenus. L'exemple le plus simple (et finalement le plus efficace à travers nos expériences) consiste à appliquer un algorithme des K-Moyennes sur les attributs (et non les données [2]) et à conserver l'attribut le plus proche de chaque centroïde. Lors de

[1] http ://www.cnes.fr/web/print-3227-pleiades.php

[2] Nous représentons nos données à l'aide d'une matrice numérique de N lignes et D colonnes, N étant le nombre d'exemples et D leur dimension. Notre méthode de sélection passe par une clusterisation des D colonnes.

ces précédents travaux, quelques questions sont restées en suspens telles que : i) le choix du nombre de caractéristiques, paramètre de toutes les méthodes de sélection utilisées (supervisées ou non), ii) l'influence de l'initialisation de l'algorithme des K-Moyennes, iii) l'analyse exploratoire des attributs sélectionnés. Afin d'y répondre, nous introduisons une méthode de consensus de clusterisations, qui nous affranchit des problèmes d'initialisation tout en offrant certaines possibilités exploratoires.

Classification consensuelle

Combiner des classificateurs n'est pas nouveau, en particulier dans la communauté de la classification supervisée. Les objectifs sont généralement d'améliorer des performances de classification, d'identifier des résultats communs à plusieurs algorithmes, de classifier des données partiellement étiquetées ou des données distribuées, ... Cependant, lorsqu'il s'agit de classification non supervisée le problème n'est pas simple (pas de classes prédéfinies ni mesures d'erreurs bien établies) et la littérature est bien moins fournie. Par ailleurs, exploiter une méthode de combinaison de clusterisations (appelée aussi consensus) peut être considéré comme une méthode très intuitive de fouiller les données. Chaque algorithme de clusterisation fournit un "point de vue" sur la distribution des données ; l'idée du consensus est d'identifier l'information commune portée par les différents résultats de classification pour ensuite en analyser la redondance et la complémentarité.

Nous proposons ainsi d'exploiter une méthode de consensus, reposant sur une représentation par matrice de co-association des clusterisations individuelles. Ces matrices illustrent le fait que deux données (parmi les N considérées) sont classifiées ou non dans le même groupe (cluster) ; elles ne dépendent pas d'un processus de numérotation des clusters et ne reflètent pas non plus les espaces de caractéristiques dans lequel sont représentées les données. Différents critères objectifs de combinaison ont déjà été proposés [3, 4] ; nous en proposons un, bien formalisé mathématiquement, pour lequel nous proposons une solution efficace s'affranchissant de tout paramètre d'initialisation. En effet, à partir de la matrice de co-association moyenne A, le problème est de trouver la matrice de partition consensuelle B^s (binaire) telle que $B^s B^{s\prime}$ [3] est la plus proche de A au sens des moindres carrés. L'exploration de l'ensemble des solutions n'est pas envisageable, du fait de la quantité de données qui nous intéresse. Nous proposons donc une heuristique, reposant sur l'algorithme agglomératif appelé "single-link". L'algorithme, dénommé LSEC (Least Square Error Combination), procède itérativement en réduisant progressivement l'erreur E ; il est présenté ci-après :

LSEC

Etape 1 Initialisation de B^s (avec la matrice identité), $i \leftarrow 1$ et calcul de l'erreur initiale $E^{(i)} = \sum_{u=1}^{N}\sum_{v=1}^{N}\left(\sum_{r=1}^{N}(B^s_{ur}B^{s\ \prime}_{rv}) - A_{uv}\right)^2$

Etape 2 Trouver les données appariées avec une probabilité maximale $(r,t) = max\{A_{uv} : u, v = 1,\ldots,N, u \neq v\}$

Etape 3 Si $A_{rt} = 0$, B^s est la partition recherchée, stop.

Etape 4 Mise à jour de A avec $A_{rt} \leftarrow 0$, $B^h \leftarrow B^s$.
Fusionner les deux clusters C_r et C_t en sommant les colonnes respectives de B^h : $B^h_{kr} \leftarrow \left(B^h_{kr} + B^h_{kt}\right)$ et supprimer la colonne B^h_{kt} (mise à 0), avec $k = 1,\ldots,N$.

[3] ' désigne l'opération de transposition.

Mise à jour de l'erreur : $E^{(i+1)} \leftarrow \sum_{u=1}^{N}\sum_{v=1}^{N}\left(\sum_{r=1}^{N}(B^h_{ur}B'^h_{rv}) - A_{uv}\right)^2$
Si $E^{(i+1)} \leq E^{(i)}$, alors $i \leftarrow i+1$, $B^s \leftarrow B^h$, $A \leftarrow A. * (\mathbf{I} - B^s B^{s\prime})$.
Aller en **Etape 2**.

'.*' représente la multiplication de matrice point par point.
Le nombre optimal de clusters est obtenu implicitement comme étant le nombre de colonnes non nulles de B^s. En pratique, il est possible de s'affranchir du stockage de la matrice A et d'initialiser efficacement B^s à l'aide de graphes de voisinage, afin de faciliter la convergence de l'algorithme. La présentation détaillée de ce nouvel algorithme est effectuée dans [5].

L'algorithme de classification consensuel est exploité dans une procédure de sélection d'attributs appelée KMeans-FS-C. Les N données à clusteriser sont alors les D attributs issus de la littérature. Les clusterisations individuelles sont produites à l'aide d'un algorithme de K-Moyennes. En pratique, pour des raisons de temps de calcul, nous appliquons les K-Moyennes avec K allant de 2 à D avec un pas de 10 et 20 initialisations aléatoires pour chaque valeur de K. Les représentants choisis pour le cluster C de taille $\#C$ sont les attributs les plus "stables" de chaque cluster résultant. La stabilité est évaluée par $S_{d\in C} = 1/\#C \sum_{n=1,n\in C}^{D} A_{dn}$.

Expérimentations

Pour nos expérimentations, nous disposons de deux bases de travail distinctes. L'une, étiquetée manuellement (Sat3600), modélise notre application finale (en l'occurrence, une classification) ; l'autre, non étiquetée (SpotRdn), est exploitée pour effectuer la sélection des attributs. Les images satellitaires traitées sont toutes issues de scènes SPOT 5 HMA panchromatiques de résolution de 5m par pixel. Il s'agit d'imagettes de taille 64 × 64 pixels ; chacune est considérée comme un échantillon et fournit une signature représentée par un vecteur numérique dont la taille dépend des caractéristiques extraites : les images SPOT 5 contiennent une information essentiellement texturale, nous proposons donc de comparer différents modèles de texture (coefficients d'Haralick, statistiques après décompositions sur une base d'ondelettes de Gabor et QMF) auxquels nous ajoutons quelques attributs géométriques issus d'une analyse des segments linéaires extraits des imagettes ; au total $D = 143$ attributs sont considérés.

Dans la base d'évaluation, les exemples ont été sélectionnés manuellement, illustrant sans ambiguïté les classes prédéfinies (ville, forêt, champs, mer, désert et nuage). La base aléatoire a été obtenue en extrayant des imagettes uniformément dans chacune des 32 scènes SPOT à notre disposition. Au total, plus de 25000 imagettes ont été ainsi collectées. Faire une clusterisation par K-Moyennes de 143 exemples (attributs) ayant une dimension de 25000 pose un probleme ("curse of dimensionality"), nous appliquons donc également une procédure de sélection non supervisée des imagettes aléatoires ; 100 imagettes sont finalement retenues. L'étude détaillée de cette sélection n'est pas présentée dans le cadre de cet article.

Résultats

Les performances de classification sont présentées dans la table 1. Nous remarquons que les taux d'erreur obtenus sont très bons, ce qui signifie que les attributs extraits permettent de répondre à nos attentes applicatives. En outre, après sélection, le potentiel discriminatoire des attributs retenus est équivalent, ce qui valide l'usage de notre méthode non

supervisée de sélection.

	Kppv(3)	SVM (rbf 10)	D
Tous les attributs	3.5±0.8	1.7±0.6	143
Consensus	3.7±0.6	1.5±0.4	28

TAB. 1 – Evaluation du taux d'erreur moyen de classification (%) de la base Sat3600, sans sélection et après sélection des attributs par KMeans-FS-C. Le résultat présente la moyenne et l'écart-type de l'erreur obtenue en validation croisée, avec deux classificateurs différents.

Le deuxième aspect intéressant de la méthode de sélection par consensus vient de ses capacités exploratoires. Dans le cadre de notre expérience, la classification consensuelle contient 76 clusters dont 48 unitaires. Ces 48 clusters unitaires correspondent à des attributs situés à la frontière de clusters plus conséquents (ceci se déduit directement des valeurs contenues dans la matrice de co-association moyenne A). Ayant choisi de représenter les clusters par les attributs les plus stables, nous ne retenons pas ces attributs isolés, la sélection finale contient donc 28 clusters. Cette analyse permet ainsi de différencier trois types d'attributs : i) ceux qui sont faciles à classer ii) ceux qui se retrouvent à la frontière des clusters les mieux définis iii) les isolés (outliers).

Conclusion et perspectives

Nous avons présenté une méthode originale de sélection d'attributs, non supervisée, reposant sur une procédure de classification consensuelle. Cette méthode, dérivée à l'aide de l'algorithme des K-Moyennes, a prouvé son efficacité au travers d'une application de classification. Elle a permis de répondre aux trois problèmes que nous nous posions, à savoir : l'estimation du nombre d'attributs à sélectionner, la gestion des paramètres des méthodes de clusterisation ainsi que les moyens d'analyse des résultats produits.

Les perspectives sont multiples. L'approche doit encore être validée sur des tâches de classification plus complexes, notamment sur des données obtenues à partir d'images de très haute résolution pour lesquelles les "bonnes caractéristiques" ne sont pas disponibles dans la littérature. Enfin, la méthodologie présentée doit être dérivée avec différents algorithmes de clusterisation et/ou de consensus de classifications.

Références

[1] M. Campedel and E. Moulines, "Classification et sélection automatique de caractéristiques de textures", *RNTI*, 2004, C-1, 25-37.

[2] J. Weston, A. Elisseeff, G. Bakir, F. Sinz, "The Spider for Matlab - v1.4", Max Planck Institute for Biological Cybernetics, department : Empirical Inference for Machine Learning and Perception http ://www.kyb.tuebingen.mpg.de/bs/people/spider, 2004.

[3] A. Jain, R. C. Dubes, "Algorithms for Clustering Data", *Prentice-Hall, Englewood Cliffs, NJ*, 1988.

[4] A. Topchy, A.K. Jain, W. Punch, "A Mixture Model for Clustering Ensembles", *in Proc. SIAM Conf. on Data Mining*, 2004, 379-390.

[5] Y. Kyrgyzov, H. Maître, M. Campedel "A method of clustering combination applied to satellite image analysis", *ICIAP*, 2007, à paraître.

Une approche divisive de classification hiérarchique de variables quantitatives

M. Chavent[1], V. Kuentz[1] and J. Saracco[1,2]

1. Institut de Mathématiques de Bordeaux (IMB), UMR CNRS 5251, Université Bordeaux 1, 351 Cours de la Libération, 33405 Talence Cedex, France
2. GREThA, UMR CNRS 5113, Université Montesquieu - Bordeaux IV, Avenue Léon Duguit, 33608 Pessac Cedex, France
(Marie.Chavent, Vanessa.Kuentz, Jerome.Saracco)@math.u-bordeaux1.fr

Mots clés : classification automatique, applications.

Résumé

La plupart des méthodes de classification existantes ont été développées pour la typologie d'individus. En ce qui concerne la classification de variables, peu de méthodes existent. Elles peuvent pourtant être utilisées dans différentes situations : réduction de dimension, sélection de variables, etc. Nous présentons une approche divisive de classification hiérarchique de variables quantitatives en lien avec la procédure VARCLUS (VARiables CLUStering) du logiciel SAS.

Introduction. Avec l'émergence de bases de données toujours plus grandes, la réduction de dimension est un problème central en Statistique. Il s'agit de remplacer un grand nombre de variables par un nombre plus restreint, en minimisant la perte d'information. Les techniques les plus utilisées sont les méthodes factorielles (A.C.P., A.F. ou A.C.M.). Cependant, la classification de variables peut aussi être une solution. En effet, ces méthodes permettent de créer des groupes de variables corrélées, à partir desquels on peut extraire des représentants des classes, encore appelés variables synthétiques. Cet ensemble réduit de variables est alors plus facile à gérer et à interpréter lors d'analyses ultérieures.

La classification de variables peut également être utilisée dans des problèmes de sélection de variables, comme par exemple en régression multiple ou en analyse discriminante. Elle est une alternative aux techniques de sélection de variables développées par exemple dans [7], [8] ou [9].

Dans certaines applications, on peut vouloir s'intéresser à la classification de variables plutôt que d'observations : analyse sensorielle (mise en place de groupes de descripteurs), biochimie (classification de gènes), marketing (segmentation d'un panel de consommateurs), économie (détection de stratégies financières), etc.

La méthode divisive la plus connue à ce jour est la procédure VARCLUS du logiciel SAS. Cette méthode, largement utilisée, fournit des résultats simples (partition ou hiérarchie de variables quantitatives) mais son fonctionnement est complexe. Il est pourtant fondamental de comprendre les étapes et les différentes options de cette méthode afin de l'utiliser "correctement". Les explications données dans le guide SAS pour cette procédure sont succinctes et la compréhension détaillée de cette procédure nécessite un important travail d'investigation. On lit par exemple : "The chosen cluster is split into two clusters by finding the first two principal components, performing an orthoblique rotation (raw quartimax rotation on the eigenvectors) and assigning each variable to the rotated components with which it has the highest squared correlation". S'agit-il de la

rotation orthoblique telle qu'elle est définie dans [6] ou bien d'une rotation orthogonale quartimax sur les vecteurs propres de la matrice des corrélations entre les variables de la classe, ces vecteurs propres étant les composantes principales standardisées ? D'autres questions se posent. Par exemple, VARCLUS procède après chaque division à une phase de réallocation en deux étapes des variables. La première étape de cette phase, appelée NCS (Nearest Component Sorting), semble être un algorithme itératif de type Nuées Dynamiques, le représentant d'une classe étant la première composante principale des variables de la classe. Cet algorithme est-il appliqué à la partition en deux classes issue de la dernière division ou bien à la partition composée de toutes les classes déjà obtenues ? Selon les options choisies, VARCLUS ne maintient donc pas nécessairement une structure hiérarchique. Dans le même ordre d'idée, VARCLUS propose différentes initialisations consistant à ne pas démarrer la procédure avec une partition en une classe mais à partir d'une partition fournie par l'utilisateur, ou bien d'une partition aléatoire, etc. Toutes ces remarques montrent que VARCLUS est une procédure difficile à maîtriser, parfois plus proche du partitionnement que de la classification hiérarchique.

Dans cette communication, nous présentons une approche divisive pour la classification hiérarchique de variables quantitatives, cette méthode pouvant être vue comme une version simplifiée et cohérente de VARCLUS. Plusieurs approches ascendantes de classification hiérarchique de variables ont déjà été proposées. Elles définissent un indice de dissimilarité entre variables puis utilisent la classification ascendante hiérarchique sur matrice de dissimilarités (lien minimal, lien maximal, etc.). Cependant, la classification hiérarchique descendante, bien que moins répandue, présente des avantages par rapport aux méthodes ascendantes. Par exemple, lorsqu'on souhaite créer des groupes de variables, pour ensuite éventuellement réduire leur nombre, on s'intéressera généralement aux partitions en peu de classes, ces partitions étant obtenues dans les premières étapes d'un algorithme divisif et dans les dernières itérations d'un algorithme agglomératif.

Le critère. Pour mesurer la qualité d'une classe, la plupart des méthodes de classification de variables quantitatives utilisent comme critère (parfois sans le mentionner) la somme des carrés des corrélations des variables à la première composante principale standardisée des variables de la classe. Cela consiste à rechercher des groupes de variables fortement corrélées, sans tenir compte du signe de la liaison linéaire.

Soient $x^1, \ldots, x^p$, p variables quantitatives observées sur n individus. Soit $P_K = (C_1, \ldots, C_K)$ une partition des p variables en K classes. Nous noterons X_k la matrice formée par les variables de C_k. La qualité de la classe C_k est définie par :

$$Q(C_k) = n \sum_{x^j \in C_k} corr^2(x^j, c^k) = \frac{1}{n}(c^k)' Z_k Z_k' c^k$$

où c^k est la première composante principale standardisée de X_k et Z_k est la matrice X_k centrée-réduite. La première composante principale standardisée c^k étant le vecteur propre normé associé à la plus grande valeur propre $\lambda_1^{(k)}$ de la matrice des corrélations $\frac{1}{n} Z_k Z_k'$ des variables de C_k, on a $Q(C_k) = \lambda_1^{(k)}$. La quantité $Q(C_k)$ correspond donc à la variance de la première composante principale de la classe. La qualité de la partition P_K est ainsi définie par :

$$W(P_K) = \sum_{k=1}^{K} Q(C_k) = \sum_{k=1}^{K} \lambda_1^{(k)}. \quad (1)$$

L'algorithme de partitionnement des Nuées Dynamiques (voir [4], [5]) permet d'optimiser localement ce critère. En effet à chaque itération, la décroissance du critère à minimiser

(resp. la croissance si l'on maximise le critère) est assurée si le représentant des classes minimise (resp. maximise) le critère d'adéquation choisi. La première composante principale standardisée vérifiant $c^k = \arg\max_{u\in\mathbb{R}^n} \frac{1}{n} u' Z_k Z_k' u$, le choix de cette dernière comme représentant d'une classe C_k assure la convergence de l'algorithme vers un maximum local de (1). Il s'agit de l'algorithme utilisé dans :

- la méthode CLV (Clustering around Latent Variables) (voir [12], [13]) qui propose en outre une version du critère permettant de tenir compte du signe des corrélations ;
- la méthode Diametrical Clustering (voir [3]) développée pour la classification de gènes avec une dernière étape permettant d'identifier à partir de chaque classe deux sous-classes de gènes anti-corrélés ;
- l'algorithme NCS (Nearest Component Sorting) utilisé dans la phase de réallocation de VARCLUS.

Dans l'approche divisive de classification hiérarchique proposée dans cette communication, nous cherchons à maximiser le même critère (1). Comme pour toute méthode divisive de classification il faudra définir comment diviser une classe et comment choisir la classe à diviser. Les divisions sont arrêtées après un nombre fini d'étapes, généralement avant d'atteindre les singletons. Il s'agit donc d'une hiérarchie "partielle" (voir [10]) et cette hiérarchie doit donc nécessairement être indicée par le critère utilisé dans le choix de la classe à diviser.

Algorithme de bi-partitionnement. On cherche une bi-partition (C_l^1, C_l^2) d'une classe C_l qui maximise (1). Pour cela, on applique l'algorithme des Nuées Dynamiques avec comme représentant des classes la première composante principale standardisée. Comme nous l'avons vu, la partition finale dépend de la partition initiale. Afin d'initialiser "au mieux" l'algorithme, nous utilisons la procédure suivante :

(1) A.C.P. sur la matrice X_l.

(2) Rotation orthogonale (varimax ou quartimax) des deux premières composantes principales standardisées. La rotation permet en effet de rendre les corrélations des variables aux composantes standardisées après rotation, notées f^1, f^2, aussi proches que possible de 0 ou de 1. Nous choisissons une rotation orthogonale plus facile à comprendre et à interpréter que la rotation oblique. Une étude plus détaillée des propriétés, avantages et inconvénients de ces différents types de rotation serait utile.

(3) Affectation des variables aux sous-classes :
$C_l^1 = \{x^j | corr^2(x^j, f^1) \geq corr^2(x^j, f^2)\}$ et $C_l^2 = \{x^j | corr^2(x^j, f^1) < corr^2(x^j, f^2)\}$.

Cette partition en deux classes des variables est ensuite utilisée comme partition initiale pour le bi-partitionnement de C_l par l'algorithme des Nuées Dynamiques.

Choix de la classe à diviser. A chaque étape k, connaissant une partition des variables en k classes, on choisit de diviser la classe C_l qui fournit la meilleure partition en $k+1$ classes au sens de notre critère (1). Ce critère étant additif, on montre facilement que cela revient à choisir de diviser la classe C_l qui maximise la variation du critère suivant :

$$\Delta(C_l) = Q(C_l^1) + Q(C_l^2) - Q(C_l) = \lambda_1^{(C_l^1)} + \lambda_1^{(C_l^2)} - \lambda_1^{(C_l)}. \quad (2)$$

Les classes de la hiérarchie partielle ainsi construites seront donc indicées par $\Delta(C_k)$. Cet indice est bien toujours positif (voir par exemple [12] pour la démonstration) mais pour l'instant, nous ne sommes pas parvenus à démontrer qu'il est bien monotone croissant c'est-à-dire que si $A \subset B$ alors $\Delta(A) \leq \Delta(B)$. Notons qu'en pratique, sur l'ensemble des jeux de données que nous avons utilisés, nous n'avons jamais observé d'inversion.

Dans la procédure VARCLUS, ce critère de choix n'est pas proposé. Selon l'option choisie, VARCLUS (sur la matrice des corrélations) divise la classe C_l qui possède la plus grande seconde valeur propre ($\lambda_2^{(l)}$) ou le plus petit pourcentage de variance expliquée par la première composante principale ($\lambda_1^{(l)}/p_l$) où p_l désigne le nombre de variables de la classe C_l. La normalisation $\lambda_2^{(l)}/p_l$ semblerait cependant plus appropriée pour comparer des classes de variance et de taille différentes. De plus, choisir la classe qui maximise $\lambda_1^{(l)}/p_l$ ne veut pas dire que l'on construira la meilleure partition possible (issue d'une division) comme avec le critère (2).

Résultats. Nous avons implémenté sous le logiciel `R` cette méthode de classification descendante hiérarchique et la représentation du dendrogramme de la hiérarchie partielle indicée . Dans cette communication, nous donnerons également des résultats de comparaisons, sur des données simulées et réelles, entre l'approche proposée ici et celle de la procédure VARCLUS, ainsi qu'avec d'autres méthodes de classification de variables.

Références

[1] S. Camiz, V. de Patta Pillar, "Comparaison d'une classification hiérarchique factorielle de variables avec des méthodes classiques", *Actes du XIeme Congrès de la Société Francophone de Classification*, Bordeaux, 2004, 134-137.

[2] J.J. Denimal, "Hierarchical Factorial Analysis", *Actes du 10th International Symposium on Applied Stochastic Models and Data Analysis*, Compiègne, 2001, 369-374.

[3] I.S. Dhillon, E. M. Marcotte, U. Roshan, "Diametrical clustering for identifying anti-correlated gene clusters", *Bioinformatics*, Vol. 19, 2003, 1612-1619.

[4] E. Diday, "La méthode des nuées dynamiques", *Revue de Statistique Appliquée*, XXX (2), 1971, 19-34.

[5] E. Diday, J.C. Simon, "Clustering analysis", In: K.S. Fu (ed.): *Digital Pattern Clasification*, Springer Verlag, 1976, 47-94.

[6] H.H. Harman, *Modern Factor Analysis*, Third Edition, Chicago:University of Chicago Press, 1976.

[7] I.T. Jolliffe, "Discarding variables in a principal component analysis I:Artificial data", *Applied Statistics*, 21, 1972, 160-173.

[8] W.J. Krzanowski, "Selection of variables to preserve multivariate data structure, using principal components", *Applied Statistics*, 36, 1987, 22-33.

[9] G.P. Mccabe, "Principal variables", *Technometrics*, 26, 1984, 137-144.

[10] B. Mirkin, *Clustering for Data Mining*, Chapman & Hall, 2005.

[11] SAS Institute Inc. 2004. SAS OnlineDoc 9.1.3. Cary, NC: SAS Institute Inc.

[12] E. Vigneau, E.M. Qannari, "Clustering of variables around latent components", *Communications in statistics, Simulation and Computation*, Vol. 32, No 4., 2003, 1131-1150.

[13] E. Vigneau, E.M. Qannari, K. Sahmer, D. Ladiray, "Classification de variables autour de composantes latentes", *Revue de Statistique Appliquée*, LIV (1), 2006, 27-45.

[14] `R` version 2.4.1 - A Language and Environment Copyright, 2006.

Mesures de tendance centrale et de dispersion d'une série d'intervalles

Marie Chavent[1] et Jérôme Saracco[1,2]

1. IMB, UMR CNRS 5251, Université Bordeaux1,
351 cours de la libération, 33405 Talence Cedex
2. GREThA, UMR CNRS 5113, Université Montesquieu - Bordeaux IV
Avenue Léon Duguit, 33608 Pessac Cedex
{Marie.Chavent, Jerome.Saracco}@math.u-bordeaux1.fr

Mots clés : statistique descriptive, analyse de données symboliques, distance de Hausdorff.

Les statistiques descriptives classiques (valeurs centrales et mesures de dispersions) telles que la moyenne, la médiane, l'écart-type ou encore l'étendue peuvent être définies géométriquement comme solution d'un problème d'optimisation. Pour cela, on représente un échantillon aléatoire de n observations réelles x_i par un vecteur $\mathbf{x} = (x_1, x_2 \ldots, x_n)^t \in \mathbb{R}^n$. Une valeur centrale $c \in \mathbb{R}$ de cette série de n observations est alors définie de manière à être aussi proche que possible des x_i. On mesure cette proximité par une fonction S_p définie par

$$S_p(u) = \| \mathbf{x} - \mathbf{u} \|_p \ , \tag{1}$$

où $\| \cdot \|_p$ désigne la norme L_p sur $\mathbb{R}^n$ et $\mathbf{u} = u\mathbb{I}_n$ avec $\mathbb{I}_n$ le vecteur identité. Plus précisément, on a :

$$S_p(u) = \begin{cases} (\sum_{i=1}^n \mid x_i - u \mid^p)^{1/p} & \text{pour } p < \infty, \\ \max_{i=1\ldots n} \mid x_i - u \mid & \text{pour } p = \infty. \end{cases} \tag{2}$$

La valeur centrale de cette série est alors

$$c = \arg\min_{u \in \mathbb{R}} S_p(u)$$

et $S_p(c)$ est la mesure de dispersion associée. Ce problème de minimisation a des solutions explicites pour $p = 1$, 2, ∞ :

- Pour $p = 1$, c est la médiane notée x_M, et la mesure de dispersion associée est $S_1(x_M) = \sum_{i=1}^n \mid x_i - x_M \mid = ns_M$ où s_M est l'écart absolu à la médiane, moyen.
- Pour $p = 2$, c est la moyenne empirique notée $\bar{x}$, et la mesure de dispersion associée est telle que $S_2^2(\bar{x}) = \sum_{i=1}^n (x_i - \bar{x})^2 = (n-1)s^2$ où s^2 est la variance empirique.
- Pour $p = \infty$, $c = \frac{x_{(n)} - x_{(1)}}{2}$ où $x_{(n)}$ et $x_{(1)}$ sont respectivement la plus grande et la plus petite observation. On note x_r cette valeur centrale appelée *midrange* en anglais, et la mesure de dispersion associée est $S_\infty(x_r) = \max_{i=1\ldots n} \mid x_i - x_r \mid = (1/2)w$ où $w = x_{(n)} - x_{(1)}$ est l'étendue.

Les couples $(\bar{x}, s^2)$, (x_M, s_M) et (x_r, w) sont donc cohérents avec l'utilisation des normes L_1, L_2 et L_∞ dans la fonction S_p. Ces couples peuvent en particulier être utilisés pour "centrer et réduire" les données.

On considère maintenant un échantillon de n intervalles $\tilde{x}_i = [a_i, b_i]$. Des extensions pour le calcul de la moyenne et de la variance empirique d'une série d'intervalles et pour la construction d'histogrammes ont été proposées par [1] et [2]. Ici nous adoptons une approche différente qui consiste à généraliser les définitions géométriques rappelées ci-dessus. Pour cela on représente un échantillon aléatoire de n intervalles par un vecteur d'intervalles $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_n)^t = ([a_1, b_1], \ldots, [a_n, b_n])^t$. Un intervalle central $\tilde{c} = [\alpha, \beta]$ de cette série de n intervalles est alors défini de manière à être aussi proche que possible des $\tilde{x}_i$. Pour cela, on remplace dans la définition de S_p donnée en (2), les distances $\mid x_i - u \mid$ par des distances $d(\tilde{x}_i, \tilde{u})$ entre les intervalles $\tilde{x}_i$ et $\tilde{u}$. La fonction correspondante permettant de mesurer cette proximité est alors :

$$\widetilde{S}_p(\tilde{u}) = \| \ (d(\tilde{x}_i, \tilde{u}))_{i=1,\ldots,n} \ \|_p \quad . \tag{3}$$

L'intervalle central $\tilde{c} = [\alpha, \beta]$ est alors

$$\tilde{c} = \arg\min_{\tilde{u} \in I} \widetilde{S}_p(\tilde{u}), \tag{4}$$

où I est l'ensemble des intervalles de $\mathbb{R} \times \mathbb{R}$, et la mesure de dispersion associée est $\widetilde{S}_p(\tilde{c})$.

Dans cette communication, nous étudions ce problème d'optimisation pour $p = 1, 2, \infty$ en choisissant pour d la distance de Hausdorff. Cette distance possède en effet la propriété intéressante d'être à la fois :

- une distance entre ensembles,
- égale à la distance L_∞ entre les vecteurs $(a_1, b_1)^t$ et $(a_2, b_2)^t$ des bornes inférieures et supérieures des intervalles $\tilde{\mathbf{x}}_1$ et $\tilde{\mathbf{x}}_2$,
- égale à la distance L_1 entre les vecteurs $(m_1, l_1)^t$ et $(m_2, l_2)^t$ des milieux et des demi-longueurs de $\tilde{\mathbf{x}}_1$ et $\tilde{\mathbf{x}}_2$.

Pour $p = 1$ et $p = \infty$, il existe une forme explicite d'une solution du problème de minimisation (4).

- Pour $p = 1$, le milieu $\hat{\mu}$ et la demi longueur $\hat{\lambda}$ de l'intervalle central $\tilde{c}$ sont définis par :

$$\begin{array}{rcl} \hat{\mu} & = & \text{médiane } \{m_i \mid i = 1, \ldots, n\}, \\ \hat{\lambda} & = & \text{médiane } \{l_i \mid i = 1, \ldots, n\}. \end{array} \tag{5}$$

  La mesure de dispersion associée est $\widetilde{S}_1(\tilde{c}) = \sum_{i=1}^n |m_i - \hat{\mu}| + \sum_{i=1}^n |l_i - \hat{\lambda}|$.

- Pour $p = \infty$, la borne inférieure $\hat{\alpha}$ et la borne supérieure $\hat{\beta}$ de l'intervalle central $\tilde{c}$ sont définies par :

$$\begin{array}{rl} \hat{\alpha} = & \dfrac{a_{(n)} - a_{(1)}}{2}, \\ \hat{\beta} = & \dfrac{b_{(n)} - b_{(1)}}{2}, \end{array} \tag{6}$$

  où $a_{(n)}$ (resp. $b_{(n)}$) est la plus grande borne inférieure (resp. supérieure) et $a_{(1)}$ (resp. $b_{(1)}$) est la plus petite borne inférieure (resp. supérieure). La mesure de dispersion associée est $\widetilde{S}_\infty(\tilde{c}) = \max \left\{ \mid a_{(n)} - \hat{\alpha} \mid, \mid b_{(n)} - \hat{\beta} \mid \right\}$.

Ces résultats ont déjà été démontrés et utilisés pour le calcul des prototypes optimaux en classification des Nuées Dynamiques, voir par exemple [6], [7], [8]. Ils ont également été utilisés pour définir une distance normalisée entre vecteurs d'intervalles en adéquation avec le prototype optimal choisi et ce afin de pouvoir tenir compte des échelles de mesures très différentes. Ce problème du centrage-réduction des variables décrites par des intervalles en classification a également été abordé par [5].

Les intervalles centraux définis en (5) et (6) peuvent être vus comme une généralisation (dans le cadre d'une série d'intervalles et pour la distance de Hausdorff) de la médiane et du midrange vus dans le cadre d'une série univariée. Pour $p = 2$, on aimerait donc aussi trouver le pendant de la moyenne et de la variance. Cependant la résolution du problème de minimisation est dans ce cas plus difficile : aucune formule explicite de la solution ne semble pouvoir être obtenue. L'objet de cette communication est de montrer comment pour $p = 2$ un intervalle central peut être calculé en un nombre d'opérations fini proportionnel à n^3 ainsi que l'algorithme permettant ce calcul. Pour plus de détails, nous renvoyons le lecteur à [3].

[1] Bertrand, P. and Goupil, F. (2000), "Descriptive statistics for symbolic data", In: H.-H. Bock and E. Diday (eds.): *Analysis of symbolic data. Exploratory methods for extracting statistical information from complex data*, Springer Verlag, 103-124.

[2] Billard, L. and Diday, E. (2003), "From the statistics of data to the statistics of knowledge: Symbolic data analysis", *Journal of the American Statistical Association*, 98, 470-487.

[3] Chavent, M. et Saracco, J. (2007), "On central tendency and dispersion measures for intervals and hypercubes", soumis à *Communications in Statistics - Theory and Methods*.

[4] Chavent, M. (2005), "Sur la normalisation pour le classification de données intervalles", *13èmes rencontres de la Société Francophone de Classification (SFC05)*, Montréal, 100-103.

[5] Chavent, M. (2005), "Normalized k-means clustering of hyper-rectangles", *XIth International Symposium of Applied Stochastics Models and Data Analysis (ASMDA05)*, Brest, 670-677.

[6] Chavent, M. (2004), "An Hausdorff distance between hyper-rectangles for clustering interval data", In: D. Banks et al. (eds.): *Classification, Clustering and Data Mining Applications*, Springer, 333-340.

[7] Chavent, M. and Lechevallier Y. (2002), "Dynamical Clustering of interval data. Optimization of an adequacy criterion based on Hausdorff distance", In: K. Jaguga et al. (Eds.): *Classification, Clustering and Data Analysis*, Springer Verlag, 53-60.

[8] Chavent, M., Carvhalo, F. de A.T., Lechevallier, Y. and Verde R. (2006), "New Clustering methods for interval data", *Computational Statistics*, 26, 211-229.

[9] De Carvhalo, F. de A.T., Brito, P. and Bock, H.-H. (2006), "Dynamic Clustering for Interval Data Based on L_2 Distance", *Computational Statistics*, 21, 231-250.

Exploitation des données d'enquêtes avec des méthodes d'analyse de données symboliques

M.C Rahal[1], F. Afonso[1], M. Touati[1] E. Diday[1], A. Peradotto[2], Y. Quatrain[2], S. Nugier [2], M. Hurault-Plantet [3]

1. CEREMADE - Paris Dauphine, Place du Ml de Lattre de Tassigny 75775 Paris Cedex 16
2. EDF R&D, 1 avenue du général de Gaulle 92141 Clamart cedex
3. LIMSI - CNRS B.P. 133 91403 ORSAY CEDEX FRANCE
(rahal, afonso, touati, diday)@ceremade.dauphine.fr,
(anne.peradotto, yasmina.quatrain,sylvaine.nugier)@edf.fr , mhp@limsi.fr

Mots clés : Analyse de Données Symboliques, Données textuelles

1 Introduction

Les besoins des entreprises pour mieux valoriser leurs données, dont la collecte et le stockage coûtent cher, sont de plus en plus pressants. L'exploitation des données est un véritable enjeu stratégique et peut apporter un avantage concurrentiel majeur. Toutefois, les données à appréhender pour embrasser un domaine sont de plus en plus nombreuses et complexes. Les analyses et les modèles qui en sont issus ou qui cherchent à décrire la complexité des systèmes marchands ou de relations sociales, sont de plus en plus pointus et sophistiqués. De plus, l'exploitation des résultats elle-même devient une tâche délicate et complexe. Le constat est que les outils permettant de rendre assimilable et exploitable de grands volumes de données, des analyses ou des résultats complexes par des utilisateurs métier de profils différents font défaut. Dans ce cadre un projet ANR, nommé SEVEN, a été monté en 2006 avec plusieurs partenaires dont EDF, le LIMSI et le CEREMADE pour la partie analyse des données. Son objectif est de proposer un outil de visualisation de classifications d'informations symboliques afin de permettre un accès intuitif et interactif à de grandes masses de données complexes ou à des résultats d'analyses. C'est dans le cadre de ce projet que les travaux décrits ci-après ont été menés.

2 Le problème et les données

EDF, au même titre que les entreprises de production ou de service, est intéressée par l'étude de la satisfaction et de la fidélité de ses clients. Pour cela, des enquêtes sont effectuées. Cet article présente une analyse innovante des données issues de 4 vagues d'enquête comprenant à la fois des questions fermées et des questions ouvertes. L'analyse porte sur environ 5 000 réponses. L'objectif de cette analyse est de décrire le lien entre d'une part des variables descriptives des clients (zone géographique, niveau de consommation, etc.) et leurs réponses aux questions fermées et d'autre part le texte libre correspondant à trois questions ouvertes sur les raisons de la satisfaction ou du mécontentement, sur les raisons de l'intention de fidélité ou de départ et enfin une question sur des remarques supplémentaires éventuelles.

3 Classification automatique des clients à partir de leurs réponses aux questions ouvertes

Une classification automatique des clients à partir des mots de leur réponse libre a été effectuée par la méthode de classification non supervisée (Jardino (2004)) qui est basée sur une méthode automatique pour extraire une structure, si elle existe, dans des textes ou des ensembles de textes. Cette structure prend la forme d'un arbre de profondeurs et de ramifications variables qui sont déterminées par les données. Pour cela nous réalisons des partitions successives quasi-optimales des textes à l'aide d'un algorithme de classification non supervisée de type " centres de gravité mobiles ". Pour chacune des trois questions ouvertes, on a obtenu 4 classes de réponses notées C1, C2, C3, C4 (représentant chacune les " thèmes " différents abordés dans les réponses). Ces classes ont été libellées par les experts de l'EDF à partir des mots les plus discriminants. Ces trois classifications ont ainsi permis de définir trois nouvelles variables nominales caractérisant les clients. A partir de ces résultats, un tableau **T** est construit croisant clients (environ 5000 individus) et variables caractéristiques (descriptives, réponses aux questions fermées, nos trois nouvelles variables issues de la classification des données textuelles). Les variables de **T** sont de type numériques ou nominales et sont une trentaine.

4 Extraction de règles liant les variables descriptives et les variables textuelles

Afin de faire des liens entre les variables descriptives et le thème abordé dans les questions ouvertes nous avons appliqué la méthode SApriori (Afonso (2005)). Cette méthode permet d'extraire des règles d'association entre les valeurs des variables. Nous appliquons cette méthode au tableau **T** précédemment décrit. Nous obtenons ainsi un grand nombre de règles du type (exemple fictif) : $R\acute{e}gion = Ile{-}de{-}France \wedge Type = locataire \wedge Vague = 2005 \rightarrow Classe{-}fidelit\acute{e} = prix$. Cette règle a un support de 1,8% et une confiance de 53%. Cette règle d'association se lit : "les clients de la région Ile-de-France locataires, interrogés en 2005 sont, dans 53% des cas, préoccupés par le prix". Nous avons sélectionné les règles avec uniquement les modalités des trois variables textuelles en conclusion, pour un support minimum entre 0,5% et 1% et une confiance supérieure au support de la conclusion de la règle.

5 Concepts issus des classifications par SCLUST

Dans le but d'étudier le lien entre des classes de clients et les thèmes abordés dans les réponses aux questions ouvertes, nous avons commencé par classifier les clients selon les variables descriptives et les réponses aux questions fermées (classification automatique effectuée à l'aide de la méthode SCLUST du logiciel publique SODAS,(Bock et Diday (2000)). Nous avons ainsi obtenu 16 classes types de clients. Une interprétation (caractérisation) de ces classes sur l'ensemble des variables, y compris les labels représentant les thèmes des questions ouvertes (Sur la Figure 1 les modalités caractéristiques sont classées par le range [1]), a ensuite été effectuée à l'aide du programme STAT amélioré de SODAS.

[1] range est l'écart entre la valeur minimal et maximale que prend la modalité sur tout les objets symboliques présents dans le tableau à traiter.

CATEGORIES WHERE INDIVIDUAL 7 IS THE HIGHEST

proba/mean	proba	category	variable	opposite individual	range
3.435911	0.899371	1	vague	13	0.831271
2.967727	0.582809	NR	sa_suivi	3	0.497988
2.438132	0.427673	tarif - consommation	Divers	13	0.335678
3.491417	0.251572	NR	fi_5pc	11	0.244905
1.373186	0.425577	ARGENT	segment_client	6	0.168001
1.194556	0.450734	service_prix	Fidelite	4	0.154438
1.296363	0.416667	O	gaz	6	0.152299
1.177148	0.601677	Plutot OK	Image	11	0.135010
2.288637	0.171908	NR	sa_tarifs	6	0.134029
2.223413	0.155136	Moy OK	sa_conseil	16	0.118342
1.597803	0.176101	Plutot OK	sa_conseil	4	0.108761
1.176820	0.358491	Region Parisienne	region_site	6	0.100915
1.804786	0.125000	Plutot OK	sa_services	11	0.096223
2.618123	0.079665	NSP	naf_17	9	0.073086
1.029014	0.958071	NR	sa_courrier	4	0.059081
2.697660	0.018868	NR	fi_globale	3	0.018868
2.980629	0.016772	NR	sa_globale	16	0.016772
16.000000	0.004193	NR	Image	2	0.004193

Range ends: highest / lowest — List mode: full / ranges over 0.0 — Sort by: proba/mean / proba / categ. label / variable label / range

FIG. 1 – STAT : caractérisation de la classe 7/16

La Figure 1 présente un exemple de sortie pour une classe donnée (classe 7). Cette analyse permet de décrire chaque classe en terme de profil client et de discours caractéristiques. On voit par exemple que la classe 7 est formée majoritairement (90%) de questionnaires provenant de la vague 1 ce qui constitue une proportion de plus de trois fois supérieure à la population (proba/mean[2] = 3, 43), plus d'un tiers provenant de la région parisienne (36%). En ce qui concerne les variables textuelles transformées en Classes-Mots (EDF-LIMSI), on peut remarquer que c'est la modalité " tarif-consommation" de la variable divers qui est caractéristique de cette classe (proba/mean = 2, 44). Une visualisation de cette même classe et de deux autres classes (13 et 6) qui lui sont opposées sur plusieurs modalités (voir la colonne opposit individual Figure 1) a été effectuée utilisant le programme VIEW de SODAS.

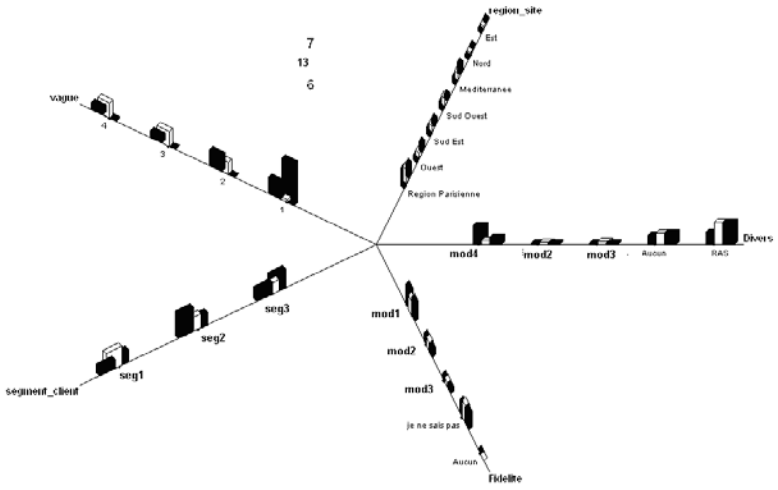


FIG. 2 – VIEW : Visualisation des classes 7, 13 et 6 /16

[2]la fréquence d'une modalité d'une classe divisée par la fréquence de la population pour cette modalité

6 Classification pyramidale avec interprétation

Une classification pyramidale a été effectuée ensuite sur les 16 prototypes "centre de gravité symbolique" issus de SCLUST. Dans la Figure 3 nous illustrons les classes importantes de la pyramide qui sont détectées grâce à un module de sélection basé sur les distances (Pak et al(2005)), la sélection est basée aussi bien sur les sauts externes (i.e. par rapport aux plus bas pères) que sur les sauts internes (ceux des paliers fils). Une interprétation de ces classes est effectuée à l'aide du programme STAT amélioré, qui offre une interprétation similaire à celle de la Figure 1, par exemple pour la classe 58 : Il s'agit de questionnaires de la vague 200 qui se préoccupe essentiellement du prix et de la consommation. Ce sont des petites entreprises du segment 2.

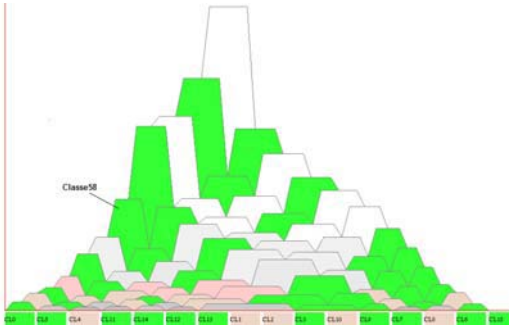


FIG. 3 – Pyramide symbolique élaguée

7 Conclusion

L'objectif de cette étude était d'étudier la faisabilité et surtout l'intérêt d'analyser conjointement des données structurées et textuelles dans le cadre de données d'enquête, avec les méthodes symboliques. Les résultats montrent qu'il existe un lien entre les variables descriptives et les réponses aux questions ouvertes et proposent de nouvelles pistes d'interprétation de ces liens.

Références

[1] F. Afonso, "Méthodes prédictives par extraction de règles en présence de données symboliques", *Thèse doctorale*,Paris Dauphine, France, 2005.

[2] L. Billard, E. Diday : Symbolic Data Analysis : conceptual statistics and data mining. (2006) Wiley. ISBN 0-470-09016-2.

[3] H.-H. Bock, E. Diday : "Analysis of Symbolic Data. Exploratory methods for extracting statistical information from complex data". Springer Verlag, Heidelberg, ISBN 3-540-66619-2. (Seconde edition).

[4] M. Jardino. Recherches de structures latentes dans des partitions de "textes" de 2 à K classes, JADT 2004, Louvain-La Neuve, Belgique, mars 2004.

[5] K. -K Pak, M. -C Rahal, E. Diday : Elagage et aide à l'interprétation symbolique et graphique d'une pyramide. EGC 2005 : 135-146.

Classification recouvrante avec pondération locale des attributs

Guillaume Cleuziou

LIFO, Université d'Orléans, rue Léonard de Vinci, 45067 Orléans cedex 2
guillaume.cleuziou@univ-orleans.fr

Mots clés : Classification automatique.

1 Introduction

La classification recouvrante consiste à extraire, à partir d'un ensemble d'individus, une collection de classes d'individus similaires constituant un recouvrement de cet ensemble. On parle également de pseudo-partition par opposition aux partitions traditionnellement recherchées en classification. Ce type particulier de classification est étudié de façon plutôt marginale depuis les années 70. Plusieurs approches ont alors été proposées, nous citerons en particulier les pseudo-hiérarchies et les méthodes fondées sur l'extension de partitions (e.g. l'algorithme des k-moyennes axial). Depuis quelques années, on observe un regain d'intérêt pour la classification recouvrante notamment dans la perspective de traiter des données textuelles (mots ou documents) ou biologiques (gènes) pour lesquelles ce type de schéma est particulièrement approprié. Cette nouvelle vague d'études a donné lieu à de nouveaux algorithmes tels que CBC ou POBOC, principalement fondés sur un raisonnement heuristique et intuitif; d'autres approches plus récentes visent à généraliser les méthodes traditionnelles de partitionnement (OKM [3] comme extension des k-moyennes) ou de modèles de mélanges (MOC [1] comme extension de EM) et héritent ainsi de formalisations plus solides.

Dans cet article, nous proposons d'étudier l'intérêt d'une pondération locale des attributs dans le cadre de la classification recouvrante. La pondération locale consiste à associer à chaque classe (localement) une pondération des attributs utilisés pour décrire les individus. Cette pondération locale permet de caractériser les classes en indiquant a posteriori les attributs ayant fortement contribué à leur construction; de plus la pondération joue un rôle actif dans la construction même de ces classes puisque l'appartenance d'un individu à une classe sera évaluée sur la base de critères certes identiques mais d'influences différentes selon la classe considérée. Dans le contexte de la classification recouvrante, la pondération locale des attributs apparaît appropriée en permettant notamment d'attribuer un même individu à plusieurs classes en construction sans induire un rapprochement de ces classes. Par exemple dans le cas d'une classification de documents, un même document traitant de deux thématiques distinctes chacune identifiée par un ensemble d'attributs (par exemple des mots) pourra appartenir à deux classes sur la base d'attributs différents sans que cette double appartenance ne dégrade la cohérence de ces classes.

Avant d'introduire la notion de pondération locale des attributs, nous débuterons la section suivante en rappelant le principe général de l'algorithme OKM dont nous présenterons ensuite une extension avec pondération. La dernière section sera consacrée à la présentation et à l'analyse d'une expérimentation préliminaire de cette approche.

2 Pondération locale dans OKM

L'approche OKM (*Overlapping k-means)* présentée dans [3] peut être vue comme une généralisation de l'approche bien connue des k-moyennes. En effet OKM consiste

à approcher dans l'espace de tous les recouvrements possibles (bien plus grand que l'espace des partitions), une solution qui optimise un critère plus général que le critère des moindres carrés, par itérations de deux étapes : l'affectation des individus à une ou plusieurs classes et le calcul des centres de classes. Étant donné un ensemble de n individus $\mathcal{X} = \{X_1, \ldots, X_n\}$ décrits dans $\mathbb{R}^m$ muni d'une métrique euclidienne d, ce critère objectif est défini par :

$$W(\mathcal{R}) = \sum_{j=1}^{n} d^2(X_j, \overline{X}_j) \tag{1}$$

Dans (1), $\mathcal{R}$ désigne une collection de classes $\{\mathcal{R}_1, \ldots, \mathcal{R}_k\}$ formant un recouvrement de $\mathcal{X}$ et $\overline{X}_j$ l'image de X_j dans $\mathcal{R}$, définie comme le centre de gravité des centres de classes auxquels X_j appartient. Ainsi définie, $d^2(X_j, \overline{X}_j)$ peut être interprétée comme l'erreur commise en résumant l'individu X_j à l'ensemble des centres de ses classes d'appartenance. On note que dans le cas où chaque individu n'est affecté qu'à une seule classe, on se ramène exactement au critère des moindres carrés.

Afin d'introduire la pondération locale dans le formalisme de l'approche Okm, nous effectuons un bref rappel sur cette notion de pondération locale proposée par [2] dans le cadre de l'algorithme des k-moyennes. Il s'agit dans cette approche d'optimiser le critère objectif suivant :

$$F(W, Z, \Lambda) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{l,j} \lambda_{l,i}^{\beta} d^2(z_{l,i}, x_{j,i}) \tag{2}$$

où Z désigne l'ensemble $\{Z_1, \ldots, Z_k\}$ des centres des classes respectives $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_k\}$, W est la matrice binaire $k \times n$ des appartenances (avec $\sum_{l=1}^{k} w_{l,j} = 1$ pour $1 \leq j \leq n$), Λ la matrice $k \times m$ des poids dans [0,1] (avec $\sum_{i=1}^{m} \lambda_{l,i} = 1$ pour $1 \leq l \leq k$), $z_{l,i}$ (ou $x_{j,i}$) la $i^{\text{ème}}$ composante de Z_l (respectivement X_j) et β un exposant réel supérieur à 1. L'optimisation du critère F est alors réalisée de façon heuristique par l'itération de trois étapes, chacune consistant à minimiser F selon l'un de ses paramètres en fixant les deux autres :

1. Optimiser F selon W revient à trouver une affectation optimale de chaque individu à une classe pour Z et Λ fixés ; l'optimum est atteint pour l'affectation de chaque individu au centre le plus proche (pour X_j, $Z^* = \text{argmin}_{Z_l \in Z} \sum_{i=1}^{m} \lambda_{l,i}^{\beta} d^2(z_{l,i}, x_{j,i})$).
2. La seconde étape vise à calculer les nouveaux centres de classes Z qui minimisent F pour W et Λ fixés ; on peut alors montrer que ces centres correspondent simplement aux centres de gravité des objets de la classe.
3. Enfin, la dernière étape consiste à rechercher les pondérations Λ optimales pour W et Z fixés dans F ; Chan *et al.* montrent alors que pour chaque cluster $\mathcal{Z}_l$ indépendamment, le poids optimal $\lambda_{l,i}$ sur la composante i s'obtient en fonction de l'inverse de l'inertie des individus de la classe sur cette composante (cf. théorème 3 dans [2]). De cette façon, une composante sera d'autant plus fortement pondérée que la variance des individus sur cette composante est faible.

L'introduction de la pondération locale pour rechercher un "bon" recouvrement nécessite d'une part de redéfinir le critère objectif d'un recouvrement avec pondération et donc de la notion d'image d'un individu dans un tel environnement, et d'autre part d'en déduire une adaptation de l'heuristique visant à minimiser ce critère (algorithme).

De même que le critère objectif (1) utilisé dans Okm généralise le critère des moindres carrés, nous définissons un nouveau critère (3) généralisant $F(.)$:

$$G(W,R,\Lambda) = \sum_{j=1}^{n}\sum_{i=1}^{m} \gamma_{j,i}^{\beta} d^2(x_{j,i},\overline{x}_{j,i}) \qquad (3)$$

Dans cette expression, les paramètres W, R et Λ désignant respectivement les appartenances des individus aux classes, les centres et les pondérations locales des classes, n'apparaissent pas explicitement mais définissent totalement la notion d'image dans ce contexte ainsi que les nouveaux poids Γ (utilisés ici pour ne pas alourdir les notations) :

$$\overline{x}_{j,i} = \frac{1}{\sum_{l=1}^{k} w_{l,j}\lambda_{l,i}^{\beta}} \sum_{l=1}^{k} w_{l,j}\lambda_{l,i}^{\beta} r_{l,i} \qquad \text{et} \qquad \gamma_{j,i} = \frac{1}{\sum_{i=1}^{m}\sum_{l=1}^{k} w_{l,j}\lambda_{l,i}} \sum_{l=1}^{k} w_{l,j}\lambda_{l,i}$$

L'image $\overline{X}_j$ d'un individu X_j dans un recouvrement avec pondération locale correspond à un centre de gravité pondéré des centres des classes auxquelles X_j appartient. Les poids $\{\gamma_{j,i}\}_{j=1,\ldots,n}$ peuvent être interprétés comme une pondération locale de l'image $\overline{X}_j$, obtenue par une moyenne (pondérée) des poids $\lambda_{l,i}$ des classes d'appartenance de X_j ($\forall j,\ \sum_{i=1}^{m} \gamma_{j,i} = 1$).

En complément de cette nouvelle formalisation, nous proposons une heuristique qui consiste, étant donné un ensemble initial arbitraire de k centres (k fixé), à générer itérativement des solutions (recouvrements) qui améliorent le critère (3). Chaque itération comporte trois étapes, chacune d'elle visant à mettre à jour l'un des paramètres de façon à faire décroître le critère :

1. Mise à jour de W : affecter chaque individu X_j à ses plus proches centres (au sens de $d(X_j,R_l)$) tant que l'erreur commise diminue ($\sum_{i=1}^{m} \gamma_{j,i}^{\beta} d^2(x_{j,i},\overline{x}_{j,i})$ décroît); la nouvelle affectation étant finalement conservée si et seulement si elle améliore la précédente (cf. algorithme Okm [3]).
2. Mise à jour de R : calculer pour chacune des classes un nouveau centre R_l, centre de gravité du nuage de points $\{(X_j,P_j)|X_j \in \mathcal{R}_l\}$, correspondant à l'ensemble des individus de la classe, pondérés par $p_{j,i} = \frac{(\gamma_{j,i}.\lambda_{l,i})^{\beta}}{\alpha_{j,i}^2}$ avec $\alpha_{j,i} = \sum_{l=1}^{k} w_{l,j}\lambda_{l,i}^{\beta}$. On peut montrer que ce choix et celui qui minimise le critère (3) selon R_l.
3. Mise à jour de Λ : la nouvelle pondération locale de chaque classe sera calculée pour chaque composante, relativement à la variance sur cette même composante des individus qui la compose [a]:

$$\lambda_{l,i} = \frac{\left(\sum_{j=1}^{n} w_{l,j} p_{j,i} (r_{l,i} - \hat{x}_{j,i}^{l})^2\right)^{1/(1-\beta)}}{\sum_{t=1}^{m}\left(\sum_{j=1}^{n} w_{l,j} p_{j,t} (r_{l,t} - \hat{x}_{j,t}^{l})^2\right)^{1/(1-\beta)}}$$

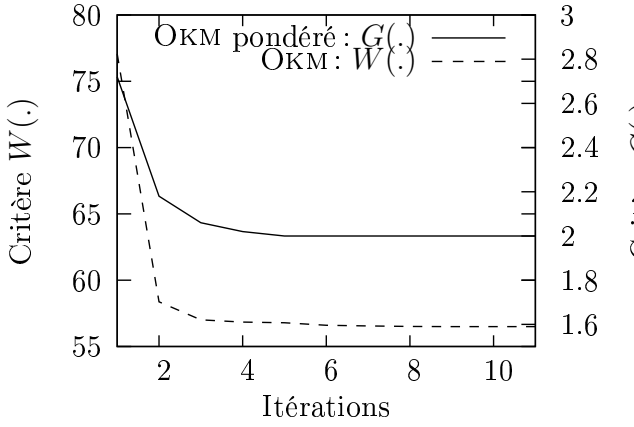
où $\hat{X}_j^l$ désigne le centre Z_l "idéal" pour X_j (i.e. tel que $d(X_j,\overline{X}_j) = 0$).
La nouvelle pondération n'étant retenue que si elle améliore le critère (3).

[a] Voir le théorème 3 dans [2] dans le cas d'un dénominateur nul.

Il est important de remarquer que si l'on autorise chaque individu à n'appartenir qu'à une seule classe (partitionnement strict), l'algorithme présenté est identique à celui proposé par Chan *et al.* [2] .

3 Résultats préliminaires et conclusion

Nous présentons une expérimentation préliminaire de l'approche proposée sur la base usuelle des Iris (150 individus dans $\mathbb{R}^4$). Les résultats exposés rapportent l'analyse d'une exécution[1] de Okm et de sa version pondérée dans des conditions initiales identiques.



Classes	1	2	3
Setosa	50	1	
	(50)	*(0)*	
Versicolour	1	50	20
	(4)	*(50)*	*(21)*
Virginica	0	22	50
	(0)	*(25)*	*(49)*

Fig. 2 – *Matrice de confusion pour* Okm *(valeurs entre parenthèses) et sa version pondérée.*

Fig. 1 – *Vitesse de convergence des algorithmes.*

La figure 1 montre une convergence aussi rapide pour les deux versions (pondérée ou non) de Okm. D'autre part, la matrice de confusion présentée en Figure 2 révèle une amélioration sensible de la qualité des classes en introduisant la pondération. On observe par exemple, que la classe "setosa", réputée pour être facile à distinguer, est totalement identifiée dans les deux versions de Okm (classe 1) mais avec une intersection plus réduite dans la variante pondérée.

Nous avons présenté dans cet article, une formalisation possible du problème de recherche d'un "bon" recouvrement en utilisant la pondération locale des attributs. L'intuition initiale, selon laquelle l'introduction de pondérations serait appropriée au contexte de la classification recouvrante est alors encouragée par les résultats préliminaires obtenus. En effet, cela permet lors de la construction des classes, d'affecter un même individu à plusieurs classes sans induire un rapprochement de ses classes; ce qui se manifeste par une limitation des recouvrements indésirables. Les perspectives de ce travail seront de confirmer les résultats observés en proposant d'autres expérimentations sur des domaines cibles tels que la classification de documents, d'étudier l'influence du paramètre β et de comparer cette approche avec d'autres méthodes telles que Moc, PoBoc et la version floue de l'algorithme des k-moyennes complété par une étape d'affectation seuillée.

Références

[1] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. Model-based overlapping clustering. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 532–537, New York, NY, USA, 2005. ACM Press.

[2] E. Y. Chan, W.-K. Ching, M. K. Ng, and J. Z. Huang. An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, 37(5):943–952, 2004.

[3] G. Cleuziou. Okm : une extension des k-moyennes pour la recherche de classes recouvrantes. In *Journées Francophones d' Extraction et de Gestion des Connaissances EGC'2007*, volume 2, Namur, Belgique, janvier 2007. Revue des Nouvelles Technologies de l'Information, Cépaduès-Edition.

1. La meilleure obtenue au sens du critère (3) sur 50 exécutions, avec la distance euclidienne et $\beta = 2$.

Indices de similarité structurelle sur des données arborescentes

N. Conruyt[1] , D. Grosser[1], H. Ralambondrainy[1]

1. IREMIA, Université de la Réunion
15 av Cassin, 97715 Saint-Denis Message Cedex 9, Réunion
(conruyt, grosser,ralambon)@univ-reunion.fr

Mots clés : Approches inspirées du vivant, Arbres, graphes et classification, Applications

1 Introduction

Les descriptions issues de l'observation de spécimens biologiques sont particulièrement complexes. Dans le cadre d'un projet intitulé "base de connaissances sur les coraux des Mascareignes", nous développons de nouvelles méthodes de représentation et d'analyse de descriptions morphologiques arborescentes, pour permettre d'une part aux experts du domaine d'exprimer au mieux leurs connaissances du domaine, et d'autre part pouvoir fournir une aide à l'identification et à la classification. Ces méthodes sont intégrées au sein d'une plate-forme logicielle, IKBS [1]. Les connaissances relatives au domaine ainsi que le schéma des descriptions sont exprimés par une entité appelé "modèle descriptif". Pour comparer les descriptions, il est nécessaire de disposer de méthodes de calcul de mesures de similarité appropriées qui prennent en compte les connaissances relatives au domaine et en particulier leur structure. Nous présentons ici une méthode visant à caractériser finement les différences structurelles existant entre deux descriptions.

2 Indices de similarité structurelle

2.1 Le modèle descriptif

Le modèle descriptif des coraux de la famille des *Pocilloporidae* est composé d'attributs "identification", "contexte" qui sont des attributs classiques prenant des valeurs dans un domaine donné et de l'attribut *description* qui est construit à l'aide de l'attribut "squelette" composé de partie inférieure "face aborale" et partie inférieure "face orale". Pour représenter un tel modèle, on considère un ensemble d'attributs dits "simples" $\{(A_q, D_q)_{q\in Q}\}$ d'identificateur A_q et de domaine D_q de nature quelconque (ensemble fini ou de réels, d'intervalles, etc). On appelle attribut structuré, une séquence $A :< A_1, \ldots, A_l, , \ldots, A_p >$ où A_l est un attribut simple ou structuré, la définition d'un attribut structuré est récursif. On dira que l'attribut A_l est un composant de l'attribut A. Un attribut structuré est utilisé pour représenter les différents composants d'une observation. Un modèle descriptif est défini par la donnée d'un attribut structuré A. Notons $\mathcal{A} = \{A_j\}_{j\in J}$ l'ensemble des attributs simples ou structurés entrant dans la définition du modèle descriptif A. Un attribut structuré A est représenté par une arborescence $\mathcal{M} = (\mathcal{A},\mathcal{U})$. L'ensemble des sommets $\mathcal{A}$ est tel que les noeuds sont les attributs structurés, les feuilles les attributs simples, et une arête $(B, B') \in \mathcal{U}$ exprime que l'un des deux attributs est un attribut composant de l'autre. Une observation ou un cas (figure 1) est une arborescence dérivée de celle de A dans lequel les attributs simples ont été valorisés par une valeur de leur domaine.

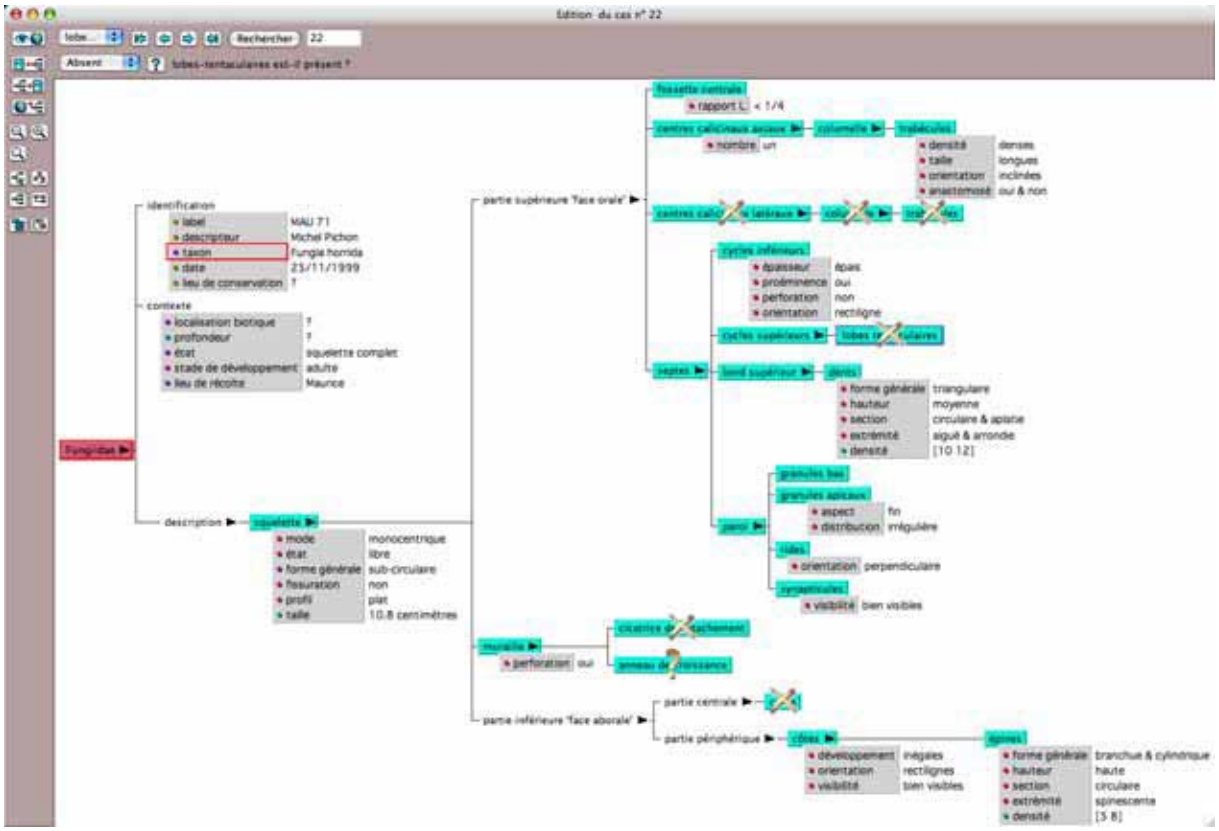


Figure 1: Un spécimen de corail de la famille *Pocilloporidae* : cas 1

Table 1: La mesure de comparaison λ_0 pour un noeud donné

$H_{\sigma_1} \backslash H_{\sigma_2}$	$+$	$-$	$*$
$+$	1	0	?
$-$	0	1	?
$*$	?	?	?

2.2 Squelette d'une observation arborescente

Un *squelette* (figure 2) décrit la structure morphologique d'un cas, il exprime la présence (+) ou l'absence (-) (l'absence d'un noeud est représentée par une croix sur la figure 1) ou l'état inconnu ($*$) d'une partie de sa description (? sur la figure 2). Plus précisément, si $S = \{+, -, *\}$, un squelette est une arborescence dans laquelle chaque noeud A_j du modèle descriptif A a été annoté par un élément de s de S. Une application $\sigma : \mathcal{A} \rightarrow S$ définit le squelette H_σ par l'arborescence $H_\sigma = (\mathcal{A}_\sigma, \mathcal{U})$ avec $\mathcal{A}_\sigma = \{(A_j, \sigma(A_j))_{j \in J}\}$.

2.3 Arborescence de comparaison

Pour pouvoir comparer deux squelettes $H_{\sigma_1}, H_{\sigma_2}$, on définit une fonction $\lambda : S \times S \rightarrow K$ qui mesure, par la valeur $\lambda(\sigma_1(A_j), \sigma_2(A_j))$, la présence ou l'absence de chaque noeud A_j dans les deux squelettes. Par exemple, $K = \{0, 1, '?'\}$ et le tableau de valeurs correspondant à λ_0 est donné dans la table 1, ou encore $K = \{0, 1\}$ avec la fonction λ_1 (table 2), $K = [0, 1]$ avec la fonction λ_2 (table 3) où $\alpha_j^1, \alpha_j^2, \beta_j^1, \beta_j^2, \gamma_j$ sont des estimations des valeurs manquantes que nous ne développons pas ici.

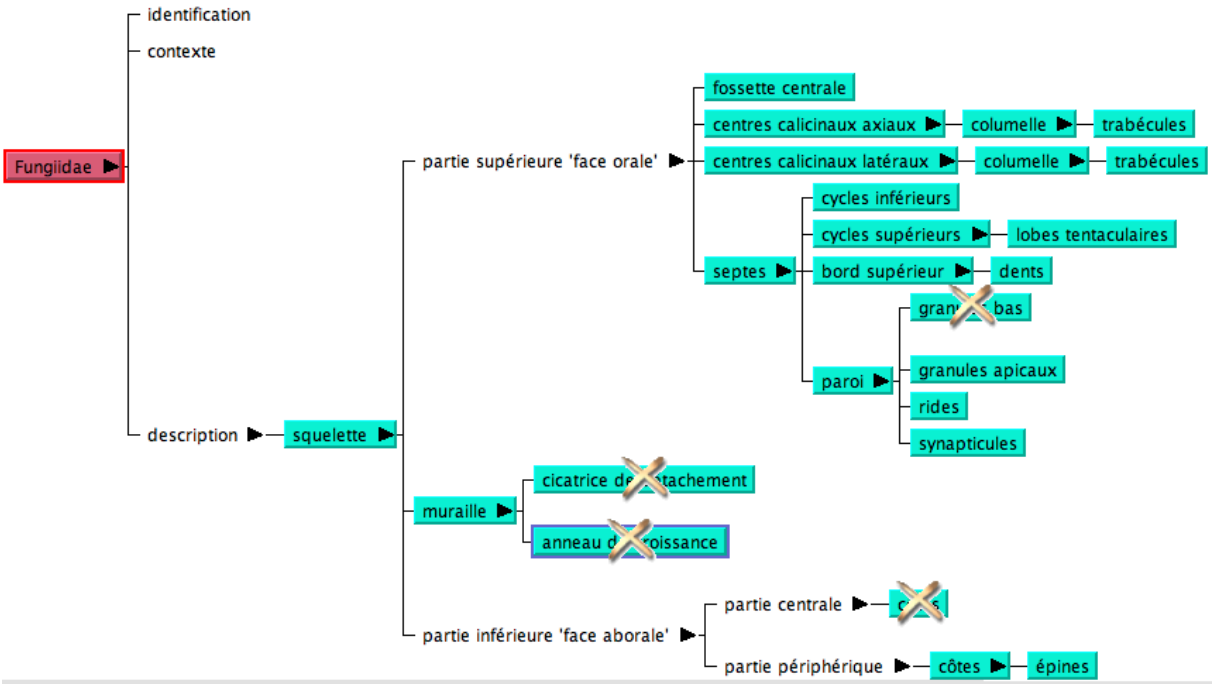


Figure 2: Squelette du cas 2

Table 2: La mesure de comparaison λ_1 pour un noeud donné

$H_{\sigma_1} \backslash H_{\sigma_2}$	$+$	$-$	$*$
$+$	1	0	0
$-$	0	1	0
$*$	0	0	0

On appelle arborescence de comparaison, une arborescence dans laquelle chaque noeud A_j du modèle descriptif A a été annoté par un élément de k de K. Une application $\kappa : \mathcal{A} \rightarrow K$ définit une arborescence de comparaison $H_\kappa = (\mathcal{A}_\kappa, \mathcal{U})$ avec $\mathcal{A}_\kappa = \{(A_j, \kappa(A_j))_{j \in J}\}$. Soient deux squelettes $H_{\sigma_1}, H_{\sigma_2}$, l'arborescence qui recense la comparaison des noeuds de ces deux squelettes est H_κ avec $\kappa(A_j) = \lambda(\sigma_1(A_j), \sigma_2(A_j))$. Comment synthétiser les similitudes et différences comptabilisées par l'arborescence H_κ ? Considérons la mesure de comparaison λ_0, et notons $J_= = \{j \in J | \kappa(A_j) = 1\}$, l'ensemble des indices des noeuds pour lesquelles on a la coprésence et $n_= = |J_=|$, $J_? = \{j \in J | \kappa(A_j) = '?'\}$ lorsque l'un des noeuds est inconnu, $n_? = |J_?|$. Le complémentaire $J_\neq$ de $J_=$ et $J_?$ dans J est l'ensemble d'indices relatifs aux "différents", on a $n_\neq = |J_\neq| = n - n_= - n_?$ où $n = |J|$. Pour synthétiser les valeurs de similitude entre deux squelettes, la première idée est d'utiliser des indices classiques comme celle de Sokal [3], $\iota_S = (H_{\sigma_1}, H_{\sigma_2}) = \frac{n_=}{n}$. ou sa variante, l'indice de

Table 3: La mesure de comparaison λ_2 pour un noeud A_j donné

$H_{\sigma_1} \backslash H_{\sigma_2}$	$+$	$-$	$*$
$+$	1	α_j^1	β_j^1
$-$	α_j^2	1	γ_j
$*$	β_j^2	γ_j	1

Estabrook [2] $\iota_E(H_{\sigma_1}, H_{\sigma_2}) = \frac{n_=}{n-n_?}$. Ces coefficients accordent un même poids égal à 1 aux noeuds dans le calcul de la similarité et ne prennent pas en compte la structure hiérarchique du squelette. Pour surmonter cet handicap, on définit une fonction poids $m : \mathcal{A} \to \mathcal{N}$. Kendrick propose que l'importance d'un noeud soit fonction de la taille du sous-arbre dont il est la racine, plus précisément si A_j est un attribut simple (une feuille) alors $m(B) = 1$ et si $B =< B_l >_{l \in L}$ est un attribut structuré alors $m(B) = 1 + \Sigma_{l \in L} m(B_l)$. Une autre possibilité est de considérer un index de filiation comme pondération, pour la racine $m(A) = 1$ et si B a pour père $pere(B)$ et comme ensemble de fils $Fils(B)$, on pose $m(B) = \frac{m(pere(B))}{|Fils(B)|}$. Pour une pondération m donnée, l'indice de Sokal s'écrit $\zeta_S(H_{\sigma_1}, H_{\sigma_2}) = \frac{\Sigma_{l \in J_=} m(A_l)}{\Sigma_{j \in J} m(A_j)}$. Pour généraliser ces indices, on considère l'ensemble $K = [0, 1]$ et une mesure de comparaison λ. La similitude structurelle de deux squelettes est évaluée par la moyenne des valeurs de similitude des noeuds de leur arborescence de comparaison:

$$\zeta(H_{\sigma_1}, H_{\sigma_2}) = \frac{\Sigma_{j \in J} m(A_j) \kappa(A_j)}{\Sigma_{j \in J} m(A_j)} = \frac{\Sigma_{j \in J} m(A_j) \lambda(\sigma_1(A_j), \sigma_2(A_j))}{\Sigma_{j \in J} m(A_j)}$$

Soit ζ_1 l'indice correspondant à la mesure de comparaison λ_1 :

$$\zeta_1(H_{\sigma_1}, H_{\sigma_2}) = \frac{\Sigma_{j \in J} m(A_l) \kappa_1(A_j)}{\Sigma_{j \in J} m(A_j)}.$$

On a $\kappa_1(A_j) = \lambda_1(\sigma_1(A_j), \sigma_2(A_j))$, compte tenu des valeurs que prend la fonction de comparaison λ_1, si $l \in J_=, \kappa_1(A_j) = \lambda_1(+, +) = \lambda_1(-, -) = 1$, et $\kappa_1(A_j) = 0$ dans toutes les autres situations, on a donc : $\Sigma_{j \in J} m(A_l) \kappa_1(A_j) = \Sigma_{j \in J_=} m(A_j)$. On remarque que $\zeta_S(H_{\sigma_1}, H_{\sigma_2}) = \zeta_1(H_{\sigma_1}, H_{\sigma_2})$, l'indice de Sokal correspond au choix de la fonction de comparaison λ_1, pour une pondération de tous les noeuds égale à 1.

Nous illustrons ci-dessous les calculs des indices de similarité relatifs aux squelettes des cas 1 et 2 (figures 1 et 2). On a $m(squelette) = 31$ et les différents effectifs $n = 29, n_{\neq} = 6, n_? = 1, n_= = 22$. Le calcul des indices classiques donne $\iota_E = 0,7857, \iota_S = 0,7586$. Les noeuds ayant été pondérés par l'importance de la taille de leurs sous-arbres associés, on a $\zeta_S = \zeta_1 = 0,8452$. La prise en compte de la structure d'arbre améliore la similitude. La mesure de similitude entre deux observations porte sur la structure mais aussi sur les valeurs des attributs simples (les contenus). Selon le type d'un attribut simple : réel, qualitatif, intervalle,etc différentes distances existent. La prochaine étape consistera à définir des stratégies ou des mesures de similitude qui combinent ces deux aspects.

[1] CONRUYT, N., GROSSER, D., RALAMBONDRAINY,H., (1997): IKBS: An Interative Knowledge Base System for improving description, Classification and identification of biological objects. *Proceedings of the Indo-French Workshop on Symbolic Data Analysis and its Applications 2, 212–224.*

[2] ESTABROOK, G.F., ROGERS D.J., (1966) : A general method of taxonomic description for a computed similarity measure, *Bioscience 16, 789–793.*

[3] SOKAL R.R., SNEATH P.H.A., (1963): Principles of numerical taxonomy. *W.H. Freeman et Cie; San Francisco et Londres.*

Une méthode de partition des images de lésions mélanocytes cutanées

Valentina Cozza[1], Mario R. Guarracino[2], Rosanna Verde[3]

1. Dip. di Matem. e Statistica, Univ. di Napoli 'Federico II', Via Cinthia, 80126 Napoli, Italie
2. ICAR-CNR, Via Pietro Castellino 111, 80131 Napoli, Italie
3. Dip. di Studi Europei e Mediterranei, Seconda Università di Napoli, Via del Setificio 15, San Leucio, 81100 Caserta, Italie
valentina.cozza@unina.it, mario.guarracino@na.icar.cnr.it, rosanna.verde@unina2.it

Mots clés : classification, images bio-médicales, descripteurs multi-valeurs

1 Introduction

Dans ce papier nous proposons d'utiliser, comme outil de diagnostic du mélanome malin, une méthode de classification de type Nuées Dynamiques ([6]) sur des images de lésions cutanées. Le mélanome malin est une forme de cancer de la peau qui se développe à partir des mélanocytes et qui a été observé en forte augmentation ces dernières années (5000-6000 cas par an en France seulement). Seuls le diagnostic précoce et l'intervention chirurgicale permettent d'envisager une guérison définitive. La différenciation entre le mélanome et les autres lésions pigmentées de la peau n'est pas triviale, même pour un dermatologue expérimenté. Cette problématique a naturellement attiré l'attention de nombreux chercheurs qui ont proposé différents systèmes pour la détection semi-automatique du mélanome (l'analyse discriminante, réseaux neuronaux, *Support Vector Machine*, ...) et pour l'individualisation des caractéristiques les plus discriminantes. Les procédures conventionnelles pour reconnaître la nature maligne d'une tâche pigmentée [9] sont :

i. la règle ABCD pour analyser: l'Asymétrie (A), le Bord (B), la Couleur (C), la Structure Différentielle (D) ;
ii. les méthodes d'analyse de patterns (globales et locaux) ;
iii. la méthode de Menzies ;
iv. la liste de contrôle de 7-points.

La première approche fournit une description des images des lésions par rapport à des mesures d'asymétrie calculées pour le bord, la couleur et les structures dermatoscopiques, à la présence des zones pigmentées qui s'interrompent brusquement au bord ou qui se dégradent lentement, au degré de la couleur des lésions et à l'existence d'un voile clair, aux mesures des différents composants de la structure (réseaux pigmentés, points, globules, zones sans structure et stries).
Les méthodes d'analyse de patterns se concentrent sur les caractéristiques de la structure de la lésion au niveau global et local. La méthode de Menzies focalise l'attention sur les principaux indicateurs de négativité et de positivité. La dernière méthode est basée sur une liste de contrôle à 7-points liés à des atypies dans le réseau pigmenté, à la présence du voile, à des zones vascularisées atypiques, à des stries, points, globules ou tâches irrégulières, à une régression dans la structure de la lésion.

En général, dans le problème d'identification automatique ou semi-automatique de forme du mélanome malin, les caractéristiques qui sont prises en compte font référence aux

quatre composantes de la règle ABCD : Asymétrie, Bord, Couleur, Structure Différentielle. Dans ce contexte, nous considérons deux indicateurs pour l'asymètrie de la forme et de la couleur; deux indicateurs pour l'irrégularité et le degré de minceur du bord, des indicateurs de la distribution des trois couleurs (Rouge, Vert et Bleu), de l'Intensité, de la Luminosité et du Contraste, des indicateurs pour l'irrégularité de la Structure Différentielle (présence/absence, localisation et couleur de pigments, de points, de globules et de stries). La structure complexe des images à analyser et l'incertitude souvent présente dans la mesure des caractéristiques ou des attributs de cette structure (qualité de la photo, subjectivité dans l'assignation d'un point, ...) s'expliquent par la multiplicité des manifestations dermatoscopiques. Après une segmentation des images, nous proposons de décrire chacune par des descripteurs ayant des valeurs multiples selon le modèle des données symboliques [2]. À partir de ce type de description des lésions, nous pouvons utiliser une méthode de classification de type Nuées Dynamiques pour l'identification semi-automatique de deux types de lésions : les mélanomes malins et les lésions bénignes. Les objectifs sont d'obtenir différentes classes de lésions malignes et bénignes en utilisant une information fournie par un expert et de focaliser l'attention sur les classes des deux ensembles qui présentent une forte similarité. L'idée est de découvrir les attributs des différentes caractéristiques qui induisent des erreurs d'attribution et qui sont les principales causes de diagnostics faussement positifs où incorrects. La procédure a été validée sur un jeux de données de 100 images de mélanomes et de lésions bénignes (échantillonnées à partir d'une base de 5380 images).

2 Description symbolique des images

Une description symbolique [2] d'une image de lésion est obtenue à partir d'un ensemble de descripteurs qui peuvent assumer des valeurs multiples (*multi-catégories*, *intervalles*, *modaux*). Nous choisissons un ensemble de descripteurs qui représentent des caractéristiques cliniques inclues dans la règle de l'ABCD. En particulier, l'asymétrie de la forme, de la structure et de la couleur par rapport aux deux axes principaux d'inertie. Car l'asymétrie de la structure réticulaire est plutôt une mesure liée à la régularité du réseau, elle sera prise en compte dans la description de la Structure Différentielle. L'asymétrie des différentes composantes est mesurée en termes d'intervalles de valeurs [*min, max*] sur les deux axes. Pour l'asymétrie de la forme d'une lésion m on considère la différence entre le nombre de pixels de l'image et le nombre de pixels concordants dans les quadrants de cette image.
Pour évaluer l'asymétrie de la couleur, nous considérons la partition de l'image en n cercles concentriques autour du centre des axes d'inertie. Le long des quatre directions des axes nous associons des histogrammes relatifs aux intensités de niveaux de gris. Soit I_{uv} l'intensité du niveau de gris présent dans le cercle v dans le demi-plan $u = d, g, i, s$, une mesure de l'asymétrie de la couleur par rapport aux deux axes est donnée par:

$$K_r = \sum_{v=1,n} (I_{u_1,v} - I_{u_2,n-v+1})^2; \quad u_r = \{d, g\}, \{i, s\} \text{ et } r = 1, 2$$

De la même façon que pour l'asymétrie de la forme, le descripteur de l'asymétrie de la couleur prend en compte les intervalles des valeurs minimales et maximales de K_1 et K_2.

Le Bord, ou contour est mesuré par le diamètre le plus grand, l'aire, l'irrégularité du bord, le rapport de minceur, l'index de circularité, la variance des distances des points sur le bord du barycentre. Considérant une partition de l'image de la lésion en 8 sections

radiales, avec des angles à 45°, on calcule la mesure d'inertie pour chacune des α configurations obtenues en faisant pivoter de θ° (par exemple $\theta = 5^\circ$) la configuration autour du centre et en prenant les valeurs minimales et maximales.

L'épaisseur du Bord identifie une interruption brusque de la lésion. On considére donc une bande de dimension fixée le long du bord et par rapport aux 8 sections radiales, on calcule le gradient d'intensité des niveaux de gris, du centre vers le bord dans la direction orthogonale au bord. De manière plus simple, la description du bord peut être exprimée par les valeurs minimales et maximales du gradient. Si on considére la distribution des gradients alors le descripteur sera de type modal.

Les caractéristiques de la Couleur se basent sur la distribution de la palette des trois couleurs (Rouge, Vert et Bleu) et sur les distributions des valeurs de l'Intensité, de la Luminosité et du Contraste. Nous utilisons des descripteurs de type *modal* qui représentent ces caractéristiques par la distribution correspondante de chaque image.

Dans la littérature, l'irrégularité de la Structure Différentielle est considérée comme un indicateur de haut risque des lésions malignes. Les descripteurs pris en considération sont les caractéristiques relatives aux aspects globaux : la présence d'un réseau pigmenté, de points, de globules et de stries ramifiées. Les points et les globules sont décrits par des attributs multi-catégoriques ordinaux comprenant leur *absence* ou *présence*, leur forme *circulaire* ou *ovale*, leur couleur *noire, gris* ou *marron* et leur localisation *centrale* ou *périphérique*. Les stries sont décrites en fonction de leur *régularité* ou *irrégularité*. Pour tous les descripteurs considérés, des valeurs élevées correspondent à une caractérisation maligne ou à haut risque et des valeurs basses caractérisent des lésions bégnines.

3 Algorithme des Nuées Dynamiques

Nous proposons pour partitionner de l'ensemble E des images des lésions en k classes un algorithme de type Nuées Dynamiques ([**?**], [3]) généralisé au cas de données multi-valeurs. Dans son schéma classique, cet algorithme recherche une partition P^* de E en k classes non vides et un vecteur L^* de k prototypes $(g_1, \ldots, g_i, \ldots, g_k)$ qui représente, au mieux, par rapport à un critère Δ, les k classes $(C_1, \ldots, C_i, \ldots, C_k)$ de la partition P^* :

$$\Delta(P^*, L^*) = Min\{\Delta(P, L) \mid P \in P_k, L \in \Lambda_k\}.$$

avec : P_k l'ensemble des partitions de E en k classes non vides et Λ_k l'espace de représentation des prototypes. Ce critère exprime l'adéquation entre la partition P et le vecteur L des k prototypes. Il est défini comme la somme sur toutes les classes C_i et sur tous les objets m de C_i des mesures de proximités $\delta(x_m, G_i)$ entre chaque vecteur x_m de description de m et le vecteur de description G_i du prototype g_i de la classe C_i. Les mesures $\delta(.)$ sont supposées différentes par rapport au type de descripteur. Une mesure de proximité dans le cas de descripteurs à intervalles a été proposée par [5]. De Carvalho [7] a proposé une mesure à deux composantes pour les descripteurs multi-catégoriques. Une mesure L_2 entre distributions a été introduite par [6]. La mesure globale de dissimilarité entre m et g_i est obtenue par une combinaison linéaire des mesures de proximité choisies en fonction de différents descripteurs. L'algorithme procède alternativement par une étape de représentation suivie d'une étape d'allocation. Afin de séparer les classes de lésions bénignes des classes malignes, un contrôle est introduit dans la procédure.

4 Les caractéristiques discriminantes

Á partir des classes appartenant aux deux groupes (lésions bénignes et malignes), issues dans l'algorithme de classification automatique, nous calculons les dissimilarités entre les prototypes respectifs : g_i et $g_{i'}$, avec $i \neq i'$. Nous sommes intéressés par l'évaluation des dissimilarités entre les prototypes de classes de lésions bénignes et de lésions malignes. Les valeurs de la mesure de dissimilarité considérée sont calculées au niveau global par rapport à tous les descripteurs pris en compte dans l'analyse, avec un système de pondération arbitraire. Afin de tenir compte du pouvoir discriminant des différents descripteurs dans la procédure, nous proposons d'introduire un système de pondérations sur les quatre caractéristiques principales A, B, C, D et sur les leurs composantes:

$$\delta(g_i, g_{i'}) = \sum_{t=1}^{4} \lambda_t \sum_{j \in W_t} \mu_{jt} \nu_j(G_i^j, G_{i'}^j) \quad (\text{avec} \sum_{j \in W_t} \mu_{jt} = 1 \text{ et } \sum_{t=1}^{4} \lambda_t = 1) \tag{1}$$

où : W_t est l'ensemble des descripteurs appartenant aux caractéristiques A, B, C et D. Les points sont donc recherchés de façon optimale, par la maximisation d'un critère de séparabilité des classes, considérant aussi un système de pondération des classes.

Un critère de recherche des meilleurs systèmes de ponderation des descripteurs et des classes équivalent 'a été proposé in ([4], [8]). Ceci est basé sur la minimisation de la variabilité entre les classes et il reprend l'algorithme ISODATA de la classification floue [1].
La procedure a été testée sur un échantillon de 100 images choisies entre différents types de lésions en considérant quinze descripteurs. Les caractéristiques les plus discriminantes dans une partition en 7 classes sont celles liées à l'asymétrie et à la couleur. En raison d'un nombre de pages limité, les résultats seront présentés lors de la prsentation orale.

5 Références bibliographiques

[1] J.C. Bezdek, "A convergence theorem for fuzzy ISODATA clustering algorithms", IEEE Trans. Pattern Anal. Machine Intell., vol. 2, 1980 p.1-8.

[2] H.-H. Bock, E. Diday, *Analysis of Symbolic Data, Exploratory methods for extracting statistical information from complex data*. Springer, 2000.

[3] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, H. Ralambondrainy, *Classification Automatique des Données*. Bordas, 1989, Paris.

[4] E.Y. Chan, W.K. Ching, M.K. Ng, J.Z. Huang, "An optimization algorithm for clustering using weighted dissimilarity measures", *Pattern Recognition*, vol. 37, n. 5, 2004, p. 943-952.

[5] M. Chavent, "Analyse des Données Symboliques. Une méthode divisive de classification", *Thèse de l'Université de PARIS-IX Dauphine, 1997.*

[6] M. Chavent, F. A. T. De Carvalho, Y. Lechevallier, R. Verde, "Trois nouvelles méthodes de classification automatique de données symboliques de type intervalle", *Revue de Statistique Appliquées* n. 4 , 2003, p. 5-29.

[7] F.A.T. de Carvalho, "Extension based proximities between Boolean symbolic objects", in *Data Science, Classification and Related Methods*, Hayashi, C. et al.(eds.), Springer-Verlag, 1998, Tokyo, 370-378.

[8] H. Frigui and O. Nasraoui, "Unsupervised learning of prototypes and attribute weights", *Pattern Recognition*, vol. 34, n. 3, 2004, p. 567-581.

[9] I. Maglogiannis, D. I. Kosmopoulos, "Computational vision systems for the detection of malignant melanoma", *Oncology reports* n. 15 , 2006, p. 1027-1032.

CrossStream: Résumé de Flux Relationnels

B. Csernel[1], F. Clerot[2] and G. Hébrail[1]

1. ENST, 46, rue Barrault, 75634 Paris cedex 13
2. France Télécom R&D, 2 avenue P. Marzin, 22307 Lannion
(csernel@enst.fr, fabrice.clerot@francetelecom.com, hebrail@enst.fr

Mots clés : Fouille de Flux de Données Relationels, Fouille de Flux de Données, Classification, Résumé de Flux de Données, Jointure de Flux de Données.

1 Introduction

Au cours de la dernière décade, la quantité d'informations générée par la plupart des processus commerciaux et industriels a considérablement augmenté, que ce soit en rythme ou en quantité. Ceci a amené le développement d'un nouveau champ dans le domaine de l'analyse de données : l'analyse de flux de données. Il a pour objet l'étude de flux de données potentiellement infinis, arrivant à un rythme si rapide qu'il n'est pas possible de les stocker sur disque et nécessitent donc un traitement à la volée. Ce nouveau champ, a été le sujet d'une attention croissante ces dernières années, aussi bien de la part des universitaires que des industriels, en éveillant l'intérêt de membres issus de plusieurs communautés, fouille de données, mais aussi, base de données, statistiques et apprentissage.
Ce travail étudie la conception de résumés construits à partir de tels flux de données. La tâche consistant à résumer un flux donné a déjà fait l'objet de nombreuses recherches. En particulier, un certain nombre de méthodes, dont celles sur lesquelles sont basées ce travail, ont à leur base des méthodes de classification automatique utilisées dans le but de résumer de l'information. Cependant, toutes ces méthodes sont destinées au traitement d'un flux unique. Hors, dans les applications industrielles, les données sont rarement isolées mais incluent souvent des références à d'autres données produites par des sources différentes. C'est pourquoi ce travail porte sur le résumé non pas d'un seul flux, mais de plusieurs flux de données liés par des relations.
Pour simplifier le problème, dans cet article, seul sera considéré un exemple comportant trois flux de données : deux flux d'entités joints par un flux de relations.

2 Notions de Flux de Données

Formellement parlant, on définit un flux de données comme une séquence infinie d'éléments datés produite à un rythme rapide par rapport aux capacités de stockage et de traitement [5]. De par la nature des flux, les algorithmes de traitement des flux de données sont soumis à des contraintes supplémentaires par rapport aux algorithmes classiques. Tout d'abord, ils ne peuvent opérer qu'une seul passe séquentielle sur les données, puisqu'une fois le flux passé les informations qu'il contenait sont perdues. Par ailleurs, le temps de traitement par élément doit impérativement être court. En effet, il faut que ces algorithmes opèrent en utilisant des quantités d'espace disque et d'espace mémoire

bornées et faibles pour chaque éléments de façon à pouvoir supporter la volumétrie et le débit des flux de données.
Un autre problème lié à la gestion des flux de données est celui de la gestion du temps. En effet, s'il est possible d'appliquer le même traitement sur l'intégralité d'un flux, il est souvent souhaitable de n'appliquer ce traitement qu'à une portion du flux non nécessairement connues à l'avance. Il est donc important pour un algorithme de traitement de flux de données, plus particulièrement pour un algorithme de résumé de flux de données, de pouvoir préciser à posteriori sur quelle partie du flux doit porter le résumé. On parle souvent d'horizon temporel sur lequel porte le traitement.

3 État de l'Art

Ce travail est lié à de nombreux travaux réalisés sur les flux de données pendant ces dernières années [4] [5]. Deux autres sujets d'étude lui sont plus particulièrement liés.
Le premier est l'étude des techniques de résumé. Ces techniques sont bien sûr nombreuses et souvent spécialisées dans la production de résumés destinés à une tache bien particulière : comptage, classification, fouille de donnée supervisée. La technique de résumé sur laquelle s'appuie CrossStream est basée sur CluStream, un algorithme de classification automatique de flux de données présenté par Aggarwal dans [1] dont le fonctionnement est basé sur la conservations de nombreuses micro-classes qui sont ensuite utilisées pour réaliser la classification finale. Nous l'adaptons ici à la création de résumés généralistes.
Le second est l'étude de la jointure entre deux flux. En base de données classique, ce processus consiste à assembler deux tables. La première, qu'on appelle la table d'entité contient une liste d'éléments, par exemple des clients, chacun d'entre eux indexé par un identifiant unique qu'on appelle une clé et possédant un nombre quelconque d'attributs. La seconde, qu'on appelle la table de faits contient elle aussi une liste d'éléments, par exemple des ventes de billets, mais chaque élément possède parmi ses attribut une valeur de la clé de la table d'entité. Dans notre exemple, chaque vente de billet a comme propriété l'identifiant du client qui a acheté ce billet. Joindre ces deux tables consiste alors à former une nouvelle table semblable à la table des faits, où chaque vente de billet possède les attributs qu'il avait dans la table originelle, et où l'identifiant du client à été remplacé par l'ensemble des attributs que possède ce dernier dans la table d'entité. Réaliser une jointure totale et exacte entre deux flux est impossible car cela demanderais de stocker l'intégralité du flux d'entités, ce qui est par définition impossible. En revanche, des algorithmes ont tout de même été établis pour réaliser une jointure ne portant pas sur l'intégralité des flux, mais seulement sur un horizon temporel donné de chacun des deux flux [3].
Bien que ces deux problèmes aient déjà été étudiés séparément et que le problème étudié ici semble n'être que la conjonction des deux, il s'agit bien d'un problème nouveau. En effet, le but ici n'est pas de réaliser d'abord la jointure des flux puis de les résumer ensuite, mais bien de constituer un résumé de tous les flux sans avoir à calculer la jointure, tout en tenant compte des informations relationnelles.
Ce problème n'a pas encore été le sujet de beaucoup d'attention à notre connaissance, et c'est pourquoi nous avons décidé d'essayer d'y apporter une solution.

4 CrossStream

4.1 Présentation du Problème

Le problème considéré ici est de résumer l'information contenue dans trois flux de données partageant un lien relationnel, de façon à ce que ce résumé puisse servir à récupérer des informations aussi bien sur n'importe lequel des trois flux que sur les liens qu'ils partagent, et ce pour n'importe quel période de temps. Plus formellement, on considère deux flux d'entités E et F et un flux de relations R. Chacun de ces flux est une succession d'éléments possédant un nombre fixe d'attributs propres à chaque flux. Les deux flux d'entités sont de plus indexé chacun par une clé tandis que chaque élément du flux de relation a parmi ses attributs un couple de clés. On peut par exemple imaginer que E représente un flux de clients, chaque client possèdent un identifiant de client et que F est un flux de services dont chaque service posséderait un code service. Le flux R peut alors être le flux des utilisations d'un service par un client, chaque élément comportant un code service, un identifiant client et les informations liées à l'utilisation de ce service par ce client.
Nous faisons trois hypothèses supplémentaires sur la nature des flux. La première est de présupposer que les deux flux d'entités ont un rythme faible comparé à celui du flux de relations. La seconde est de supposer que les distributions de données des éléments constituant les deux flux d'entités changent lentement au cours du temps. Finalement, on suppose aussi que les clés qui apparaissent dans le flux de relations R sont déjà apparus dans les flux d'entités E et F.

4.2 Principe de l'Algorithme

L'algorithme présenté ici se base tout comme CluStream sur l'utilisation de micro classes comme structure de résumé. Chacune de ces micro classes est représenté dans la structure de donnée de l'algorithme par un vecteur, appelé Vecteur Caractéristique de Classe (CFV, Zhang [6]), et un identifiant. Ce CFV contient pour chaque micro-classe son effectif, et pour chaque variable, la somme des valeurs prises par les différents éléments contenus dans la classe, la somme de leurs carrés ainsi que la somme des étiquettes de temps de tous les éléments de la micro-classe et la somme de leurs carrés.

Chaque flux d'entités est résumé en utilisant un nombre fixe de micro-classes (N_E pour E et N_F pour F). Ces dernières sont mises à jour à l'arrivée de chaque nouvel élément du flux, ce qui peut se traduire par la création de nouvelles micro-classes et la fusion de deux micro classes. De plus, un filtre de Bloom [2] est rattaché à chacune des micro-classes. Sa fonction est de mémoriser la valeur des clés de tous les éléments qui ont été assignés à cette micro classe. En effet, un filtre de Bloom est une structure de données très compacte permettant de mémoriser un ensemble de nombres et de déterminer ensuite si un nouveau nombre fait ou non partie des nombres ayant été précédemment mémorisés.

Par ailleurs, le flux de relations est quant à lui résumé en utilisant comme structure un tableau croisée d'effectifs de taille N_ExN_F. Dans le flux R, chaque élément est relié à travers son couple de clés à un élément unique de chaque flux d'entités. Chaque élément est donc connecté à deux micro classes, une associé à E et l'autre à F. Si la micro classe de E (resp. F) contenant l'élément de E partageant la même valeur de clé est C_{Ei} (resp. C_{Fj}), alors cet élément contribue à l'effectif maintenu dans la case d'index (i, j) du tableau croisé des effectifs.
Enfin, pour prendre en compte l'aspect évolutif du flux et rendre compte de son état à

différentes périodes de l'état du flux, un système de cliché similaire à celui utilisé dans CluStream est utilisé par CrossStream. Ce système sauvegarde sur disque à intervalles réguliers l'état du système, en l'occurrence les CFV des micro classes et le tableau croisé des effectifs. Ces clichés sont ensuite conservés selon une structure géométrique ou pyramidale de façon à en conserver un plus grand nombre pour les temps proches du temps courant que pour les périodes plus anciennes tout en utilisant une quantité bornée de mémoire. A partir de ces clichés peut ensuite être reconstitué l'état qu'aurait eu le système s'il n'avait tourné que pendant un horizon temporel donné, permettant d'appliquer des traitements uniquement sur les données relatives à cet période.

5 Conclusion et perspectives

Cet article décrit un système de résumé de plusieurs flux de données partageant des liens relationnels entre eux. Il construit des résumés fournissant des informations aussi bien sur chaque flux pris individuellement, que sur les relations qu'ils partagent, et ce pour n'importe quel horizon temporel.
Les travaux que nous effectuons actuellement concernent l'analyse des performances de CrossStream dans diverses situations de façon à bien évaluer ses capacités de résumé en fonction de la nature des données. Dans un futur plus lointain, nous envisageons d'étendre l'utilisation de CrossStream à des structures relationnelles de flux plus complexes voir à une structure relationnelle quelconque.

Références

[1] C.C. Aggarwal, J. Han, J. Wang, and P.S. Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th VLDB Conference*, Berlin, Germany, 2003.

[2] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7) :422–426, 1970.

[3] A. Das, J. Gehrke, and M. Riedewald. Approximate join processing over data streams. In *SIGMOD '03 : Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 40–51, New York, NY, USA, 2003. ACM Press.

[4] L. Golab and M.T. Ozsu. Data stream management issues - a survey. Technical report cs 2003-08, University of Waterloo, Waterloo, Canada, April 2003.

[5] S. Muthukrishnan. Data streams : algorithms and applications. *Found. Trends Theor. Comput. Sci.*, 1(2) :117–236, 2005.

[6] T. Zhang, R. Ramakrishnan, and M. Livny. Birch an efficient data clustering method for very large databases. In *SIGMOD*, Montreal, Canada, June 1996. ACM.

Différences, distances entre textes Sanskrits, élaboration d'édition critique.

M. Csernel[1,2] and P. Bertand[3]

1. Inria,projet AXIS, Domaine de Voluceau 78153 le Chesnay Cedex
2. Univeristé Paris-Dauphine, 1 place du M^al^Delattre de Tassigny75016 ParisCedex 16
3. GET - ENST Bretagne, Equipe Lussi (UMR 2872 Tamcic), CS 17607, 35576 Cesson Sévigné Cedex, France
Marc.Csernel@inria.fr, Patrice.Bertrand@enst-bretagne.fr

Mots clés : édition critique, distance d'édition, phylogénie de manuscrits, L.C.S.

Etablir les différences entre deux textes, en termes de mots, peut apparaître évident dans une langue comme le français où les mots sont identifiés par des espaces séparateurs. Ceci est exact non seulement en français mais dans toutes les langues européennes modernes. Dans une langue comme le Sanskrit, où la frontière entre les mots n'apparaît pas grâce à la présence de séparateurs, le problème prend une toute autre dimension, surtout si on ne dispose pas d'un lexique informatique fiable. Dans cet article nous indiquons comment procéder pour déterminer, en terme de mots, les différences entre deux textes Sanskrits, puis, à partir de ces différences, générer des distances intertextuelles et le texte d'un apparat critique. Après cette étape, la détermination et l'étude de filiations entre textes devient possible via la construction d'arbres phylogénétiques.

L'identification des différences entre deux versions d'un même texte dépend évidemment des règles d'écriture inhérentes au langage qui est utilisé. Par exemple, la différence entre "Le chat est mort" et "Le petit chat est mort" [7] est triviale non seulement pour un lecteur attentif de *l'école des femmes*, mais aussi pour un programme informatique relativement simple, qui compare les textes caractère par caractère, et qui "sait" qu'une séquence de caractères commençant et se terminant par un espace délimite un mot. Ceci est exact non seulement en français mais dans toutes les langues européennes modernes.

Lorsque les différences portent sur des versions d'un même texte, et que ce texte constitue un document de quelque importance littéraire ou scientifique, on peut commencer à parler d'édition critique. Une édition critique est une édition où l'on fait apparaître toutes les différences connues entre les différentes versions de ce texte, en termes de mots supprimés, ajoutés, ou remplacés. Dans la plupart des langues européennes, il existe depuis longtemps des outils informatiques [5][9] permettant d'aider le philologue dans l'accomplissement de cette tache qui peut devenir longue et fastidieuse lorsque le nombre de versions différentes du texte à prendre en compte est important.

Notre but est d'apporter une aide aux philologues dans l'établissement d'éditions critiques de textes écrits en Sanskrit, ou dans toute autre langue possédant des caractéristiques similaires. Le Sanskrit s'écrit sans blanc ni autre séparateur entre les mots, comme c'était l'usage en Europe dans les manuscrits du haut moyen age. Dans une langue comme le Sanskrit, où il n'existe pas dans les textes de frontière entre les mots, la détection des différences entre les textes exprimée sous forme de mots, prend une toute autre dimension, surtout en l'absence de lexique informatique fiable. En effet un programme qui n'a pas ni lexique ni séparateur pour déterminer les frontières de mots se heurte à deux problèmes :

- La notion de mot devenant floue, il faut essayer de se baser sur d'autres notions telles les phonèmes pour essayer de les déterminer. Le programme ne peut proposer qu'un choix parmi un ensemble de solutions possibles.
- Le nombre de solutions à examiner augmentant de manière exponentielle en fonction de la longueur des chaînes de caractères, les temps de traitement suivent naturellement. On se trouve avec des programmes qui du fait de leurs temps d'exécution trop long n'arrivent plus à apporter une aide efficace aux philologues.

Pour éviter ces problèmes, nous avons décidé de bâtir les éditions critiques à partir de deux textes de types différents : un texte lemmatisé où tous les mots apparaissent de manière séparée, (qui les textes sanskrit s'appelle un *padapatha* en suivant le nom d'une forme de récitation faite en détachant bien toutes les syllabes), et un texte de manuscrit (saisi sous forme d'un fichier "texte", dans lequel apparaissent aussi toutes les remarques et annotations du collationneur).

Le texte lemmatisé (*padapatha*) par rapport auquel les comparaisons de tous les manuscrits vont s'effectuer est constitué à partir du texte de l'édition. Les différences qui vont apparaître entre ce texte et les textes des manuscrits, constituent une part essentielle de l'apparat critique. Le *padapatha* constitue alors un lexique implicite.

L'on pourrait penser que l'essentiel des problèmes qui président à l'élaboration de programmes informatique produisant une édition critique de textes sanskrit est résolu, mais il n'en n'est rien. D'une part le Sanskrit s'écrivant à l'aide d'un alphabet de 48 lettres transliterré suivant un codage mis au point par Franz Velthuis [10], une lettre de Sanskrit translittérée correspond à 1, 2 ou 3 caractères latin, ce qui entraîne nécessairement un prè-traitement pour ne pas comparer les lettre latines de la translittération mais les lettres de l'alphabet sanskrit. D'autre part le Sanskrit offre, pour le profane en la matière, d'autres particularités surprenantes, dont la plus importante, de notre point de vue, est celle des transformations morpho phonétiques appelées *shandi*.

L'alphabet du sanskrit s'apparente à un syllabaire : il traduit exactement la prononciation d'un texte dans son écriture. Aussi les liaisons qui s'effectuent entre les mots lorsque l'on parle, se reflètent-elles dans l'écriture. Deux mots écrits avec une séparation entre eux sont écrits de manière différente que s'ils étaient écrits sans séparation. L'ensemble des transformations qui président au passage de 2 mots séparés à une seule séquence de caractères s'appelle un *shandi*. Il existe un ensemble de règles strictement codifiées par la grammaire qui déterminent les *shandi* [8]. Cela signifie que le texte lemmatisé ne peut pas être comparé directement avec le texte d'un manuscrit, il doit, avant d'être comparé, être transformé en un texte sans séparation avec application des transformations générées par les *sandhis*, en gardant en mémoire la place des diverses séparations. C'est seulement grâce à la mise en mémoire des séparations trouvées dans le texte lemmatisé que le programme peut déterminer quels mots ont été modifiés, omis, ou rajoutés. Notons que pour les mots rajoutés, ces mots n'étant pas lemmatisés, le programme se contente d'effectuer des hypothèses, c'est le travail de l'éditeur de fournir sous forme de mots l'ensemble des termes rajoutés.

Avant de décrire la manière de procéder, il convient de rappeler la définition d'une distance d'édition entre deux chaînes de caractères X et Y. C'est le nombre minimum d'opérations nécessaires pour passer de X à Y, chaque opération pouvant être la suppression, l'ajout, ou la transformation d'une lettre.

Pour trouver quels sont les mots qui différent entre le texte de l'édition (*padapatha*) et le texte d'un manuscrit, la technique que nous employons est proche des techniques employées pour comparer les séquences moléculaires en biologie[3][4]. Elle est basée sur l'algorithme

L.C.S. (Longest Common Sequence) [4][6], qui permet d'obtenir une ou plusieurs des plus Longues Séquences Communes entre deux chaînes de caractères X et Y. Cet algorithme bâtit, en utilisant la technique de l'algorithme de la programmation dynamique, une matrice T où le texte X apparaît en ligne, et Y en colonne, le i ème caractère de chaque chaîne étant noté X[i] et Y[i] respectivement.

Chaque terme T[i,j] de la matrice contient le nombre de caractères communs entre les *i* premiers caractères de la chaîne X et les *j* premiers caractères de la chaîne Y, le coin bas droite de la matrice contenant la longueur de toutes les L.C.S. possibles entre X et Y. Cette valeur est parfois notée *lcs* (X,Y) et peut être considérée comme le dual de la distance d'édition entre X et Y calculée sans utiliser d'opération de transformation.

Chacun des T[i,j] est calculé en utilisant l'algorithme de la programmation dynamique suivant la formule :

$$T[i,j] = \begin{cases} T[i-1, j-1] + 1 & \text{si } X[i] = Y[j] \\ \max(T[i-1, j], T[i, j-1]) & \text{dans les autres cas} \end{cases}$$

Le lecteur attentif aura observé qu'une édition critique peut être considérée comme une distance d'édition formulée en terme de mots entre plusieurs textes, et que par un curieux détour de la pensée, pour obtenir cette distance, nous nous servons d'une distance d'édition formulée en terme de lettres entre les chaînes de caractères qui constituent les textes à comparer. L'exemple ci-après montre comment nous procédons pour comparer deux séquences de Sanskrit: Y= **tasmai "srii_gurave namas** et X = **"sriiga.ne"saayanama.h**, la première appartient au *padapatha* et sert de lexique et de guide pour la comparaison. Les mots du *padapatha* apparaissent séparés par des barres horizontales.

		"s	r	i i	g	a	n	e	"s	a a	y	a	n	a	m	a	.h
t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
s	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
m	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2
ai	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2
"s	0	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
r	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
ii	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3
g	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4
u	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4
r	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4
a	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5
v	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5
e	0	1	2	3	4	5	5	6	6	6	6	6	6	6	6	6	6
n	0	1	2	3	4	5	5	6	6	6	6	6	7	7	7	7	7
a	0	1	2	3	4	5	5	6	6	6	6	7	7	8	8	8	8
m	0	1	2	3	4	5	5	6	6	6	6	7	7	8	9	9	9
a	0	1	2	3	4	5	5	6	6	6	6	7	7	8	9	10	10
.h	0	1	2	3	4	5	5	6	6	6	6	7	7	8	9	10	11

La case en bas à droite, contient la valeur 11 qui indique le nombre de lettres communes entre X et Y, les rectangles grisés ont tous une borne correspondant aux limites fournies par le *padapatha.* Le rectangle en gris foncé correspond au mot tasmai qui manque dans Y, les 2 rectangles gris moyens correspondent aux mots présents à la fois dans X et Y, **"srii** et **nama.h** (**nama.h** et **namas** sont équivalent du fait d'un *sandhi*), enfin le rectangle gris clair correspond au remplacement de **gurave** par **ga.ne"saaya.** Parmi tous les alignements possibles proposés par l'algorithme L.C.S. nous choisissons celui qui suit et qui reflète les

commentaires que nous venons d'effectuer. Y le *padapatha* apparaît dans la ligne du haut, X: le texte du manuscrit, dans la ligne du bas.

t	a	s	m	ai	"s	r	ii	g	u	r	a	-	v	e	-	-	-	-	n	a	m	a	.h
-	-	-	-	-	"s	r	ii	g	-	-	a	.n	-	e	"s	aa	y	a	n	a	m	a	s

Cet exemple **simple** montre sur quelles bases peut s'effectuer une comparaison de textes sanskrits. Malheureusement tout n'est pas toujours aussi simple et le processus peut se compliquer considérablement. Un des problèmes est lié à la disparition d'un mot ayant généré un sandhi, quand ce mot manque le sandhi n'a plus de raison d'être et il faut "défaire" après coup le sandhi initialement généré.

Nous avons maintenant un programme qui peut fournir des résultats satisfaisants pour des philologues, et qui de plus peuvent servir à générer des arbres phylogénétiques [1][2] et étudier les liens de filiations existant entre les manuscrits, sans pouvoir néanmoins espérer atteindre le graal des philologues, la découverte du manuscrit original d'où découle tous les autres.

[1] BUNEMAN P. (1971) Filiation of Manuscripts Mathematics in Archeological en Historical Sciences Edinburgh University Press.

[2] BARTHELEMY J.-P. & GUENOCHE A. (1991) "Trees and Proximity Representations", *John Wiley & Sons* (première édition française : Les arbres et les représentations des proximités, *Paris : Masson 1988*).

[3] C. CHARRAS & T. LECROQ: Sequence Comparison http://www-igm.univ-mlv.fr/~lecroq/seqcomp/seqcomp.ps

[4] Crochemore M.,Hancart C, Lecroq T. "Algorithmique du texte" Vuibert, *Paris, chap 7, pages 223--264, 2001*

[5] Hagel S. "Classical Text Editor" http://www.oeaw.ac.at/kvk/cte/

[6] HUNT J.W. & SZYMANSKI T.G. "A fast algorithm for computing longest common subsequence" *CACM 20:5 1977.*

[7] Poquelin J.B. (dit Molière) "L'Ecole des femmes Acte II, Sc 5" *Librio: octobre 2003*

[8] RENOU L. "Grammaire sanskrite : phonétique, composition, dérivation, le nom, le verbe, la phrase" *Maisonneuve 3. éd. revue, corr. et augm. (1996), Paris.*

[9] Robinson, P., 'Collate: A Program for Interactive Collation of Large Textual Traditions', in Hockey and Ide (1994), 32-45.

[10] Velthuis, F. Devanagari for TeX, version 1.2 (User Manual}, University of Groningen, May 1991

La morphologie mathématique : un outils pour la classification des données multidimensionnelles et des hyper rectangles

J. P. Rasson[1] and F. A. T. De Carvalho[2]

1. Facultés Universitaires Notre Dame de La Paix, Département de Mathématique, 8 Rempart de la Vierge, B-5000 Namur, Belgique
2. CIn-UFPE, Av. Prof. Luiz Freire, s/n – Cidade Universitária, CEP 50740-540, Recife-PE, Brésil
(jean-paul.rasson@fundp.ac.be, fatc@cin.ufpe.br)

Mots clés : morphologie mathématique, classification, connexité

Au dernier congrès ISI (Sydney 2005), Y. Lechevallier et J. P. Rasson ont présenté « Une représentation parcimonieuse des hyper rectangles ». Il s´agissait simplement de l´enveloppe convexe FAIBLE de la classe des hyper rectangles (l´enveloppe convexe, s´il s´agit d´un domaine fermé convexe n´est autre que la limite de la fermeture optimale par une hyper sphère, lorsque le rayon de celle-ci tend vers l´infini). Dans ce cas-ci et sous les mêmes hypothèses, il s´agit de la fermeture par un carré dont le coté tend vers l´infini. Ceci permet, entre autre, de ne pas tenir compte, contrairement aux autres représentations, du « vide » inutile qui existe entre deux hyper rectangles. Il convenait bien sur de chercher les règles de classification correspondantes :

- Pour la classification non-supervisée, le règle du maximum de vraisemblance est, dans sa conception, d´une facilité dérisoire, quant à l´algorithme …
- Pour l´analyse discriminante, non SEULEMENT la règle est très facile, mais le calcul de la règle d´affectation se résume à celui de calculs de volumes d´hyper rectangles.

Il va de soi que dans d´autres conditions de fermetures optimales, sans que le rayon ou le coté tende vers l´infini, d´autres règles et formules (y compris analytiques) seront exposées. Si on prend en compte le concept de la « 4-connexité » en morphologie mathématique, la classification des données multidimensionnelles se ramène aux règles exposées ci-dessus.

[1] J.-P. Rasson, Y. Lechevallier, “Parcimonious representation of symbolic objects : the case of multidimensional intervals, ISI 2005 Sydney, Invited paper.
[2] M. Schmitt, J. Mattiolli, “Morphologie Mathématique.”, Ed. MASSON, Paris, 1994.

Une méthode de partitionnement sur un ensemble de tableaux de distances

F.A.T de Carvalho[1] and Y. Lechevallier[2]

1. CIn Centro de Informtica UFPE Universidade Federal de Pernambuco Av. Prof. Luiz Freire s/n Cidade Universitria CEP 50740-540 Recife-PE Brasil
2.INRIA-Rocquencourt Domaine de Voluceau - Rocquencourt B.P. 105 - 78153 Le Chesnay Cedex, France
(fatc@cin.ufpe.br,yves.lechevallier@inria.fr)

Mots clés : tableau de distances, partitionnement.

1 Introduction

L'objectif de ce texte est de proposer une nouvelle méthode de classification sur un ensemble de tableaux de distances. Soient E l'ensemble des objets à classer et p tableaux de distances définis sur E. L'objectif est de classer les objets de E en tenant compte de ces p tableaux de distances. La première stratégie est de construire un nouvel espace métrique (E, d) avec d comme combinaison linéaire des p distances issues des p espaces métriques $(E, d_1), ..., (E, d_j), ..., (E, d_p)$. Si les pondérations sont positives alors (E, d) est espace métrique sur lequel on peut utiliser toutes les méthodes de classification adaptées au partitionnement de E sur cet espace. Le rôle d'un tableau de distances est lié à la pondération qui lui est associée, plus cette pondération est importante plus ce tableau va jouer un rôle dans la classification. Le choix de ces pondérations est très important aussi nous proposons de calculer automatiquement ces pondérations dans notre processus de classification.

La solution généralement proposée est d'associer une structure classificatoire à cet ensemble E muni de la distance d. La plupart des algorithmes proposent la structure hiérarchique et utilisent une procédure aggregative (*cf.*[9], [8]) mais aussi on peut trouver quelques approches divisives (*cf.*[11], [2]). Si l'objectif est d'obtenir une partition alors on peut utiliser toutes les méthodes de partionnement adaptées au tableau de distances par exemple [1] ou [12].

La seconde stratégie est de reprendre les travaux de G. Govaert (*cf.* [10], [6]) sur les distances adaptatives. Dans notre cas l'adaptation consiste à recherche un vecteur de pondérations entre les différents espaces métriques qui optimise un critère d'évaluation Δ du partitionnement obtenu. Remarquons que si chaque tableau de distances est calculé à partir d'une variable, notre solution est très proche de celle proposée en [10]. Nous avons choisi la méthode des Nuées Dynamiques ([5], [7]) car elle permet une généralisation facile et donne une solution efficace à notre problème. Notre approche pourra être globale ou locale, dans le cas de notre approche locale ces pondérations seront dépendantes des classes.

2 Schéma de l'algorithme de classification de type Nuées Dynamiques

Le schéma de l'algorithme de partitionnement proposé est de type Nuées Dynamiques (*cf.*[5],[1]). Cet algorithme recherche une partition P^* de E en K classes non vides et un vecteur L^* de K prototypes $(c_1, \ldots, c_k, \ldots, c_K)$ qui représente cette partition. L'adéquation entre la partition et ce vecteur de prototypes est mesurée par un critère Δ fixé a priori. Γ est l'espace des prototypes.

$$\Delta(P^*, L^*) = Min\{\Delta(P, L) \mid P \in P_K, L \in \Gamma^K\}$$
$$\text{avec } \Delta(P, L) = \sum_{k=1}^{K} \sum_{s \in C_k} d^2(s, c_k),\ C_k \in P, L = (c_1, ..., c_k, ..., c_K), c_k \in \Gamma$$
$$\text{et } d^2(s, c_k) = \sum_{j=1}^{p} \lambda_j . d_j^2(s, c_k)$$

Ce critère Δ est défini comme la somme sur tous les objets s de E des carrés des distances $d(s, c_k)$. Le prototype est un élément de E donc ici l'ensemble des prototypes Γ est l'ensemble E. Si la distance d est une combinaison linéaire à coefficients positifs les p distances alors la converge de cet algorithme est démontrée dans [1].

Comme dans [10] nous proposons d'introduire dans cet algoritme une étape supplémentaire. Cette nouvelle étape permettra d'adapter la matrice des pondérations, c'est la nouvelle étape c) décrite dans l'algorithme des Nuées Dynamique suivant:

a) *Initialisation*: Soit la partition $P = (C_1, \ldots, C_k, \ldots, C_K)$ de E choisie au hasard.

b) *Étape de représentation:*

Pour toutes les classes C_k, on recherche le prototype c_k de E minimisant le critère $\sum_{s \in C_k} \sum_{j=1}^{p} \lambda_j^k . d_j^2(s, c_k)$

c) *Étape de détermination de la matrice Λ des pondérations*:

Calculer les pondérations $(\lambda_j^k), k = 1 \ldots, K, j = 1, \ldots, p$ qui minimisent le critère $\sum_{k=1}^{K} \sum_{j=1}^{p} \sum_{s \in C_k} \lambda_j^k . d_j^2(s, c_k)$ avec un système de contraintes fixé a priori sur les pondérations.

d) *Étape d'allocation:*

Pour chaque objet s rechercher sa nouvelle classe C_k d'affectation tel que $k = arg\ min_{h=1,\ldots,K} \sum_{j=1}^{p} \lambda_j^h . d_j^2(s, c_h)$

d) si les individus ne changent pas de classe alors stop, autrement aller en b)

Le critère $\Delta(P, L)$ est une fonction additive des K classes et des N éléments de E et il décroît si les conditions suivantes sont vérifiées:

- *unicité* de l'affectation des éléments de E;
- *unicité* du prototype c qui minimise le critère Δ pour toutes les classes C de E.
- *unicité* du système de pondération.

3 Les coefficients de pondération entre les distances sont variables

Comme la distance d est une combinaison linéaire de distances il faut ajouter une contrainte pour éviter le cas dégénéré $\lambda_j^k = 0$, nous avons choisi, comme dans l'approche de G. Govaert [10], que le produit de ces coefficients soit égal à 1. Les justifications de ces choix sont données dans [10] ainsi que l'utilisation des optima locaux. Nous devons maintenant calculer la matrice Λ des pondérations qui minimise le critère suivant:

$$\Delta(P, L, \Lambda) = \sum_{k=1}^{K} \sum_{s \in C_k} \sum_{j=1}^{p} \lambda_j^k . d_j^2(s, c_k) \text{ , } PetL \text{ étant fixés.}$$

d'où:

$$\Delta(P, L, \Lambda) = \sum_{k=1}^{K} \sum_{j=1}^{p} \lambda_j^k \sum_{s \in C_k} d_j^2(s, c_k) = \sum_{k=1}^{K} \sum_{j=1}^{p} \lambda_j^k . \Phi_j^k .$$

avec

$$\Pi_{k=1}^{K} \Pi_{j=1}^{p} \lambda_j^k = 1 \text{ et } \lambda_j^k > 0 \ \forall j = 1, ..., p \ \forall k = 1, ..., K$$

Ici il y a qu'une contrainte qui est globale. Les coefficients λ_j^k sont calculés avec la méthode des multiplicateurs de Lagrange. D'où:

$$\frac{\partial}{\partial \lambda_j^k} (\sum_{k=1}^{K} \sum_{j=1}^{p} \lambda_j^k . \Phi_j^k - \mu(\Pi_{k=1}^{K} \Pi_{j=1}^{p} \lambda_j^k - 1))$$

on obtient comme solution $\lambda_j^k = \frac{\mu}{\Phi_j^k}$. D'où $\mu = (\Pi_{k=1}^{K} \Pi_{j=1}^{p} \Phi_j^k)^{pK}$ avec

$$\lambda_j^k = \frac{(\Pi_{h=1}^{K} \Pi_{i=1}^{p} \sum_{s \in C_h} d_i^2(s, c_h))^{pK}}{\sum_{s \in C_k} d_j^2(s, c_k)}$$

Au lieu de prendre une contrainte globale on peut aussi prendre une contrainte par classe. Dans ce cas nous nous trouvons dans une approche locale (*cf.* [10]). Maintenant nous avons les K contraintes suivantes:

$$\Pi_{j=1}^{p} \lambda_j^k = 1 \forall k = 1, ..., K$$

D'où la solution:

$$\lambda_j^k = \frac{(\Pi_{i=1}^{p} \sum_{s \in C_h} d_i^2(s, c_h))^{p}}{\sum_{s \in C_k} d_j^2(s, c_k)}$$

4 Conclusion

Nous avons montré que l'approche proposée par G. Govaert [10] peut être facilement adaptée à des distances non euclidiennes. Dans notre cas ces tableaux de distances peuvent représenter plusieurs points de vue entre les objets d'un même ensemble. Notre approche permet d'établir automatiquement une pondération entre ces points de vue. Le choix de l'approche locale ou globale dépend du rôle que l'utilisateur veut donner à ces pondérations.

A partir d'un tableau de données ayant p variables on peut calculer sur chaque variable un tableau de distances. Si la distance utilisée est la distance euclidienne notre résultat doit être proche de celui qui peut être obtenu avec l'approche décrite en [10].

Les auteurs remercient le projet de collaboration INRIA/FACEPE (France/Brazil) de son soutien.

[1] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, H. Ralambondrainy *Classification Automatique des Données, Environnement statistique et informatique*. Bordas, Paris,1989.

[2] M. Chavent, "A monothetic clustering algorithm", *Pattern Recognition Letters*, *19*, 1998, 989–996.

[3] M. Chavent, F.A.T. De Carvalho, , Y. Lechevallier, R. Verde, "Trois nouvelles méthodes de classification automatique de données symboliques de type intervalle", *Revue de Statistique Appliquée*, 4, 2003.

[4] F.A.T. De Carvalho, , R.M.C.R. de Souza, M. Chavent, Y. Lechevallier, "Adaptive Hausdorff distances and dynamic clustering of symbolic intreval data", *Pattern Recognition Letters*, 2005.

[5] E. Diday,"La méthode des Nuées dynamiques", *Revue de Statistique Appliquée*, 19, 2, 1971, 19–34.

[6] E. Diday, G. Govaert, "Classification Automatique avec Distances Adaptatives". R.A.I.R.O. Informatique Computer Science 11 (4), 329-349, 1977.

[7] E. Diday, J.C. Simon, "Clustering Analysis". In: Fu, K. S. Eds. *Digital Pattern Recognition* Springer-Verlag, 1976, 47-94.

[8] R.O. Duda, P.E. Hart, D.G. Strok, *Pattern Classification*, Wiley-interscience, seconde édition, 2001.

[9] A.D. Gordon, *Classification*, seconde édition, Chapman & Hall/CRC, 1999.

[10] G. Govaert *Classification automatique et distances adaptatives*, Thèse de 3ème cycle, Mathématique appliquée, Université Paris VI, 1975.

[11] A. Guénoche, P. Hansen, B. Jaumard, "Efficient algorithms for divisive hierarchical clustering with the diameter criterion", *Journal of Classification*, 8, 1991, 5-30.

[12] L. Kaufman, P.J. Rousseeuw, *Finding groups in data*, Wiley, New York, 1990.

Quelques remarques sur la méthode d'ajustement de Mayer

A. de Falguerolles[1]

1. Université Paul Sabatier (Toulouse III), Laboratoire de statistique et probabilités, 31062 Toulouse cedex 9
falguero@cict.fr

Mots clés : régression, classification, droite de Mayer, Tobias Mayer.

Résumé

Le cas simple de l'ajustement de Mayer, au programme de l'enseignement secondaire il y a quelques années, avait été introduit comme un succédané de l'ajustement d'une droite de régression par la méthode des moindres carrés. Il apparaît que la démarche qui était ainsi proposée aux élèves fournit un exemple simple d'arbre de régression. Il apparaît aussi que, dans le cas général, c'est un problème de classification automatique pour lequel l'algorithme des tranferts de Régnier est particulièrement adapté sinon efficace.

1 Introduction

Cherchant à résoudre un système d'équations linéaires numériquement incompatibles, Tobias Mayer (1723–1762) propose de sommer (ou moyenner) ces équations par groupe en définissant autant de groupes d'observations qu'il y a de coefficients à estimer ; il résout alors, lorsque c'est le cas, le système de Cramer (Gabriel Cramer, 1704–1752) ainsi défini. La méthode, publiée par Mayer en 1750, exige donc qu'une partition soit fournie a priori mais ne propose pas de critère explicite permettant de guider le choix décisif de cet élément. Dans cet article, la méthode d'ajustement de Mayer est présentée comme un problème de classification. Différents critères d'optimisation pour la recherche de partitions sont introduits, et parmi ceux-ci, l'inusable critère des moindres carrés. Un algorithme de transfert (Régnier, 1965), analogue à celui introduit par Simon Régnier (1932–1980) pour rechercher une partition centrale d'une famille de partitions, permet d'affiner des solutions initiales obtenues par des méthodes usuelles de la classification automatique.

2 L'estimateur de Mayer

2.1 La méthode de construction de la droite de Mayer

Rappelons que cette méthode avait été introduite dans l'enseignement secondaire en remplacement de celle dite des « moindres carrés », vraisemblablement abandonnée en raison sa prétendue complexité. La méthode proposée, assez intuitive dans le cas simple mais difficilement généralisable au cas multiple, présente un certain nombre de propriétés qui justifient pleinement son introduction dans certains programmes officiels d'enseignement.

2.1.1 La construction officielle de la droite de Mayer

Soient X et Y deux variables statistiques quantitatives conjointement observées. Soit $\{(x_i, y_i)|i = 1, \ldots, n\})$ le nuage associé des observations. Le problème de la régression linéaire de la variable réponse Y sur une variable explicative X, avec constante, est celui de l'estimation des coefficients de l'espérance conditionnelle $\mu(x) = E[Y|X = x] = \beta_0 + \beta_1 x$. La méthode de Mayer, telle qu'enseignée en France, consiste à se donner un seuil s_X, $\min\{x_i\} < s_X < \max\{x_i\}$, et à construire la droite passant par les deux points de coordonnées $(\overline{x[x < s_X]}, \overline{y[x < s_X]})$ et $(\overline{x[x \geq s_X]}, \overline{y[x \geq s_X]})$, $\overline{\bullet[condition]}$ désignant une moyenne conditionnelle. Il est alors facile de vérifier que : 1) le point moyen $(\overline{x}, \overline{y})$ du nuage appartient à la droite de Mayer ; 2) les estimations b_0 et b_1 des coefficients β_0 et β_1 sont données par des formules simples, $b_0 = \frac{\overline{x[x \geq s_X]}\ \overline{y[x < s_X]} - \overline{x[x < s_X]}\ \overline{y[x \geq s_X]}}{\overline{x[x \geq s_X]} - \overline{x[x < s_X]}}$ et $b_1 = \frac{\overline{y[x \geq s_X]} - \overline{y[x < s_X]}}{\overline{x[x \geq s_X]} - \overline{x[x < s_X]}}$. L'estimation m_i de $\mu(x_i)$, la moyenne ajustée de l'observation i, vaut alors $m_i = b_0 + b_1 x_i = \overline{y} + b_1(x_i - \overline{x})$. Dans sa version officielle, le seuil s_X est posé égal à $\overline{x}$ mais cela peut se discuter.

2.1.2 À l'ombre d'un arbre de régression simple : une droite de Mayer

Le principe de construction de la droite de Mayer évoque celui de la construction d'un arbre de régression à seul niveau :

$$\begin{array}{lll} \textbf{si} \quad x_i < s_X & \textbf{alors} & m_i = \overline{y[x < s_X]} \\ & \textbf{sinon} & m_i = \overline{y[x \geq s_X]}. \end{array}$$

C'est la variance expliquée, maximisée dans l'approche arbre de régression, qui détermine le choix de la valeur du seuil. Elle vaut ici $\frac{\sum_{i=1}^{n} \mathbf{1}(x_i < s_X)(n - \sum_{i=1}^{n} \mathbf{1}(x_i < s_X))}{n^2}(\overline{y[x \geq s_X]} - \overline{y[x < s_X]})^2$ et, là encore, le choix $s_X = \overline{x}$ n'est pas toujours optimal.

2.2 L'exemple éponyme de Tobias Mayer

L'exemple éponyme de Mayer est en fait un problème de régression linéaire multiple. Il a trait à la modélisation de données lunaires pour lesquelles est établie la relation linéaire suivante : $\beta - (90° - h) = \alpha \sin(g - k) - \alpha \sin(\theta) \cos(g - k)$ où g et h sont des données observées et k est obtenu dans des tables établies par Euler (Leonhard Euler, 1707–1783). Mayer dispose de 27 observations donc d'un système de 27 équations linéaires à 3 coefficients inconnus (β, α et $\alpha \sin\theta$).

Compte tenu des erreurs d'observations et des approximations du modèle, ce système est incompatible et Mayer est confronté au problème ancien de recherche d'une solution de compromis (Farebrother, 1998). En fait, Mayer considère une partition des observations en 3 classes (le nombre de coefficients inconnus) de même effectif ; puis, il somme (ou moyenne) les observations (les équations) au sein de chaque classe et résout le système de Cramer ainsi obtenu. Mais l'histoire ne dit pas comment Mayer a construit cette partition !

2.3 L'articulation entre régression et partition

Soit donc le problème général de régression dans lequel on considère un vecteur aléatoire réponse $\underline{Y}$ de dimension n, d'espérance mathématique $\underline{\mu} = \mathbf{X}\underline{\beta}$, et de matrice de variance diagonale $\sigma^2\mathbf{I}$ (resp. non diagonale $\sigma^2\mathbf{\Sigma}$). La matrice expérimentale $\mathbf{X}$, de dimension(n, k), est supposée de rang k, dimension du vecteur $\underline{\beta}$ des coefficients inconnus. Soit alors une partition c de l'ensemble $\{1, \ldots, n\}$ en exactement k classes et $\mathbf{C}$ la matrice de dimension (n, k) des indicatrices des classes de cette partition. Sous

réserve que la matrice $\mathbf{C}'\mathbf{X}$ soit de rang plein, les estimateurs de Mayer sont donnés des formules simples : $\widehat{\underline{\beta}} = (\mathbf{C}'\mathbf{X})^{-1}\mathbf{C}'\underline{Y}$ et $\widehat{\underline{\mu}} = \mathbf{X}\widehat{\underline{\beta}}$. Il est facile de vérifier que ces estimateurs, linéaires par construction, possèdent la propriété classique d'absence de biais. Enfin, leurs matrice de variance ont également des expressions simples : $\mathrm{Var}(\widehat{\underline{\beta}}) = \sigma^2(\mathbf{C}'\mathbf{X})^{-1}\ \mathrm{diag}(n_1,\ldots,n_k)\ (\mathbf{X}'\mathbf{C})^{-1}$ (resp. $\mathrm{Var}(\widehat{\underline{\beta}}) = \sigma^2(\mathbf{C}'\mathbf{X})^{-1}\ \mathbf{C}'\mathbf{\Sigma}\mathbf{C}\ (\mathbf{X}'\mathbf{C})^{-1}$) et $\mathrm{Var}(\widehat{\underline{\mu}}) = \mathbf{X}\ \mathrm{Var}(\widehat{\underline{\beta}})\ \mathbf{X}'$.

3 Un problème de classification ?

La méthode générale de Mayer appelle au moins deux remarques. D'abord, les estimations fournies dépendent du choix de la partition ; comment donc choisir une bonne partition ? Ensuite, à l'opposé d'un certain nombre de méthodes usuelles de régression (moindres carrés, moindres valeurs absolues, moindre médiane des carrés, moindres carrés élagués ...), la méthode ne se réfère pas explicitement à un critère global d'optimisation. Par ailleurs, l'intérêt peut être porté plus sur l'estimation des coefficients du modèle que sur celle des moyennes ajustées. On sent donc bien que le choix d'une partition correspond au choix implicite d'un critère d'optimisation qu'il faut chercher à reconnaître.

3.1 Critères d'optimisation

On est naturellement conduit à rechercher une partition $c^\star$ de l'ensemble $\mathcal{C}_k$ des partitions c en k classes telle que :

$$c^\star \in \arg\ \min_{c\in\mathcal{C}_k} F(\underline{y}, \underline{m}(c)) \qquad (\text{resp. } c^\star \in \arg\ \min_{c\in\mathcal{C}_k} H(\underline{b}_0, \underline{b}(c))\ )$$

où $\underline{y}$ désigne le vecteur réponse observé (resp. $\underline{b}_0$ désigne une valeur préspécifiée du vecteur des coefficients), où $\underline{m}(c)$ désigne le vecteur des moyennes ajustées associées à une partition c en k classes (resp. $\underline{b}(c)$ désigne le vecteur des coefficients de régression estimés), et où F est une fonction objectif exprimant la qualité globale de l'ajustement recherché (resp. H est une fonction objectif exprimant la qualité globale de l'approximation recherchée). On aura par exemple $F(\underline{y}, \underline{m}(c)) = \|\underline{y} - \underline{m}(c)\|_2^2$, ou $\|\underline{y} - \underline{m}(c)\|_1$, ou ...(resp. $H(\underline{b}_0, \underline{b}(c)) = \|\underline{b}(c) - \underline{b}_0\|^2_{(\mathbf{X}'\mathbf{X})^{-1}}$, ou ...). On pourrait aussi vouloir combiner les deux critères F et H ou encore s'inspirer de la démarche *lasso* (Tibshirani, 1996). Dans ce dernier cas, les variables explicatives étant préalablement centrées et réduites, il faudrait rechercher une partition $c^\star$ de l'ensemble $\mathcal{C}_k$ des partitions c en k classes telle que $c^\star \in \arg\ \min_{c\in\mathcal{C}_k} \|\underline{y} - \underline{m}(c)\|_2^2$ sous les contraintes $\|\underline{b}(c)\|_1 \leq t$ et $\mathbf{C}'\mathbf{X}$ régulière.

3.2 Une réponse localement optimale

Les problèmes d'optimisation définis ci-dessus sont assez compliqués à résoudre du fait notamment de la nature particulière des contraintes. Par ailleurs, il reste hors de question d'évaluer systématiquement le critère à optimiser pour chacun des éléments de l'ensemble des partitions en k classes telles que $\mathbf{C}'\mathbf{X}$ soit régulière. L'emploi de l'algorithme des transferts de Régnier semble ici tout à fait indiqué. En effet, supposons qu'à l'étape r on dispose d'une partition admissible c ; on recherche alors s'il existe un transfert d'un élément d'une classe dans une autre qui fournisse une partition admissible améliorant la valeur du critère à optimiser ; le cas échéant, on effectue le transfert sinon la partition est déclarée localement optimale.

Considérons donc une partition courante c et les estimations $\underline{b}(c)$ et $\underline{m}(c)$ associées. Désignons par $\mathbf{G}$ la matrice $\mathbf{C}'\mathbf{X}$ et par $\underline{g}$ le vecteur $\mathbf{C}'\underline{y}$: $\underline{b}(c) = \mathbf{G}^{-1}\underline{g}$ et $\underline{m}(c) = \mathbf{X}\ \underline{b}(c)$.

Sans perte de généralité, considérons alors le transfert de l'élement i de la classe 1 vers la classe 2. Soient $\underline{x}'_i$ le $i^{\text{ème}}$ vecteur ligne de la matrice $\mathbf{X}$ et $\underline{t}'_i = (-1, 1, 0, \ldots, 0)$ le vecteur de dimension k associé à ce transfert. Il n'est pas difficile de vérifier que ce transfert aboutit à réviser simplement les éléments de calcul fournissant les estimations courantes. En effet dans ce transfert : $\mathbf{G} \leftarrow \mathbf{G} + \underline{t}_i\ \underline{x}'_i$ et donc $\mathbf{G}^{-1} \leftarrow \mathbf{G}^{-1} - \frac{1}{1+\underline{x}'_i\mathbf{G}^{-1}\underline{t}_i}\mathbf{G}^{-1}\underline{t}_i\ \underline{x}'_i\mathbf{G}^{-1}$. Alors, $\underline{g} \leftarrow \underline{g} + y_i\ \underline{t}_i$, $\underline{b} \leftarrow \underline{b} + \frac{1}{1+\underline{x}'_i\mathbf{G}^{-1}\underline{t}_i}\ (y_i - m_i)\mathbf{G}^{-1}\underline{t}_i$, $\underline{m} \leftarrow \underline{m} + \frac{1}{1+\underline{x}'_i\mathbf{G}^{-1}\underline{t}_i}\ (y_i - m_i)\mathbf{X}\mathbf{G}^{-1}\underline{t}_i$.

Il est donc assez simple de rechercher systématiquement les transferts améliorant la valeur du critère d'optimisation retenu et, le cas échéant, de modifier par un transfert élémentaire la partition initialement considérée.

Enfin, toutes les formules (tant celles de de la définition générale que celles des transferts élémentaires) montrent que la méthode se prête aussi à l'introduction de poids pour les observations. Ainsi, d'un point de vue technique, la démarche de Mayer peut être étendue à l'ajustement du prédicteur linéaire de modèles linéaires généralisés (McCullagh et Nelder, 1989).

3.3 Recherche d'une partition initiale

Il s'agit ici d'initialiser la procédure de transferts en proposant une partition en autant de classes qu'il y a de coefficients à estimer. Deux classes de méthodes semblent particulièrement adaptées : des méthodes de type K-moyennes, ou *K-means*, K étant ici égal au nombre de coefficients à estimer ; des méthodes d'arbre de régression, le nombre de « feuilles » étant alors fixé a priori et égal au nombre de coefficients à estimer. Des adaptations demeurent nécessaires pour des variables qualitatives explicatives.

4 Conclusion

L'analyse des données de l'exemple éponyme de Mayer montre d'abord que la partition qu'il propose fournit des estimations qui réalisent un excellent compromis entre celles obtenues en considérant les différents critères usuels de régression (moindres carrés, moindres valeurs absolues, moindre médiane des carrés des résidus) ; formidable intuition de Mayer ? Elle montre aussi que la démarche proposée est assez performante ; mais il s'agit là d'un petit ensemble de données ($n = 27$, k=3) ! Pour de très grands ensembles de données il ne fait pas de doute que des algorithmes de transfert du type de celui décrit dans cet article ne soient trop lents et que d'autres algorithmes doivent être recherchés.

Références

[1] R. W. Farebrother, *Fitting linear relationships, a history of the calculus of observations (1750–1900)*. Springer, New York, 1998.

[2] P. McCullagh & J. A. Nelder, *Generalized linear models (2nd edition)*. Chapman and Hall, London, 1989.

[3] S. Régnier, Sur quelques aspects mathématiques de problèmes de classification automatique. *I.C.C. Bulletin*, 4, 1965, 175–191 et *Math. Sci. hum.*, 82, 1983, 13–29.

[4] R. Tibshirani, Regression shrinkage and selection via the lasso. *J. Royal Statist. Soc. (Series B)*, 58, 1996, 267–288.

Partitionnement des arcs d'un graphe pour la planification de réseaux optiques

L. Denœud, N. Puech

GET / Telecom Paris - LTCI - UMR 5141 CNRS
46, rue Barrault, 75634 Paris cedex 13
(Nicolas.Puech, Lucile.Denoeud)@enst.fr

Mots clés : graphes, partitionnement, optimisation, méthode Tabou, réseaux optiques

1 Introduction

On considère le problème d'optimisation qui découle de la planification de réseaux optiques. On dispose d'un graphe orienté $G = (X, U)$ représentant un réseau de fibres optiques et d'une matrice de demandes qui doivent être établies dans le réseau (chaque demande correspond à un couple sommet-source, sommet-destination entre lesquels il existe une demande de connexion). Grâce à la technologie WDM (*Wavelength Division Multiplexing*), chaque fibre optique peut porter un nombre fixé W de connections simultanément, chacune sur une longueur d'onde particulière. Pour qu'une demande puisse être établie, on doit lui affecter un chemin optique, c'est-à-dire un chemin allant de la source vers la destination dans le graphe ainsi qu'une longueur d'onde. L'objectif est alors de maximiser le nombre de demandes établies en respectant les contraintes suivantes :

- continuité de longueur d'onde : le long d'un chemin optique, la longueur d'onde choisie ne peut pas être modifiée.
- chaque longueur d'onde ne peut pas être utilisée plus d'une fois sur une même fibre optique.

Ce problème de maximisation, appelé RWA (*Routing and Wavelength Assignment problem*), a déjà été beaucoup étudié [8]. Il peut tout d'abord être formulé comme un problème de programmation linéaire en nombres entiers (ILP) et résolu à l'aide d'un solveur [6]. Cette formulation étant NP-difficile [3], elle n'est valable que pour des instances de petites tailles, et de nombreuses heuristiques ont été développées pour le résoudre [2]. Elles consistent en général à découper le problème en deux sous-problèmes : un problème de routage et un problème d'affectation de longueurs d'onde, traités ensuite séquentiellement.

Un algorithme glouton consiste à considérer les différentes demandes selon un ordre prédéfini. Les longueurs d'onde encore disponibles sur chaque arc sont mises à jour au fur et à mesure du déroulement de l'algorithme (au départ, il y en a W pour chaque arc). Pour chaque couple de sommets (s, d) correspondant à une demande, on détermine préalablement les K plus courts chemins de s vers d dans le graphe et on les ordonne dans l'ordre croissant de leurs longueurs. Pour chaque demande, on examine alors dans cet ordre les chemins susceptibles de la satisfaire ; si un chemin et une longueur d'onde libre ont été trouvés pour la demande considérée, celle-ci est établie. Dans le cas contraire elle est rejetée.

Cette méthode est très rapide mais la solution obtenue est très dépendante de l'ordre d'examen des demandes. Pour pallier cet inconvénient, on effectue cet algorithme N fois

en considérant des ordres initiaux générés aléatoirement. On peut montrer que cet algorithme, noté *RS* (*Random Search*), fournit d'assez bons résultats, souvent proches de l'optimum [1].

De nombreuses versions plus sophistiquées de ce problème sont étudiées dans la littérature. Il est en effet dans certains cas intéressant de prendre aussi en compte des aspects comme le coût des installations, la dynamique du trafic (demandes programmées ou aléatoires), la protection, la qualité du signal transmis (BER) [4]... Toutes ces versions plus ou moins complexes peuvent être modélisées par des programmes linéaires mais nécessitent en général la conception d'heuristiques pour pouvoir traiter des instances de tailles réelles.

2 Partitionnement des arcs du graphe

On propose ici une nouvelle approche consistant à découper le problème initial, appelé par la suite problème global, en sous-problèmes de tailles plus réduites de façon à pouvoir les résoudre de manière exacte. Ensuite, on assemble et on complète les solutions des sous-problèmes pour construire une solution approchée du problème global.

2.1 Partitionnement du problème

Le découpage du problème revient à partitionner les arcs du réseau optique de façon à minimiser le nombre de couples correspondant à une demande et qui se retrouveraient séparés dans la partition. Un partitionnement des sommets du graphe pourrait aussi convenir mais, les contraintes ne portant que sur les arcs, les classes fournies ne doivent pas nécessairement être disjointes en sommets. Une partition des sommets serait donc trop contraignante et entraînerait la perte de certains arcs, qui n'appartiendraient à aucune classe.

On considère au départ les K plus courts chemins correspondant à chaque couple de la matrice de demandes [7]. Étant donnée une partition Π, on pondère chaque chemin P d'origine s et d'extrémité d par la valeur $c(P)$ définie par :

$$c(P) = \frac{\text{nombre de demandes entre } s \text{ et } d}{1 + \text{nombre de chemins de s vers d non coupés dans } \Pi}$$

On cherche alors déterminer une partition des arcs du graphe de manière à minimiser la somme des coûts des chemins coupés (c'est-à-dire des chemins constitués d'arcs appartenant à différentes classes de Π). D'après cette pondération, plus un chemin correspond à une grande quantité de demandes, plus il est coûteux de le couper. De même, moins il existe dans le graphe de chemins non coupés de mêmes extrémités, et plus $c(P)$ est élevé. On impose en outre une contrainte sur l'équilibre de la taille des sous-problèmes, définie par le nombre de couples correspondant à une demande.

On construit une telle partition en appliquant la méthode Tabou [5]. On détermine les différents paramètres de la méthode en l'appliquant sur une instance réelle.

2.2 Résolution des sous-problèmes

On résout alors le problème de manière exacte pour les graphes partiels issus de la partition. Pour chaque graphe, on construit la matrice de demandes correspondante. Une demande est associée à un graphe partiel s'il existe dans ce graphe au moins un chemin

pouvant la satisfaire. Dans le cas où un couple source-destination est associé à plusieurs graphes partiels, on répartit le nombre de demandes correspondantes sur les différents graphes proportionnellement au nombre de chemins présents dans ces graphes.

2.3 Retour au problème global : procédure de réassemblage

Tous les chemins optiques obtenus en résolvant les sous-problèmes sont conservés pour construire la solution du problème global. Le partitionnement en arcs assure que la contrainte selon laquelle une longueur d'onde ne peut être utilisée qu'une fois sur un même arc reste satisfaite. On considère ensuite toutes les demandes non établies et les chemins coupés par le partitionnement. On utilise une heuristique rapide pour router ces demandes en utilisant les longueurs d'onde encore disponibles.

On teste la méthode proposée en la comparant avec les résultats donnés par la résolution directe du problème, soit exacte pour de petits graphes, soit approchée pour des graphes de plus grande taille (on utilise l'heuristique *RS* présentée plus haut).

La stratégie de partitionnement présentée dans cet exposé possède l'avantage d'être applicable à de nombreux problèmes liés à la planification de réseaux optiques, moyennant de légères modifications. Il suffit en effet d'utiliser le modèle de résolution adaptée pour résoudre les sous-problèmes de manière exacte, et éventuellement de modifier la fonction de coût utilisée dans la méthode Tabou ainsi que la procédure finale de réassemblage.

Références

[1] S. Al Zahr, L. Denoeud, N. Puech "Routing and Wavelength Assignment in Optical Network : Exact Resolution vs Random Search based heuristics", soumis pour pubication.

[2] D. Banerjee, B. Mukherjee "A practical approach for routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks", *IEEE Journal on Selected areas in Communications*, vol. 14, no. 5, June 1996, pp. 903-908.

[3] I. Chlamtac, A. Ganz, G. Karmi "Lightpath Communications : An approach to High-BandWidth Optical WAN's", *IEEE Transactions on Communications*, vol. 40, no. 7, July 1992, pp. 432-436.

[4] M. Ali Ezzahdi, S. Al Zahr, M. Koubàa, N. Puech, M. Gagnaire "LERP : a Quality of Transmission Dependent Heuristic for Routing and Wavelength Assignment in Hybrid WDM Networks", *Proc. of the 15th International Conference on Computer Communications and Networks (ICCCN)*, Arlington, Virginia, USA, Oct. 9-11, 2006.

[5] F. Glover, M. Laguna "*Tabu Search*", Dordrecht, Kluwer.

[6] R. Ramaswami, K.N. Sivarajan "Routing and Wavelength Assignment in All-Optical Networks", *IEEE/ACM Transactions on networking*, vol. 3, no. 5, Oct. 1995, pp. 489-500.

[7] Yen "Finding the K shortest loopless paths in a network", *Management Sci.*, 17, pp. 712-716, 1972.

[8] H. Zang, J. P. Jue, B. Mukherjee "A review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks", *Optical Networks Magazine*, vol. 1, no. 1, Jan. 2000, pp. 47-60.

La dissimilarité de bipartition et son utilisation pour détecter les transferts horizontaux de gènes

Vladimir Makarenkov, Alix Boc et Alpha Boubacar Diallo

Département d'informatique, Université du Québec à Montréal, C.P. 8888, Succ. Centre-Ville, Montréal (Québec), H3C 3P8, Canada
Courriels : (makarenkov.vladimir, boc.alix, diallo.alpha_boubacar)@uqam.ca

Mots clés : dissimilarité, bipartition, arbre phylogénétique, transfert horizontal de gènes.

1 Introduction

Le transfert horizontal de gènes (THG) est un transfert direct du matériel génétique d'une lignée d'un arbre phylogénétique (i.e. arbre additif ou X-arbre [1]) à une autre. Les bactéries et les archéobactéries possèdent des mécanismes sophistiqués qui leur permettent d'acquérir les nouveaux gènes au moyen d'un transfert horizontal. Les trois principaux mécanismes permettant aux espèces de s'échanger des gènes sont la transformation, consistant en l'acquisition de l'ADN directement de l'environnement, la conjugaison, impliquant des plasmides et transposons conjugaux, et la transduction, consistant en des transferts horizontaux par phage [3].

Il existe trois approches pour identifier les gènes qui ont subis des transferts horizontaux. La première consiste à examiner le génome de l'espèce hôte pour voir s'il contient des gènes ayant le contenu en GC ou des motifs de codon atypiques [7]. La deuxième approche consiste à vérifier si le gène étudié, ou sa partie, est présent dans un organisme et absent dans tous les organismes proches. Dans ce cas, il est beaucoup plus probable que ce gène a été introduit dans cet organisme par le THG plutôt que perdu par tous les autres organismes. La troisième approche procède par une comparaison d'un arbre phylogénétique inféré à la base des caractéristiques morphologiques ou à partir d'un gène qui est supposée être résistant aux transferts horizontaux (e.g. souvent on considère 16S rARN ou 23S rARN) et d'un arbre phylogénétique obtenu à partir de la séquence du gène étudié. Le conflit topologique entre ces deux arbres, qui sont appelés respectivement arbre d'espèces et arbre de gène, pourrait être expliqué par les transferts horizontaux. Plusieurs algorithmes permettant d'exploiter les différences topologiques entre les arbres d'espèces et de gène ont été proposés. Mentionnons ici l'algorithme de détection des THG de Hallett et Lagergren [4] qui inscrit des arbres de gène dans un arbre d'espèces et l'algorithme permettant d'identifier simultanément des duplications de gènes, des pertes de gènes, ainsi que des transferts horizontaux de Mirkin et al. [10]. L'article de Moret et al. [11] fait un survol des méthodes utilisées pour détecter des THG de même que d'autres phénomènes d'évolution réticulée. Dans cet article nous décrivons un algorithme polynomial permettant de détecter des THG en utilisant 3 critères d'optimisation différents : les moindres-carrés (MC), la distance topologique de Robinson et Foulds (RF), et la dissimilarité de bipartition. Notons que l'algorithme basé sur les critères MC et RF est celui introduit dans l'article de Makarenkov et al. [9]. La dissimilarité de bipartition sera définie et certaines de ses propriétés seront introduites dans la section suivante. Finalement, les résultats des simulations Monte Carlo effectuées pour comparer les trois critères d'optimisation seront présentés et discutés.

2 Dissimilarité de bipartition et stratégies pour la détection des transferts horizontaux

2.1 *Dissimilarité de bipartition et autres critères d'optimisation*

L'algorithme de détection des THG présenté en détail dans [9] procède par une réconciliation progressive des arbres d'espèces et de gène définis sur le même ensemble de feuilles représentant les espèces étudiées. Ces arbres sont notés T et T', respectivement. Sans perte de généralité nous supposons que les deux arbres sont binaires. À chaque pas de l'algorithme toutes les paires d'arêtes dans T sont testées pour vérifier l'hypothèse qu'un transfert horizontal a eu lieu entre elles. Plus précisément, nous

recherchons le nombre minimum de déplacements de sous-arbres de l'arbre *T* permettant de le transformer en arbre *T'*. Évidemment, plusieurs règles d'évolution doivent être incorporées dans le modèle pour le rendre plausible du point de vue biologique. Par exemple, les transferts entre les arêtes appartenant à la même lignée doivent être interdits (voir [8] ou [12] pour plus de détails sur les règles biologiques). Remarquons que le problème de recherche du nombre minimum d'opérations de transferts des sous-arbres nécessaire pour transformer un arbre en un autre a été montré NP-complet (i.e. *Sub-tree transfer problem,* Hein et al. [5]).

Dans ce papier nous présentons 3 critères d'optimisation qui peuvent être incorporés dans un algorithme de détection des THG présenté dans le paragraphe suivant. Le premier critère est celui des moindres carrés *Q* :

$$Q = \sum_i \sum_j (d(i,j) - \delta(i,j))^2 \qquad (1)$$

où $d(i,j)$ est la distance d'arbre mesurée entre les feuilles *i* et *j* dans l'arbre d'espèces *T* (ou dans l'arbre T_1 obtenu après le premier transfert de sous-arbre dans *T*) et $\delta(i,j)$ est la distance d'arbre entre les feuilles *i* et *j* dans l'arbre de gène *T'*. Le deuxième critère qui pourrait être utilisé pour estimer la différence entre l'arbre d'espèces et celui de gène est la distance topologique de Robinson et Foulds (RF) [13]. Cette distance est égale au nombre d'opérations élémentaires, consistant en la division et la fusion des nœuds, qui sont nécessaires pour transformer un arbre en un autre. Cette distance est aussi égale au nombre de bipartitions, Buneman [2], qui sont présentes dans un arbre et absentes dans l'autre.

Le troisième critère d'optimisation est la dissimilarité de bipartition que nous introduisons ici. Soient *T* et *T'* les arbres phylogénétiques binaires sur le même ensemble d'éléments (i.e. feuilles). Soit **BT** la matrice de bipartition correspondant aux arêtes internes de *T* et **BT'** la matrice de bipartition de correspondant aux arêtes internes de *T'*. La dissimilarité de bipartition, *bd*, entre **BT** et **BT'** est définie comme suit :

$$bd = \left(\sum_{a \in BT} \underset{b \in BT'}{Min} \left(Min\left(d(a,b); d(a,\bar{b})\right)\right) + \sum_{b \in BT'} \underset{a \in BT}{Min} \left(Min\left(d(b,a); d(b,\bar{a})\right)\right)\right) / 2 \qquad (2)$$

où $d(a,b)$ est la distance de Hamming entre les vecteurs de bipartition *a* et *b*, et $\bar{a}$ et $\bar{b}$ sont les compléments de *a* et de *b*, respectivement. Une telle mesure pourrait être vue comme une généralisation de la métrique de Robinson et Foulds qui prend en considération seulement des bipartitions identiques.

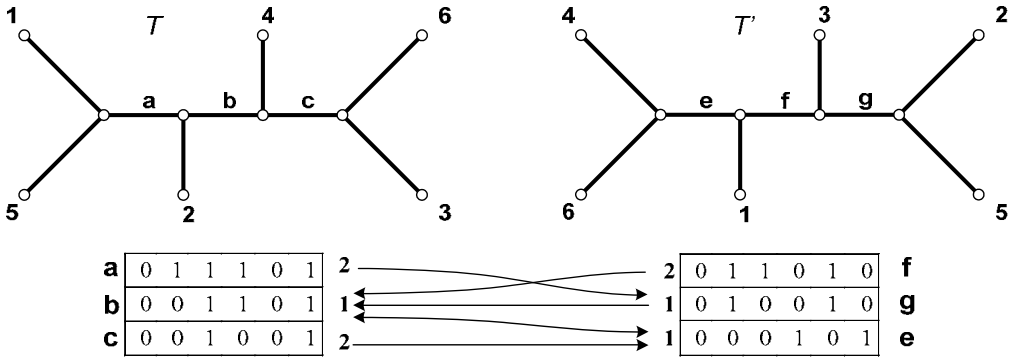


Fig. 1. Les arbres *T* et *T'* à 6 feuilles et leur tables de bipartition. Chaque ligne des tables de bipartition correspond à une arête interne. Les flèches indiquent les associations entre les bipartitions dans les deux tables. La valeur en gras à coté de chaque bipartition est la distance de Hamming associée.

Par exemple, la dissimilarité de bipartition *bd* entre les arbres *T* et *T'* à 6 feuilles montrés sur la Figure 1 est calculée comme suit : $bd(T,T') = ((2 + 1 + 2) + (2 + 1 + 1))/2 = 4{,}5$. Ici, le minimum de la distance de Hamming entre la bipartition associée à l'arête *a* et tous les vecteurs de bipartition dans **BT'** est 2 (la distance entre *a* et *e* ou entre *a* et $\bar{g}$; seulement l'association entre *a* et *e* est présentée sur Figure 1). Pour la bipartition associée à l'arête *b*, cette distance est 1 (la distance entre *b* et *e*), et pour la bipartition associée à l'arête *c*, cette distance est 2 (la distance entre *c* et *e*). De la même façon, la

distance de Hamming minimale entre la bipartition associée à *f* et toutes les bipartitions dans **BT** est 2 (voir la bipartition associée à $\bar{b}$), pour la bipartition associée à *g* cette distance est 1 (voir la bipartition associée à $\bar{b}$) et pour la bipartition associée à *e*, elle est aussi 1 (voir la bipartition associée à *b*).

Cet exemple montre que différents vecteurs de bipartition d'une table de bipartition peuvent être associés au même vecteur de bipartition de l'autre table, e. g. *e* et *g* sont associés à *b*, ainsi que *b* et *c* sont associés à *e* (Figure 1). De plus, une dissimilarité de bipartition n'est pas toujours une métrique. En commençant par des arbres à 5 feuilles nous pouvons exhiber 3 topologies d'arbre pour lesquelles l'inégalité triangulaire n'est pas satisfaite. Une condition suffisante, mais pas nécessaire, pour assurer qu'une dissimilarité de bipartition *bd* est une métrique est la suivante (ici, le symbole → désigne l'opération d'association):

Proposition 1. *Soient T_1, T_2 et T_3 des arbres phylogénétiques ayant le même nombre d'arêtes internes et le même ensemble des feuilles. Alors, $bd(T_1, T_2) \leq bd(T_1, T_3) + bd(T_2, T_3)$ si les conditions suivantes sont satisfaites*:

1. *Pour chaque paire de bipartitions a et b de 2 arbres différents*: *a → b signifie que b → a.*
2. *Pour chaque triplet de bipartitions $a \in T_1$, $b \in T_2$, $c \in T_3$: a → b et b → c signifie que a → c.*

Proposition 2. *La valeur d'une dissimilarité de bipartition entre deux arbres phylogénétiques ayant le même ensemble de n feuilles se trouve dans l'intervalle entre* 0 *et* $n(n-3)/2$, *si n est paire, et entre* 0 *et* $(n-1)(n-3)/2$, *si n est impaire.*

2.2 *Algorithme pour prédire les transferts horizontaux de gènes*

Pas préliminaire
Inférer les arbres phylogénétiques d'espèces et de gène, notés respectivement *T* et *T'*. Les feuilles de *T* et *T'* sont étiquetées par le même ensemble de *n* éléments (i.e. d'espèces). Les deux arbres doivent être enracinés. S'il existe dans *T* et *T'* des sous-arbres identiques ayant au moins 2 feuilles, réduire la taille du problème en remplaçant dans *T* et *T'* les sous-arbres identiques par les mêmes éléments auxiliaires.

Pas 1 ... *k*
Tester tous les THG possibles entre les paires d'arêtes dans l'arbre T_{k-1} ($T_{k-1} = T$ au Pas 1) à l'exception des transferts entre les arêtes adjacentes et ceux qui violent les contraintes d'évolution (voir [8] ou [12] pour plus de détails). Choisir en tant que THG optimal, le déplacement d'un sous-arbre dans T_{k-1} qui minimise la valeur du critère d'optimisation sélectionné entre l'arbre obtenu après le déplacement de ce sous-arbre et de son greffage sur une nouvelle arête, i.e. l'arbre T_k, et l'arbre de gène *T'*. Les critères d'optimisation suivants ont été testés : (1) les moindres carrés (MC), (2) la distance topologique de Robinson et Foulds (RF) et (3) la dissimilarité de bipartition (DB). Réduire ensuite la taille du problème en remplaçant des sous-arbres identiques ayant au moins 2 feuilles dans l'arbre d'espèces transformé T_k et l'arbre de gène *T'*. Dans la liste des THG retrouvés rechercher et éliminer les THG inutiles en utilisant une procédure de programmation dynamique de parcours en arrière. Un transfert inutile est celui dont l'élimination ne change pas la topologie de l'arbre T_k.

Conditions d'arrêt et complexité algorithmique
L'algorithme s'arrête quand le coefficient RF, LS ou BD devient égale à 0 ou quand aucun autre déplacement de sous-arbres n'est possible suite à des contraintes biologiques. Théoriquement, une telle procédure requière $O(kn^4)$ d'opérations pour prédire *k* transferts dans un arbre phylogénétique à *n* feuilles. Cependant, due à des réductions inévitables des arbres d'espèces et de gène, la complexité pratique de cet algorithme heuristique est plutôt $O(kn^3)$.

3 Comparaison des trois critères d'optimisation

Cette section présente les performances des 3 stratégies d'optimisation décrites ci-dessus qui peuvent être utilisées pour prédire les transferts horizontaux. Nous montrons ici seulement une partie des résultats obtenus dans nos simulations Monte Carlo. Nous avons généré 100 topologies aléatoires différentes des arbres d'espèces ayant 10, 20, …et 100 feuilles, respectivement. Chaque topologie a été obtenue en utilisant la procédure de génération d'arbres proposée par Kuhner et Felsenstein [6]. Pour chaque arbre d'espèces, nous avons généré des arbres de gène correspondant aux différents nombres de

transferts tout en respectant les contraintes d'évolution. Le nombre exact de transferts variait de 1 à 10 pour chaque arbre de gène. Nous avons testé les 3 stratégies d'optimisation, MC, RF et DB, présentées dans la section précédente pour mesurer le taux de détection des transferts générés (i.e. *Detection rate* sur la Figure 2a) et le pourcentage des cas quand le nombre exact des transferts générés a été retrouvé (*Same number of HGTs* sur la Figure 2b). Remarquons qu'un transfert correct est celui dont l'emplacement et la direction exacts ont été retrouvés. Pour les deux critères considérés (Figure 2a et b), la stratégie algorithmique basée sur la dissimilarité de bipartition était plus performante que les stratégies basées sur les moindres carrés et la métrique de Robinson et Foulds. Les deux dernières stratégies avaient les tendances similaires, mais celle basée sur la distance RF a toujours fourni des meilleurs résultats que celle basée sur les moindres carrés.

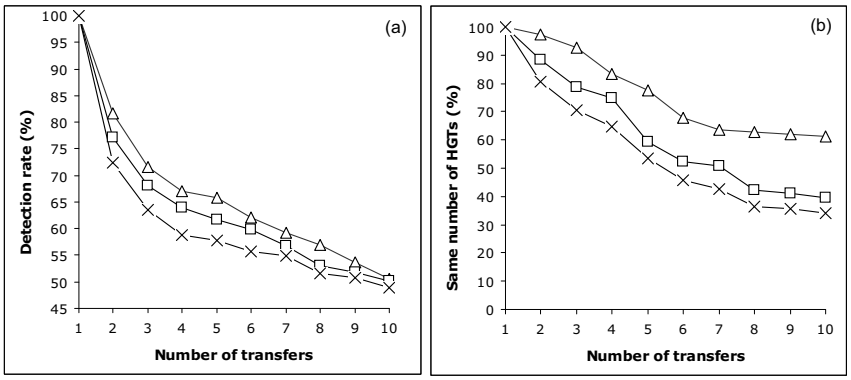


Fig. 2. Pourcentage des cas quand l'algorithme prédit : (a) les transferts corrects ; (b) le nombre exact des transferts - versus le nombre de transferts. Pour chaque point du graphique la moyenne des résultats obtenus pour 100 arbres à 10, 20, … et 100 feuilles est montrée. Les 3 stratégies comparées sont : la dissimilarité de bipartition (Δ), la distance de Robinson et Foulds (□) et les moindres carrés (×).

4 Références

[1] J.-P. Barthélemy, A. Guénoche, *Les arbres et les représentations des proximités*, Paris, Masson, 1988.

[2] P. Buneman, "The recovery of trees from measures of dissimilarity", *In Mathematics in the Archeological and Historical Sciences, Edinburgh University Press*, 1971, 387–395.

[3] W.F. Doolittle, "Phylogenetic classification and the universal tree", *Science* 284, 1999, 2124-2129.

[4] M. Hallett, J. Lagergren, "Efficient algorithms for lateral gene transfer problems", *In: El-Mabrouk,N., Lengauer,T., Sankoff,D. (eds.): RECOMB, ACM, New-York* 2001, 149-156.

[5] J. Hein, T. Jiang, L. Wang, K. Zhang, "On the complexity of comparing evolutionary trees", *Discr. Appl. Math.* 71, 1996, 153-169.

[6] M. Kuhner, J. Felsenstein, "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates", *Mol. Biol. Evol.* 11, 1994, 459-468.

[7] J.G. Lawrence, H. Ochman, "Amelioration of bacterial genomes: rates of change and exchange", *J. Mol. Evol.* 44, 1997, 383-397.

[8] W.P. Maddison, "Gene trees in species trees", *Syst. Biol.* 46, 1997, 523-536.

[9] V. Makarenkov, A. Boc, C. F. Delwiche, A.B. Diallo, H. Philippe, "New efficient algorithm for modeling partial and complete gene transfer scenarios", *Data Science and Classification, V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Ziberna (Eds.), Series: Studies in Classification, Data Analysis, and Knowledge Organization, Springer,* 2006, 341-349.

[10] B. Mirkin, T. I. Fenner, M. Galperin, E. Koonin, "Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of HGT in the evolution of prokaryotes", *BMC Evol. Biol.* 3, 2003.

[11] B.M.E. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, R. Timme, "Phylogenetic networks: modeling, reconstructibility, and accuracy", *IEEE/ACM Trans. on Comput. Biol. and Bioinf.* 1, 2004, 13-23.

[12] R.D.M. Page, M.A. Charleston, "Trees within trees: phylogeny and historical associations", *Trends Ecol. Evol*. 13, 1998, 356-359.

[13] D.R. Robinson, L.R. Foulds, "Comparison of phylogenetic trees", *Math. Biosc.* 53, 1981, 131-147.

Règles d'association dans un contexte de descriptions ordonnées

J. Diatta[1], H. Ralambondrainy[1], et A. Totohasina[2]

1. IREMIA, Université de la Réunion
15 av Cassin, 97715 Saint-Denis Message Cedex 9, Réunion
2.Département de Mathématiques et Informatique
ENSET, Université d'Antsiranana -B.P. 0
Antsiranana 201 Madagascar
totohasina@yahoo.fr, (ralambon,jdiatta)@univ-reunion.fr

Mots clés : règles d'association, Analyse de données symboliques, Classification

1 Introduction

Les règles d'association sont parmi les méthodes d'Analyse de Données les plus populaires ([1] [2]). Une règle d'association (RA) est une implication $X \to Y$ qui exprime un certain lien entre des attributs binaires. Dans cette communication, nous proposons une approche, fondée sur la classification, pour rechercher des règles d'association dans un contexte de descriptions ordonnées, c'est-à-dire un contexte où l'espace de descriptions des objets est un ensemble (partiellement) ordonné. L'idée est de classifier l'ensemble des objets, puis de considérer comme candidat prémisse ou conséquent d'une règle d'asociation l'intension d'une des classes resultant de cette classification. Ainsi, un espace de recherche de règles d'association valides est entièrement déterminé par les classes obtenues, et variera selon la mesure de dissimilarité utilisée, la méthodes de classification adoptée, ou la structure de classification construite. L'association des règles à des classes optimisant un critère est un facteur de pertinence qui renforcerait la qualité de ces règles, évaluée par ailleurs en utilisant une ou plusieurs des mesures de qualité proposées dans la littérature [3].

2 Règles d'association dans un contexte binaire

Un *contexte binaire* de fouille de données est une relation binaire $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ de graphe $\mathcal{R}$, entre un ensemble fini non vide d'objets $\mathcal{O}$ et un ensemble fini non vide d'attributs $\mathcal{A}$. Parfois on assimile la relation binaire $\mathbb{K}$ à son graphe $\mathcal{R}$. L'appartenance d'un couple $(o, a) \in \mathcal{O} \times \mathcal{A}$ au graphe $\mathcal{R}$ de la relation $\mathbb{K}$ exprime le fait que l'objet o possède l'attribut a. Les éléments de $\mathcal{A}$ sont souvent appelés *items* et les parties de $\mathcal{A}$ *motifs*.

Une *règle d'association* dans un contexte binaire $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ est un couple (X, Y) de motifs. Comme son nom l'indique, elle exprime une association ou un lien (orienté) entre X et Y et est notée $X \to Y$. Ainsi, le motif X est appelé la *prémisse* de la règle $X \to Y$ et Y le *conséquent*.

Remarque 1 *En général, on requiert d'une part que le conséquent d'une RA ne soit pas vide et, d'autre part, que la prémisse et le conséquent soient disjoints, pour ne pas considérer des règles triviales qui n'apportent à l'utilisateur aucune information utile.*

Toutesfois, malgré ces restrictions, le nombre de RA conformes à la définition ci-dessus reste très élevé. Ainsi, dans un soucis d'informativité et de pertinence, on en retient que ceux qui sont valides au sens d'une (un ensemble de) mesure(s) de qualité. La plupart de ces mesures de qualité sont probabilistes (*i.e.* se définissent entièrement à partir d'un tableau de contingence) et les plus connues d'entre elles sont le support, et la confiance définis respectivement par :

$$\text{Supp}(X \rightarrow Y) := \text{Supp}(X \cup Y) = p(X' \cap Y') \text{ et } \text{Conf}(X \rightarrow Y) = \frac{\text{Supp}(X \cup Y)}{\text{Supp}(X)} := p(Y'|X'),$$

où $Z' = \{o \in \mathcal{O} : \forall a \in Z[(o,a) \in \mathcal{R}]\}$ est l'extension du motif Z, $p(Z') = \frac{|Z'|}{|\mathcal{O}|}$, et $p(Y'|X')$ est la probabilité conditionnelle de Y' sachant X'.

3 Règles d'association dans un contexte de descriptions ordonnées

Un contexte binaire peut être vu comme un triplet $\mathbb{K} = (\mathcal{O}, \mathcal{P}(\mathcal{A}), \delta)$ où $\mathcal{P}(\mathcal{A})$ est l'ensemble ordonné $(\mathcal{P}(\mathcal{A}), \subseteq)$ et δ l'application qui associe à chaque objet o le motif constitué des items que possède l'objet o. Ainsi, pour chaque motif X, l'extension X' de X sera définie par $X' = \{o \in \mathcal{O} : X \subseteq \delta(o)\}$.

Cette représentation des contextes binaires permets de considérer naturellement les règles d'association dans un *contexte de descriptions ordonnées*, *i.e.*, un contexte $\mathbb{K} = (\mathcal{O}, \mathcal{D}, \delta)$ où $\mathcal{D} := (\mathcal{D}, \leq)$ est un ensemble ordonné et δ est une application qui associe à chaque objet o sa description $\delta(o)$ dans $\mathcal{D}$. L'ensemble $\mathcal{D}$ est alors appelé l'espace de description des objets. Les motifs d'un tel contexte seront des éléments de $\mathcal{D}$ et l'extension X' d'un motif $X \in \mathcal{D}$ sera définie par $X' = \{o \in \mathcal{O} : X \leq \delta(o)\}$ comme cela est défini pour des objets symboliques [4].

Une règle d'association dans un contexte de descriptions ordonnées $\mathbb{K} = (\mathcal{O}, \mathcal{D}, \delta)$ est un couple de motifs $(X, Y) \in \mathcal{D}^2$, que l'on notera également $X \rightarrow Y$.

La plupart des ensembles de données à traiter peuvent être représentés comme des contextes de descriptions ordonnées $(\mathcal{O}, \mathcal{D}, \delta)$ tels que la borne inférieure de deux éléments quelconques de $\mathcal{D}$ existe dans $\mathcal{D}$. Dans ce cas, on désignera par $*$ la borne inférieure de l'ensemble des descriptions de tous les objets, *i.e.*, $* = \inf\{\delta(o) : o \in \mathcal{O}\}$. Les restrictions de la remarque 1 s'expriment alors respectivement par : le conséquent n'est pas inférieur ou égal à $*$ et la borne inférieure de la prémisse et du conséquent est égale à $*$.

4 Recherche de règles d'association

Soit $\mathbb{K} = (\mathcal{O}, \mathcal{D}, \delta)$ un contexte de descriptions ordonnées tel que la borne inférieure de deux descriptions $\delta(o_1)$ et $\delta(o_2)$ existe toujours dans $\mathcal{D}$. L'idée est de spécifier un espace de recherche de règles d'association dans $\mathbb{K}$ en classifiant l'ensemble $\mathcal{O}$ des objets du contexte. Plus précisément, il s'agit d'associer à un système $\mathcal{C}$ de classes dans $\mathcal{O}$, l'ensemble $R_{\mathcal{C}}$ de candidats prémisses ou conséquents de règles, défini par

$$R_{\mathcal{C}} = \{\text{int}(C) : C \in \mathcal{C}\}$$

où $\text{int}(C) = \inf\{\delta(x) \in \mathcal{D} : x \in C\}$ est l'intension de C. La qualité des règles d'association formées de couples d'éléments de $R_{\mathcal{C}}$ est alors évaluée en utilisant une ou plusieurs des diverses mesures de qualité de règles proposées dans la littérature.

[1] AGRAWAL, R., IMIALINSKI, T., SWAMI, A. (1993): Mining association rules between sets of items in large databases. In: P. Buneman and S. Jajodia (Eds.): *ACM SIGMOD International Conference on Management of Data*. ACM press,Washington,207–216.

[2] PASQUIER, N.,BASTIDE,Y.,TAOUIL,R.,LAKHAL, L.,(2000): Efficient mining of association rules using closed itemset lattices. *Information Systems 24,25–46*

[3] DIATTA,J., RALAMBONDRAINY,H., TOTOHASINA,A., (2007): Towards a unifying probabilistic implicative normalized quality measure for association rules. *Book Series Studies in Computational Intelligence* Springer Berlin/Heidelberg 43, 237-250.

[4] DIDAY E., (1995): Probabilistic, Possibilist and Belief Objects for Knowledge Analysis. *Annals of Operations Research, 55, 227–276.*

Nonnegative matrix factorization algorithms: Tweedie quasi-likelihoods approach

Simplice Dossou-Gbété

April 2007

Université de Pau et des Pays de l'Adour
Laboratoire de Mathématiques et leurs Applications UMR-CNRS 5546
Av. de l'université, B.P. 576, 64012 Pau
simplice.dossou-gbete@univ-pau.fr

Mots clés : nonnegative matrix factorization, Tweedie families of distributions, quasi-likelihood, algorithm

1 Introduction

Let $Y = [y_{ij}]_{i=1:n}^{j=1:m}$ denote a matrix which entries are records of positive measurements or counts. Each column $(y_{ij})_{i=1:n}$ of the data matrix Y can be understood as a spectra, that is the resultant plot of a detected signal versus wavelength. Spectra are often used in colormetric to identify the components of a color sample. One may assume that each spectra is the mixture of some hidden components and that this mixture is additive. Since the recorded measurements or counts are noise corrupted, it seems natural to consider, in first approximation, that data are independent realizations of distributions with positive support and respective means $\mu_{ij} > 0$.

The non negative matrix factorization relies upon the modelling of μ_{ij} as

$$\mu_{ij}(w, \beta) = \sum_{k=1}^{r} w_{ik}\beta_{jk} \tag{1.1}$$

where $\beta = [\beta_{ik}]_{i=1}^{k=1:r}$ and $w = [w_{jk}]_{j=1:m}^{k=1:r}$ are matrixes which entries are unknown positive parameters. The sequences $(\beta_{ik})_{i=1:n}$ can be understood as plots of constituent spectra versus a n regular-spaced wavelength values while $(w_{jk})_{k=1:r}$ gathers the weights of the r constituent signals in the mixture resulting in the detected spectra $(y_{ij})_{i=1:n}$. The statistical model (1.1) is a fixed effects model for an unsupervised statistical learning, since the sources separation is done with no prior knowledge about the constituent signals pattern.

The statistical learning of the model (1.1) is a part of the toolkit for the blind sources separation and has deserved attention in the recent past. Setting a suitable probabilistic framework is essential for such a statistical learning task. However, a complete knowledge of the distributions underlying the data generation is often not available. One of the currently used approachs to overcome this lack of prior knowledge is to rely the statistical learning on the optimization of cost functions as quasi-likelihoods that do not requier strong distributional assumptions. Very often the choice of the cost function to be optimized is a trade-off between the nature of the problem on hand, the available statistical methods including algorithms and the computational ressources.

We intend to investigate in this paper the minimization of a class of quasi-likelihoods resulting from the assumption that the distributions variances are proportionnal to some specified power function of the corresponding means. This property is shared by the distributions that belongs to a Tweedie family that is a special class of exponential dispersion family of distributions [4]. Such quasi-likelihoods are related to Bregman distances as it is noticed. In this framework, auxiliary functions prove to be a convenient gateway on the route of designing algorithms, efficient and easy to implement, to solve the optimization of the quasi-likelihoods with respect to the positivity constraint on the model parameters.

2 Tweedie models and related algorithms

Let $(y_{ij})_{i=1:n}^{j=1:m}$ denote a matrix of $n \times m$ independent realizations of distributions with respective means μ_{ij} and variances σ_{ij}^2. Assuming that data $y_{ij},\ i = 1:n,\ j = 1:m$ are positive numbers, the aim is to analyse them through the following modelization of the means $\mu_i > 0$

$$\mu_{ij}(\beta) = \sum_{k=1}^{r} w_{jk}\beta_{ik} \tag{2.2}$$

where $(w_{jk})_{j=1:m},\ k = 1:r$ are fixed vectors of real positive numbers and $\beta_k = (\beta_{ik})_{i=1:n} \in \mathbb{R}^n$ are vectors of positive unknown parameters.

Tweedie class of distributions provides suitable probabilistic framework to handle the statistical analysis of data for which the scale of values is the positive half line. One of the main property of Tweedie distributions is that their variances are specified power function of their means. The support of a Tweedie distribution is the positive half-line if the power of the variance function is one or greater. The distributions belonging in Tweedie class are a special case of the reproductive exponential families and the former prove to be limiting distributions for many distributions belonging in an exponential family when the mean is too small or too large [4]. Tweedie class of distributions include gaussian family and the Poisson family.

Taking into account the considerations above, the main distributional assumptions upon which this work is based is that the variances of the distribution a specified power function of the means. Among the more attractive algorithms proposed to tackle the statistical learning of such a model are the multiplicative update rules for the minimization of the least-squares and Poisson quasi-likelihood. Our aim is to provide such multiplicative update rules when considering the class of Tweedie quasi-likelihoods which includes the least-squares and Poisson quasi-likelihood.

2.1 Tweedie class of quasi-likelihood

Assume that $y_{ij}, i = 1:n, j = 1:m$ are independent realizations of random variables with means $\mu_{ij} > 0$ and variance $\sigma_{ij}^2 = \phi_j \mu_{ij}^p$, where the positive dispersion parameter ϕ_j and the power $p \in \{0\} \bigcup [1, +\infty[$ are known. By considering the model (2.2), Tweedie class of quasi-likelihoods is defined as follows

$$Q(\beta) = \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{1}{\phi_j}\left[\frac{y_{ij}^{2-p}}{(1-p)(2-p)} - \frac{y_{ij}\left[\mu_{ij}(\beta)\right]^{1-p}}{1-p} + \frac{\left[\mu_{ij}(\beta)\right]^{2-p}}{2-p}\right]$$

$$\frac{y_{ij}^{2-p}}{(1-p)(2-p)} - \frac{y_{ij}\left[\mu_{ij}(\beta)\right]^{1-p}}{1-p} + \frac{\left[\mu_{ij}(\beta)\right]^{2-p}}{2-p}$$

is the unit deviance of data when the statistical model underlying the data analysis is based on a Tweedie family of distributions. Hence

$$\frac{y_{i}^{2-p}}{(1-p)(2-p)} - \frac{y_{i}\left[\mu_{i}(\beta)\right]^{1-p}}{1-p} + \frac{\left[\mu_{i}(\beta)\right]^{2-p}}{2-p}$$

is the Bregman distance between y_{ij} and $\mu_{ij}(\beta)$ with respect to the convex function , G say, such that $G(x) = \frac{x^{2-p}}{(1-p)(2-p)}$. Therefore, one can formulate a minimization algorithm for Tweedie quasi-likelihoods by using a multiplicative update rule as follows:

2.2 Algorithm for Tweedie quasi-likelihood minimization

i. pick a starting value $\beta^{(0)}$for the the parameters vector

ii. repeat until convergence of the quasi-likelihood within a numerical tolerance

$$\beta_{ik}^{(t+1)} = \beta_{ik}^{(t)} \left[\frac{\sum_{j=1}^{m} \frac{y_{ij}}{\left[\mu_{ij}\left(\beta^{(t)}\right)\right]^{p}} \frac{w_{jk}}{\phi_j}}{\sum_{j=1}^{m} \left[\mu_{ij}\left(\beta^{(t)}\right)\right]^{1-p} \frac{w_{jk}}{\phi_j}} \right]^{\frac{1}{p}}$$

3 Alternating minimization for nonnegative matrix factorization

Assuming that data are generated by distributions for which the means μ_{ij} and the variances σ_{ij}^2 are related through a power function as $\sigma_{ij}^2 = \phi_j \mu_{ij}^p$ and the means are modelled as (1.1), the quasi-likelihood of the data with respect to the parameters matrixes $w = [w_{jk}]_{j=1:m}^{k=1:r}$ and $\beta = [\beta_{ik}]_{i=1:n}^{k=0:r}$ is

$$Q(w,\beta) = \begin{cases} \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{1}{\phi_j}\left(y_{ij} - \mu_{ij}(w,\beta)\right)^2 \text{ if } p = 0 \\ \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{1}{\phi_j}\left[\frac{y_{ij}^{2-p}}{(1-p)(2-p)} - \frac{y_{ij}[\mu_{ij}(w,\beta)]^{1-p}}{1-p} + \frac{[\mu_{ij}(w,\beta)]^{2-p}}{2-p}\right] \text{ if } p > 1,\ p \neq 2 \\ \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{1}{\phi_j}\left[-y_{ij}\log\left(\frac{\mu_{ij}(w,\beta)}{y_{ij}}\right) + \mu_{ij}(w,\beta) - y_{ij}\right] \text{ if } p = 1 \\ \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{1}{\phi_j}\left[-1 + \frac{y_{ij}}{\mu_{ij}(w,\beta)} + \log\left(\frac{\mu_{ij}(w,\beta)}{y_{ij}}\right)\right] \text{ if } p = 2 \end{cases}$$

The quasi-likelihood function Q is convex in each variable when holding the second variable fixed. One possible method to obtain a minimizing pair of parameters w and β is to go through an alternating minimization scheme which consists in circling the two following steps:

(i) minimize Q with respect to β when ψ is hold fixed

(ii) minimize Q with respect to ψ when β is hold fixed at the optimum value obtained at the previous step.

References

[1] Della Pietra V. et al.: Induce features of random fields. IEEE Transactions on pattern recognition and machine intelligence. 19-4, pp.1-13, (1997)

[2] Guillamet, D. et al.: Color histogram classification using NMF. Computer Visison Center Technical report #057, (2001).

[3] Guillaumet D. et al.: Analysing non-negative matrix factorization for image classification. proceedings of 16th International Conference on pattern recognition. (2) pp.116-119, Quebec (2002)

[4] Jörgensen,B.: Theory of dispersion models. Chapman & Hall, (1997).

[5] Lee D. & Seung H.: Algorithm for non-negative matrix factorization. Advances in neural information processing system, 13, pp.556-562, MIT press, (2001).

[6] Lee D. & Seung H. Learning the parts of objects by non-negative matrix factorization. Nature, 401, pp 788-791, (1999).

[7] Sajda et al.: Recovery of constituent spectra in 3D chemical shift imaging using non-negative matrix factorization. 4th International Symposium on idependant Components Analysis and Blind Source Separation (ICA2003), pp.71-76 (2003)

Génération de bases pour les règles d'association M_{GK}-valides

D. Feno[1,2], J. Diatta[1] and A. Totohasina[2]

1. IREMIA-Université de La Réunion, 15, Avenue René Cassin, 97715 St-Denis cedex 9
2. ENSET, Université d'Antsiranana, Antsiranana 201 Madagascar
(fenodaniel2,totohasina)@yahoo.fr,(drfeno, j.diatta)@univ-reunion.fr

Mots clés : Base, Mesure de qualité, Règle d'association négative

1 Introduction

Les règles d'association révèlent des liens pertinents et non triviaux entre les attributs dans de grande base de données. Cela contribue probablement à faire l'extraction des règles d'association l'une des techniques les plus populaires de la fouille de données [1]. La validité des règles d'association est évaluée par une (ou plusieurs) mesure(s) de qualité. Toutefois, le nombre des règles extraites est souvent élevé du fait, entre autres, de la présence des règles rédondantes relativement à un ensemble d'axiomes d'inférence. Ainsi, plusieurs auteurs ont proposé des bases (familles minimales génératrices) de règles pour faire face à ce problème ([4], [13]). Notons que la mesure Confiance [1], souvent utilisée par les différentes méthodes de la fouille des règles d'association, autorisent certaines règles non pertinentes telles que la prémisse et le conséquent sont indépendants [7]. Le présent travail concerne la fouille des règles d'association au sens de la mesure de qualité $\mathrm{M_{GK}}$ [6]. Contrairement à la Confiance, $\mathrm{M_{GK}}$ ne séléctionne que des règles telles que la prémisse et le conséquent sont positivement dépendants. Par ailleurs, $\mathrm{M_{GK}}$ vérifie les trois principes de Piatetsky-Shapiro [12] tout en reflétant les situations de référence en cas de l'implication totale et l'incompatiblité entre la prémisse et le conséquent, ce qui n'est pas le cas pour la mesure de Piatetsky-Shapiro [12]. Dans ce papier, nous proposons des algorithmes pour générer des bases que nous avons caractérisées dans [3] pour les règles d'association valides au sens de la mesure de qualité $\mathrm{M_{GK}}$ [6].

2 Règles d'association $\mathrm{M_{GK}}$-valides

Nous nous plaçons dans le cadre d'un contexte binaire $\mathbb{K} = (\mathcal{E}, \mathcal{V})$, où $\mathcal{E}$ est un ensemble fini d'entités et $\mathcal{V}$ un ensemble fini de variables binaires définies sur $\mathcal{E}$. Les sous ensembles de $\mathcal{V}$ seront appelés *motifs*. L'extension d'un motif X est l'ensemble noté X' des entités x telles que pour tout $v \in X, v(x) = 1$. A tout motif X sera associée sa négation $\overline{X}$ telle que pour $x \in \mathcal{E}$, $\overline{X}(x) = 1$ si et seulement si il existe $v \in X$ tel que $v(x) = 0$. Nous considérerons ces négations de motifs comme des motifs que nous qualifierons de négatifs pour les distinguer de ceux dont ils sont la négation et qui seront ainsi qualifiés de positifs. On notera que $\overline{X}' = \mathcal{E} \setminus X'$.

Une *règle d'association* (RA) est un couple (X, Y) noté $X \rightarrow Y$, où X et Y sont des motifs positifs ou négatifs. Le motif X est appelé la *prémisse* de $X \rightarrow Y$ et Y son *conséquent*. Ainsi, quatre types de RAs peuvent être obtenus à partir de deux motifs positifs X et Y : (a) une RA dite *positive*, de la forme $X \rightarrow Y$ ou $Y \rightarrow X$; (b) une RA dite *négative à droite*,

de la forme $X \rightarrow \overline{Y}$ ou $Y \rightarrow \overline{X}$; (c) une RA dite *négative à gauche*, de la forme $\overline{X} \rightarrow Y$ ou $\overline{Y} \rightarrow X$; (d) une RA dite *bilatéralement négative*, de la forme $\overline{X} \rightarrow \overline{Y}$ ou $\overline{Y} \rightarrow \overline{X}$.

La validité des règles d'association est évaluée par une (ou plusieurs) mesure(s) de qualité. Une *mesure de qualité* des RAs de $\mathbb{K}$ est une application μ à valeurs réelles, définie sur l'ensemble des RAs de $\mathbb{K}$. Les plus connues des mesures de qualité sont le *Support* et la *Confiance* [1]. Pour un ensemble A, désignons par $|A|$ la cardinalité de A. Le support d'un motif X est défini par $\text{supp}(X) = \frac{|X'|}{|\mathcal{E}|}$. Le Support de $X \rightarrow Y$ noté $\text{supp}(X \rightarrow Y)$ est défini par : $\text{supp}(X \rightarrow Y) = \text{supp}(X \cup Y)$. La Confiance de $X \rightarrow Y$ est définie par : $\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \rightarrow Y)}{\text{supp}(X)}$. La mesure de qualité M_{GK} à laquelle nous nous intéressons dans ce papier a été introduite dans [6] et est définie par :

$$\text{M}_{\text{GK}}(X \rightarrow Y) = \begin{cases} \frac{\text{conf}(X \rightarrow Y) - \text{supp}(Y)}{1 - \text{supp}(Y)} & si\ \text{conf}(X \rightarrow Y) \geq \text{supp}(Y) \\ \frac{\text{conf}(X \rightarrow Y) - \text{supp}(Y)}{\text{supp}(Y)} & si\ \text{conf}(X \rightarrow Y) \leq \text{supp}(Y). \end{cases}$$

Elle prend ses valeurs sur l'intervalle $[-1, 1]$ et réflète les situations de référence telles que l'incompatibilité, la dépendance négative, l'indépendance, la dépendance positive et l'implication logique entre la prémisse et le conséquent d'une RA. En effet, $M_{GK}(X \rightarrow Y) = -1$ si et seulement si X et Y sont incompatibles, $-1 < M_{GK}(X \rightarrow Y) < 0$ si et seulement si X défavorise Y ou encore si et seulement si X et Y sont négativement dépendants, $M_{GK}(X \rightarrow Y) = 0$ si et seulement si X et Y sont indépendants, $0 < M_{GK}(X \rightarrow Y) < 1$ si et seulement si X favorise Y ou X et Y sont positivement dépendants, $M_{GK}(X \rightarrow Y) = 1$ si et seulement si X implique logiquement Y.

Soit $\alpha \in [0, 1]$. Une RA $X \rightarrow Y$ sera dite (M_{GK}, α)-*valide* (ou tout simplement M_{GK}-*valide* ou *valide*) si $M_{GK}(X \rightarrow Y) \geq \alpha$. Les propriétés de M_{GK} permettent de se restreindre aux RAs positives ou négatives à droite. Ainsi, nous distinguons deux classes de RAs (M_{GK}, α)-valides : (a) les RAs M_{GK}-exactes, *i.e.*, telles que $M_{GK}(X \rightarrow Y) = 1$; (b) les RAs M_{GK}-approximatives, *i.e.*, telles $\alpha \leq M_{GK}(X \rightarrow Y) < 1$, où X est un motif positif et Y un motif positif ou négatif.

3 Génération des bases

3.1 Règles positives exactes

Les RAs positives M_{GK}-exactes sont caractérisées par :
$M_{GK}(X \rightarrow Y) = 1 \Leftrightarrow conf(X \rightarrow Y) = 1$. L'ensemble des RAs positives M_{GK}-exactes est identique à celui des RAs Confiance-exactes. Ainsi, la base de Guigues-Duquenne [5] pour les RAs Confiance-exactes est une base pour les RAs positives M_{GK}-exactes. Cette base est caractérisée en fonction de l'opérateur de fermeture $\varphi = g \circ f$, où f et g sont définies par $f : \mathcal{P}(\mathcal{E}) \rightarrow \mathcal{P}(\mathcal{V})$ avec $f(E) = \{v \in \mathcal{V} : v(x) = 1, \forall x \in E\}$, et $g : \mathcal{P}(\mathcal{V}) \rightarrow \mathcal{P}(\mathcal{E})$, avec $g(X) = X'$. Un motif positif X tel que $\varphi(X) = X$ est dit φ-*fermé* ou tout simplement *fermé*. Un algorithme de construction de cette base des RAs peut être trouvé dans [13].

3.2 Règles négatives exactes

Notons que les RAs négatives M_{GK}-exactes sont caractérisées par :
$M_{GK}(X \rightarrow \overline{Y}) = 1 \Leftrightarrow \text{supp}(X \rightarrow Y) = 0$.
Cette caractérisation nous conduit à considérer la bordure positive de l'ensemble des motifs de support non nul $Bd^+(0)$ [10] définie par : $Bd^+(0) = \{X \subseteq \mathcal{V} : supp(X) >$

0 *et* $\forall\ x \notin X, supp(X \cup \{x\}) = 0\}$. C'est l'ensemble des motifs maximaux de support non nul. On peut trouver dans la littérature des algorithmes de génération des motifs maximaux fréquents ([2], [8]). Dans [3], nous proposons la base $BNE = \{X \rightarrow \{\overline{x}\} : X \in Bd^+(0)$ *et* $x \notin X\}$, pour les RAs négatives M_{GK}-exactes, relativement aux axiomes d'inférence (NE1) et (NE2) définis par : (NE1) si $X \rightarrow \overline{Y}$, alors pour tout motif T tel que $supp(Y \cup T) > 0$, on a $X \rightarrow \overline{Y \cup T}$; (NE2) si $X \rightarrow \overline{Y}$ alors pour tout $Z \subset X$ tel que $\text{supp}(X \cup Z) = 0$ on a $Z \rightarrow \overline{Y}$.

Les algorithmes, que nous proposons dans ce papier, supposent que les fermés sont déjà calculés. L'algorithme 1 génère la base BNE.

Algorithme 1 : Base Négative Exacte
Entrée : Bordure positive $Bd^+(0)$;
Sortie : BNE ;

1. Initialiser BNE l'ensemble à l'ensemble vide ;
2. Pour tout $X \in Bd^+(0)$ et pour tout $x \notin X$ insérer la RA $X \rightarrow \overline{x}$ dans BNE.

3.3 Règles positives approximatives

Notons que les RAs positives M_{GK}-approximatives sont caractérisées par
$\alpha \leq M_{GK}(X \rightarrow Y) < 1 \Leftrightarrow \text{supp}(Y)(1-\alpha) + \alpha \leq conf(X \rightarrow Y) < 1$.
Soit $PA(\alpha) = \{X \rightarrow Y : \alpha \leq M_{GK}(X \rightarrow Y) < 1\}$ l'ensemble de toutes les RAs positives M_{GK}-approximatives. Soient $0 = \alpha_0 < \alpha_1 < ... < \alpha_p = 1$ les différentes valeurs possibles de support des motifs. Considérons $PA_i(\alpha) = \{X \rightarrow Y : \alpha_i(1-\alpha) + \alpha \leq conf(X \rightarrow Y) < \alpha_{i+1}(1-\alpha)+\alpha\}$. Alors, on montre que $PA(\alpha)$ est la réunion disjointe des $PA_i(\alpha)$. De plus, si $BPA_i(\alpha)$ est la base de Luxenburger [9] de $PA_i(\alpha)$ alors $BPA(\alpha) = \bigcup_{i=0}^{p} BPA_i(\alpha)$ est une base pour les RAs positives M_{GK}-approximatives. L'algorithme ci-dessous construit $BPA(\alpha)$. **Algorithme 2 : Base positive Approximative**
Entrée : Ensemble de tous les motifs fermés ;
Sortie : BPA ;

1. Initialiser $BPA(\alpha)$ à l'ensemble vide ;
2. Pour tout motif fermé Y chercher les fermés prédécesseurs de Y ;
3. Pour tout X prédécesseur de Y calculer $\text{conf}(X \rightarrow Y)$;
4. Si $\text{supp}(Y)(1-\alpha) + \alpha \leq \text{conf}(X \rightarrow Y)$, alors insérer $X \rightarrow Y$ dans $BPA(\alpha)$.

3.4 Règles négatives approximatives

Notons que les RAs négatives M_{GK}-approximatives sont caractérisées par
$\alpha \leq M_{GK}(X \rightarrow \overline{Y}) < 1 \Leftrightarrow 0 < conf(X \rightarrow Y) \leq supp(Y)(1-\alpha)$.
Dans [3], nous proposons la base $BNA(\alpha)$, définie par : $BNA(\alpha) = \{X \rightarrow \overline{Y} : \varphi(X) = X,\ et\ \varphi(Y) = Y, 0 < conf(X \rightarrow Y) \leq supp(Y)(1-\alpha)\}$ pour les RAs négatives M_{GK}-approximatives valides, relativement à l'axiome d'inférence (NA) : Si $X \rightarrow \overline{Y}$, alors $\forall\ Z, T$ tels que $\varphi(X) = \varphi(Z)$ et $\varphi(Y) = \varphi(T)$, on a $Z \rightarrow \overline{T}$.

Remarquons que si X et Y sont deux motifs comparables, alors X favorise Y et réciproquement. Cela permet, pour un motif fermé X, de restreindre l'espace de recherche de ses conséquents potentiels aux fermés qui lui sont incomparables. L'algorithme ci-dessous construit $BNA(\alpha)$.

Algorithme 3 : Base Négative Approximative
Entrée : Ensemble des fermés, α ;
Sortie : $BNA(\alpha)$;

1. Initialiser $BNA(\alpha)$ à l'ensemble vide ;
2. Pour tout motif fermé X, trouver les fermés incomparables X ;
3. Pour chaque fermé Y incomparable à X, calculer $\text{conf}(X \rightarrow Y)$;

4. Si $0 < \text{conf}(X \rightarrow Y) \leq \text{supp}(X \rightarrow Y)(1-\alpha)$, alors insérer $X \rightarrow \overline{Y}$, $Y \rightarrow \overline{X}$ dans $BNA(\alpha)$.

Le tableau ci-dessous donne le nombre des règles dans les bases positives exacte et approximative sur les données *Mushroom* (8124 entités, 120 attributs et en moyenne 23 attributs par entités). Les programmes de génération des autres bases sont en cours de développement. Ne disposant pas encore du programme calculant le nombre total des règles M_{GK}-approximatives, le nombre de règles Confiance-approximatives est donné à titre indicatif.

minsupp	α	BPA	Règles Confiance-Approximatives	BPE	Règles Exactes
50%	0.5	46	928	52	220
	0.6	42	658		
	0.7	37	447		
	0.8	29	413		
	0.9	20	250		
40%	0.7	163	2889	170	939
	0.8	128	2363		
	0.9	93	1465		

Tableau 1. Illustration des bases de règles positives sur les données *Mushroom.*

[1] R. Agrawal,T. Imielinski, A. Swami, "Mining association rules between sets of items in large databases", *Proc. of the Int. Conf. on Management of Data*,1993, 207-216.

[2] R. J. Bayardo, "Efficiently mining long patterns from databases", *Proc. of the ACM SIGMOD Conference*, 1998, 85-93.

[3] D. R. Feno, J. Diatta, A. Totohasina, "Une base pour les règles d'association d'un contexte binaire valides au sens de la mesure de qualité M_{GK}", *Proc. of the 13ème Rencontre de la SFC*, 2006, 105-109.

[4] M. J. Zaki, "Generating non-redundant association rules", *Proc of the Int. Conf. Knowledge Discovery and Data Mining*, 2000,34-43

[5] J. L. Guigues, V. Duquenne, "Famille non redondante d'implications informatives résultant d'un tableau de données binaires", *Math. et Sci. hum* 95, 1986, 5-18

[6] S. Guillaume, *Traitement des données volumineuses. Mesures et algorithmes d'extraction des règles d'association et règles ordinales*, Thèse de Doctorat, Universté de Nantes, France, 2000.

[7] S. Lallich and O. Teytaud, "Evaluation et validation de mesures d'intérêt des règles d'association", *RNTI-E-1*, 2004, 193–217

[8] Dao-I Lin, Zvi M. Kedem, "Pincer Search: A New Algorithm for Discovering the Maximum Frequent Set", *Lecture Notes in Computer Science* 1377, 1998, 105-121.

[9] M. Luxenburger, "Implications partielles dans un contexte", *Math. Inf. Sci. hum* 113, 1991, 35–55.

[10] H. Mannila, H. Toivonen,"Levelwise search and borders of theories in knowledge discovery", *Data Mining Knowledge Discovery* 1.3, 1997, 241-258.

[11] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal,"Efficient mining of association rules using closed itemset lattices", *Information Systems* 24, 1999, 25-46.

[12] G. Piatetsky-Shapiro,"Discovery, analysis, and representation of strong rules", *Knowlegde Discovery in Databases*, 1991, 229–248,

[13] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier and L. Lakhal, "Intelligent Structuring and Reducing of Association Rules with Formal Concept Analysis", *Advances in Artificial Intelligence* LNAI 2174, 2001, 335-350.

Risque structurel et sélection d'instances pour la règle du plus proche voisin

S. Ferrandiz

ENST, 46, rue Barrault, 75634 Paris cedex 13
sylvain.ferrandiz@enst.fr

Résumé

La sélection d'instances pour la classification suivant le plus proche voisin constitue un problème d'apprentissage pour lequel l'ensemble des hypothèses dépend des données. Nous nous plaçons dans le cadre de l'apprentissage statistique et proposons un risque structurel adapté à ce cas. Nous comparons le critère obtenu avec un critère bayésien et illustrons les différences à l'aide d'une expérience sur données synthétiques.

Mots clés : classification supervisée, plus proche voisin, sélection d'instances, risque structurel

1 La sélection d'instances

En classification supervisée, la règle de classification suivant le plus proche voisin [3] consiste à attribuer à un nouvel individu l'étiquette de l'individu le plus proche parmi ceux constituant l'échantillon. Sa mise en œuvre soulève, entre autres, la question des instances à conserver.

Soit $D = (x_n, y_n)$ un échantillon de taille N, où chaque x_n est un élément d'un espace métrique $(\mathbb{X}, \delta)$ et chaque y_n est un élément de $[\![1, J]\!]$ (l'ensemble des J étiquettes). L'ensemble $\mathcal{H}$ des hypothèses est l'ensemble des parties $H \subset \{x_n; 1 \leq n \leq N\}$. C'est un ensemble d'hypothèses dépendant des données.

Le critère d'évaluation de la qualité d'une hypothèse est à définir avec soin. En effet, la considération du seul risque empirique n'est pas suffisante, puisque la meilleure hypothèse suivant ce critère est nécessairement $\{x_n; 1 \leq n \leq N\}$ (toutes les instances sont conservées).

Nous nous attachons à définir un risque structurel pour le problème de la sélection, autrement dit à pénaliser le risque empirique. Dans la section 2, nous rappelons le travail de [1] sur la minimisation du risque structurel dans le cas de la dépendance aux données. Dans la section 3, nous tirons de ce travail un risque structurel pour la sélection d'instances. Dans le section 4, nous comparons le comportement du critère obtenu avec celui du critère bayésien introduit dans [2].

2 Risque structurel et hypothèses dépendantes des données

Dans [4], l'auteur propose de minimiser le risque empirique plus une borne sur l'erreur en généralisation : le risque structurel. Mais le cas considéré ici, celui d'hypothèses

dépendantes des données, n'entre pas dans ce cadre. C'est dans [1] que le cadre est adapté afin de proposer un principe de minimisation du risque structurel (abrégé en principe SRM) applicable à notre cas. Nous présentons brièvement ce travail. Celui-ci se place dans le cas binaire, i.e. $J = 2$.

Si $f : \mathbb{X} \to \{0, 1\}$ est un classifieur, le risque empirique $R_{emp}(f)$ de f est le nombre d'instances d'un échantillon D mal classifiées par f. On note $I(f) = \{x \in \mathbb{X};\ f(x) = 1\}$. Si $\mathcal{C}$ est un ensemble de classifieurs, on se donne une application associant à tout échantillon D un ensemble $\mathcal{C}(D)$ inclus dans $\mathcal{C}$ (les hypothèses dépendent des données).

Soit $D = (x_n, y_n)$ un échantillon de taille N. Si $M \leq N$, on note $\mathcal{N}_M(D)$ le cardinal :

$$\mathcal{N}_M(D) = \#\{D_M^{(x)} \bigcap I(f);\ D_M \subset D, \#D_M = M \text{ et } f \in \mathcal{C}(D_M)\}, \tag{1}$$

où $D_M^{(x)}$ est l'ensemble des x_n tels que $(x_n, y_n) \in D_M$. Intuitivement, le nombre $\mathcal{N}_M(D)$ est le nombre de partitions en deux groupes de l'échantillon D réalisées par les classifieurs définis à partir d'un échantillon de taille M inclus dans D.

Pour $M \leq N$, on définit le *coefficient de pulvérisation* $\mathcal{S}_{M/N}$ de $\mathcal{C}$ par la relation

$$\mathcal{S}_{M/N} = \sup_{D, \#D=N} \mathcal{N}_M(D). \tag{2}$$

Intuitivement, le coefficient de pulvérisation mesure le nombre maximal d'instances d'un échantillon de taille N que les classifieurs peuvent séparer et quantifie ainsi la capacité de discrimination d'un ensemble de classifieurs.

Supposons donnée une application de $\mathcal{C}$ dans $\mathbb{N}$. Pour un échantillon D et $K \in \mathbb{N}$, notons $\mathcal{C}_K(D)$ l'ensemble des éléments de $\mathcal{C}(D)$ dont l'image est K par cette application. Notons $\mathcal{C}_K$ l'union des $\mathcal{C}_K(D)$ et $\mathcal{S}_{M/N}^{(K)}$ le coefficient de pulvérisation de $\mathcal{C}_K$ $(M \leq N)$.

Dès lors, il est possible de définir une pénalisation du risque empirique pour traiter des hypothèses dépendantes des données.

Principe SRM avec dépendance aux données [1]. – Il est préconisé de sélectionner le classifieur f de $\mathcal{C}$ minimisant la quantité $R_{emp}(f) + r(K, M)$, où

$$r(K, M) = \sqrt{2 \frac{\log eM}{M \log M} \log \mathcal{S}_{M/N}^{(K)}}, \tag{3}$$

avec $M = N/2$ et sous la condition que $f \in \mathcal{C}_K$. e désigne le nombre de Néper.

3 Risque structurel d'un ensemble d'instances

Soit H un ensemble de K instances de l'échantillon D. Nous lui associons le classifieur suivant. Pour le k^{eme} élément de H, on considère l'ensemble D_k des instances de l'échantillon dont k est le plus proche élément dans H. Pour une instance quelconque, on détermine le plus proche parmi les élément de H, par exemple k, et on lui attribue l'étiquette correspondant à la classe majoritaire dans D_k. Nous notons $\mathcal{C}_K(D)$ l'ensemble des classifieurs obtenus à partir d'un sous-ensemble de cardinal au plus K de D. En reprenant les notations précédentes, $\mathcal{C}_K$ désigne l'union des $\mathcal{C}_K(D)$.

Proposition 1 (Borne sur le coefficient de pulvérisation) *Pour $K \leq M$, le coefficient de pulvérisation $\mathcal{S}_{M/N}^{(K)}$ de $\mathcal{C}_K$ est borné par $\binom{N}{M} \sum_{k=1}^{K} \binom{N}{k}$.*

Preuve. – Par définition, il suffit de majorer le cardinal $\mathcal{N}_M^{(K)}(D)$. On a :

$$\begin{aligned}\mathcal{N}_M^{(K)}(D) &= \#\{D_M^{(x)} \bigcap I_f;\ D_M \subset D, \#D_M = M \text{ et } f \in \mathcal{C}_K(D_M)\} \\ &\leq \# \bigcup_{D_M \subset D, \#D_M = M} \ \bigcup_{f \in \mathcal{C}_K(D_M)} \{D_N \bigcap I_f\} \\ &\leq \binom{N}{M} \sum_{k=1}^{K} \binom{N}{k}.\end{aligned}$$

En accord avec le principe SRM et l'équation (3), nous évaluons une hypothèse H par le risque structurel du classifieur f_H associé :

$$c^{SRM}(H) = R_{emp}(f_H) + \sqrt{\frac{4}{N}\frac{\log eN}{\log N}\left(\log\binom{N}{N/2} + \log\sum_{k=1}^{K}\binom{N}{k}\right)}. \tag{4}$$

Ce critère emploie le risque empirique afin de mesurer l'adéquation de l'hypothèse aux données et le pénalise en tenant compte de la capacité de l'hypothèse à séparer les classes.

4 Comparaison des critères

Dans l'article [2], le critère d'évaluation bayésien suivant a été proposé afin d'évaluer un ensemble H de K instances. Il emploie une vraisemblance pour mesurer l'adéquation de l'hypothèse aux données et la pénalise par un a priori sur les hypothèses :

$$c^{Bayes}(H) = \log N + \log\binom{N+K-1}{K} + \sum_{k=1}^{K}\log\binom{N_k+J-1}{J-1} + \log\frac{N_k!}{N_{k1}!\ldots N_{kJ}!}, \tag{5}$$

où, en reprenant les notations de la section précédente, N_k désigne le nombre d'instances dans D_k et N_{kj} le nombre d'instances dans D_k de classe j. Les trois premiers termes sont l'opposé du logarithme de l'a priori, le dernier terme est l'opposé de la log-vraisemblance.

Nous illustrons les différences de comportement des deux critères à l'aide d'un jeu de données synthétiques. Nous générons uniformément 2000 points dans un carré et considérons deux classes dont la distribution conditionnelle est $(0.9, 0.1)$ dans les coins supérieur droit et inférieur gauche, $(0.6, 0.4)$ dans les autres coins. Dans ce cas, la classe majoritaire est la même en tout point.

Nous définissons la notion de *trajectoire*. L'ensemble des 2000 instances est parcouru aléatoirement et uniformément, et chaque instance est éliminée à son tour. A chaque étape, l'ensemble des instances restantes est évalué pour chaque critère. On obtient ainsi une courbe de valeurs pour chacun des deux critères. Nous les visualisons sur la fig.1.

Le risque structurel, parce qu'il emploie le risque empirique comme mesure de qualité, ne s'intéresse qu'au caractère majoritaire d'une classe. Il est donc aveugle aux variations de la densité de la cible conditionnellement aux instances du moment que la classe majoritaire reste majoritaire. Sur le jeu de données considéré, cela conduit à sélectionner un singleton comme ensemble d'instances.

Le critère bayésien procède à une évaluation plus fine : il ne tient pas uniquement compte du caractère majoritaire d'une classe. En considérant la distribution des classes, il distingue plusieurs comportements de la densité conditionnelle sur le jeu de données. Il conduit ici à sélectionner un ensemble de quatre instances. Le critère bayésien est plus fin que le critère basé sur le risque empirique.

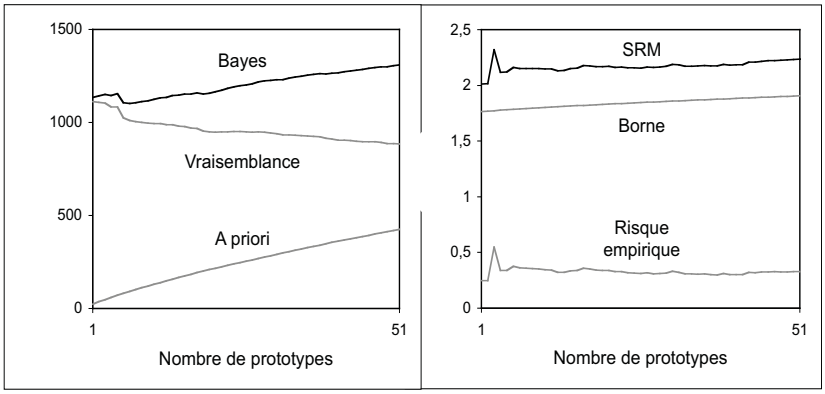


FIG. 1 – Exemple de trajectoires sur un jeu de données synthétiques. La meilleure partition au sens du risque structurel est celle constituée par un unique groupe : la classe majoritaire est la même en tout point du plan. La meilleure partition au sens du critère bayésien est constituée de 4 groupes : les variations de densité conditionnelle sont détectées.

5 Conclusion

La sélection d'instances pour la classification suivant le plus proche voisin constitue un problème d'apprentissage statistique nécessitant l'évaluation d'hypothèses dépendantes des données. Nous avons proposé un nouveau critère d'évaluation structurelle pour ce problème, basé sur la théorie de l'apprentissage statistique. Celui-ci permet de comparer des ensembles d'instances de tailles différentes, contrairement au risque empirique.

Nous avons comparé le critère structurel et un critère bayésien proposé par ailleurs. Le premier est moins fin que le second : il détecte les différences de classe majoritaire là où le critère bayésien détecte les différences de densité conditionnelle. Nous avons illustré cet état de fait sur des données synthétiques. Pour la sélection d'instances, l'emploi du critère bayésien est donc préconisé.

Références

[1] A. Cannon, J. Ettinger, D. Hush, and C. Scovel. Machine learning with data dependent hypothesis classes. *Journal of machine learning research*, 2 :335–358, 2002.

[2] S. Ferrandiz and M. Boullé. Supervised evaluation of voronoi partitions. *Journal of intelligent data analysis*, 10(3) :269–284, 2006.

[3] E. Fix and J. Hodges. Discriminatory analysis. nonparametric discrimination : Consistency properties. *Technical Report 4, Project Number 21-49-004, USAF School of Aviation Medicine, Randolph Field, TX*, 1951.

[4] V. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, New-York, 1996.

Une nouvelle méthode de classification pour des données intervalle.

A. Hardy[1] et N. Kasoro[2]

1. FUNDP - Université de Namur, Département de Mathématique, 8 Rempart de la Vierge, 5000 Namur Belgique
2. Université de Kinshasa, Département de Mathématique et Informatique, B.P. 190, Kinshasa, République Démocratique du Congo
(andre.hardy@fundp.ac.be, kasoro.mulenda@yahoo.fr

Mots clés : classification, données intervalle.

1. Un modèle statistique basé sur le processus de Poisson homogène ([5], [6])

   Le problème de classification auquel nous nous intéressons est le suivant.

   $E = \{x_1,\ x_2,\ ...,\ x_n\}$ est un ensemble de n objets sur lesquels on mesure la valeur de p variables quantitatives $Y_1,\ Y_2,\ ...,\ Y_p$. Nous recherchons une partition naturelle $P = \{C_1,\ C_2,\ ...,\ C_k\}$ de l'ensemble E des objets en k classes.

   On suppose que les points $x_1, x_2,\ \cdots,\ x_n$ que nous observons sont générés par un processus de Poisson homogène dans un domaine $D \subset R^p$, où D est l'union de k domaines convexes compacts disjoints $D_1,\ D_2,\ ...,\ D_k$. $C_i \subset \{x_1,\ x_2,\ ...,\ x_n\}$ est le sous-ensemble des observations appartenant à D_i $(1 \leq i \leq k)$. Le problème revient à estimer les domaines inconnus D_i dans lesquels les points ont été générés. Les estimateurs du maximum de vraisemblance des k domaines inconnus $D_1,\ D_2,\ ...,\ D_k$ sont les k enveloppes convexes $H(C_i)$ des sous-groupes C_i de points tels que la somme des mesures de Lebesgue des enveloppes convexes disjointes $H(C_i)$ est minimale. Le critère des Hypervolumes est alors défini par

   $$W_k = \sum_{i\ =\ 1}^{k} m(H(C_i)).$$

   Dans le contexte d'un problème de classification, l'objectif est donc de trouver la partition P^* telle que

   $$P^* = arg \min_{P \in \mathcal{P}_k} \sum_{i\ =\ 1}^{k} \int_{H(C_i)} m(dx).$$

2. Deux tests d'hypothèses pour la détermination du nombre de classes

   Un des problèmes centraux de la validation en classification est la détermination du bon nombre de classes naturelles. Nous utiliserons deux tests d'hypothèses basés sur le critère de classification des Hypervolumes: le test des Hypervolumes et le Gap test. Le test des Hypervolumes ([7], [8], [10]) est un test du quotient de vraisemblance déduit du modèle statistique pour la classification basé sur le processus de Poisson homogène. On teste l'hypothèse $H_0 : t = k$ contre l'alternative $H_1 : t = k - 1$ où t représente le nombre de classes naturelles $(k \geq 1)$. La statistique du test est

donnée par $S(x_1, x_2, \cdots, x_n) = \frac{W_k}{W_{k-1}}$ où W_k (respectivement, W_{k-1}) est la valeur du critère de classification des Hypervolumes associé à la meilleure partition en k classes (respectivement, en $k-1$ classes). Malheureusement, la distribution de la statistique du test n'est pas connue. La première approche est heuristique. Elle utilise la propriété suivante: $S(x_1, x_2, \cdots, x_n)$ appartient à $[0, 1[$. Le test est alors réalisé de manière séquentielle. Si k_0 est la première valeur de $k \geq 2$ pour laquelle on rejette H_0, alors on considèrera $k_0 - 1$ comme le nombre approprié de classes naturelles. Récemment des tests de permutation ont été utilisés afin de calculer des p-valeurs pour cette statistique ([9]). D'autre part le Gap test ([12]) est un autre test du quotient de vraisemblance basé sur le même modèle. On teste H_0 : les $n = n_1 + n_2$ points sont générés par un processus de Poisson homogène dans un ensemble D contre l'alternative H_1: n_1 points sont la réalisation d'un processus de Poisson homogène dans D_1 et n_2 points sont la réalisation d'un processus de Poisson homogène dans D_2 où $D_1 \cap D_2 = \emptyset$. La statistique du test, appelée "gap statistic", est la mesure de Lebesgue de l'espace vide entre les classes. Des distributions exacte et asymptotique existent pour cette statistique.

3. La nouvelle méthode de classification SPART

L'analyse des données symboliques ([1], [3]) a pour objectif d'étendre les méthodes classiques d'analyse des données et de statistique, à des données plus complexes, appelées données symboliques.

J.-Y. Pirçon ([13]) propose cinq nouvelles méthodes de classification monothétiques de partitionnement pour des données quantitatives classiques. L'une d'elles, HOPP (HOmogeneous Poisson Process), est développée en faisant l'hypothèse que les points observés sont la réalisation d'un processus de Poisson homogène. Le but de cet article est de proposer une extension de cette méthode à des données intervalles.

Une variable Y est appelée "à valeurs d'ensemble" de domaine $\mathcal{Y}$, si pout tout x_i appartenant à E,

$$\begin{array}{rcl} Y : E & \rightarrow & D \\ x_i & \longmapsto & Y(x_i) \end{array}$$

où l'ensemble de description D est défini par $D = \mathcal{P}(\mathcal{Y}) = \{\mathcal{U} \neq \emptyset \mid \mathcal{U} \subseteq \mathcal{Y}\}$.

Une variable à valeurs d'ensemble Y est appelée variable intervalle si son ensemble de description D est l'ensemble des intervalles bornés fermés de R.

De manière à réaliser l'extension de HOPP à des données intervalles, on utilise une modélisation. On représente un intervalle par un point (m, l) dans un espace bidimensionnel, où m est le centre de l'intervalle, et l sa demi-longueur. Cette nouvelle méthode, appelée SPART, est une procédure monothétique divisive de classification basée sur une extension à des données intervalle du critère de classification des Hypervolumes. Les principales étapes de la méthode sont les suivantes.

(a) Le critère de coupure

SPART est une méthode monothétique divisive. Grâce à la modélisation (m, l), l'ensemble des intervalles est transformé en un ensemble de points dans un espace bidimensionnel. Pour réduire la complexité de la méthode, on choisit la meilleure bipartition parmi les bipartitions obtenues en effectuant des coupures perpendiculaires à l'axe horizontal. Nous respectons ainsi l'ordre des centres

des intervalles. Le critère de coupure est une extension du critère des Hypervolumes. L'objectif est de trouver l'intervalle $]m_i, m_{i+1}[$ qui minimise

$$W_2^* = \int_{m_i}^{m_{i+1}} dm + \int_{min\{l_i,l_{i+1}\}}^{max\{l_i,l_{i+1}\}} dl.$$

La valeur de coupure c est n'importe quelle valeur de l'intervalle $]m_i, m_{i+1}[$ qui maximise W_2^*; on choisit habituellement le centre de l'intervalle. On définit une fonction binaire $q_c : E \longrightarrow \{vrai, faux\}$, et la bipartition $\{C_1, C_2\}$ de C est induite par les questions binaires du type "$m_k \leq c$?" où m_k est le centre de l'intervalle $Y(x_k)$ et c la valeur de coupure. On a donc $C_1 = \{x_k \in C : q_c(x_k) = vrai\}$ et $C_2 = \{x_k \in C : q_c(x_k) = faux\}$. L'algorithme commence avec tous les objets dans une seule classe, et il divise successivement chaque classe en deux. Le processus s'arrête après L itérations (L est généralement plus grand que le nombre de classes naturelles; il est fixé par l'utilisateur) ou lorsqu'un critère d'arrêt est vérifié (par exemple un nombre minimum de points dans une classe). A chaque étape on choisit la variable, et la classe à diviser, de manière à minimiser le critère étendu des Hypervolumes.

(b) L'élagage.

A l'étape précédente, les noeuds sont systématiquement coupés en deux, jusqu'à ce qu'une règle d'arrêt soit satisfaite. Le résultat est un arbre de grande taille. Un processus d'élagage est appliqué de manière à réduire la taille de l'arbre. Pour ce faire nous utilisons deux tests d'hypothèses basés sur le processus de Poisson homogène: le test des Hypervolumes et le Gap test. Ces tests sont appliqués à chaque noeud de l'arbre, de la racine aux extrémités, afin de répertorier les bonnes et les mauvaises coupures. Nous adoptons la règle suivante: les extrémités des branches pour lesquelles il n'y a que des mauvaises coupures sont élaguées. A la fin de cette étape nous obtenons un arbre hiérarchique, dont les singletons sont les classes obtenues à la fin du processus d'élagage. C'est aussi un arbre de décision et les classes peuvent être interprétées en fonction des variables originales.

(c) Le recollement

Lorsqu'on est en présence de structures particulières (par exemple lorsqu'aucune des classes n'est linéairement séparable des autres), il peut arriver que la partition naturelle des données ne soit pas obtenue à la fin de l'étape d'élagage. C'est la raison pour laquelle un processus de recollement est inclus dans la procédure de manière à vérifier s'il est pertinent de recoller certaines des classes obtenues la fin de l'étape d'élagage. On ne considère que les couples de classes qui ne proviennent pas de la division d'un même groupe au niveau précédent de la hiérarchie. Pour ce faire nous faisons à nouveau appel au test des Hypervolumes et au Gap test. Si un recollement a lieu, le caractère hiérarchique de la méthode est perdu. SPART est alors une méthode de partitionnement.

Une autre façon d'obtenir le bon nombre de classes est de combiner SPART avec NBCLUST, un ensemble de méthodes de détermination du nombre de classes pour des données intervalles, multivaluées et modales ([10]).

4. Illustrations et applications

Nous comparons les résultats produits par SPART avec ceux obtenus par d'autres méthodes de classification monothétiques divisives applicables à des données intervalle. La première, SCLASS ([14]), est une méthode hiérarchique divisive de classification basée sur une extension du critère généralisé des Hypervolumes ([4]). La seconde, DIV ([2]), est une méthode de classification monothétique divisive basée sur une extension du critère de la variance. DIV, SCLASS et NBCLUST sont disponibles dans le logiciel SODAS 2.

Des ensembles de données artificielles illustrent la nouvelle méthode. Quelques applications réelles sont également présentées.

Références

[1] H.-H. Bock, E. Diday (Eds.), "Analysis of Symbolic Data, Exploratory methods for extracting statistical information from complex data", *Studies in Classification, Data Analysis and Knowledge Organisation*, Springer Verlag, 2000.

[2] M. Chavent, "A monothetic clustering method", *Pattern Recognition Letters* 19, 1998, 989-996.

[3] E. Diday, M. Noirhomme (Eds.), "Symbolic data analysis and the SODAS 2 software", *Wiley*, 2007.

[4] V. Granville, "Bayesian filtering and supervised classification in image remote sensing.", *Thèse de doctorat. FUNDP-Université de Namur, Namur, Belgique*, 1993.

[5] A. Hardy, J.P. Rasson, "Une nouvelle approche des problèmes de classification automatique", *Statistique et Analyse des Données*, 23, 1982, 41-56.

[6] A. Hardy, "Statistique et classification automatique: un modèle, un nouveau critère, des algorithmes, des applications", *Thèse de doctorat, FUNDP - Université de Namur, Namur, Belgique*, 1983.

[7] A. Hardy, "On the number of clusters", *Computational Statistics and Data Analysis*, 1996, 83-96.

[8] A. Hardy, P. Lallemand, "Determination of the number of clusters for symbolic objects described by interval variables", *Studies in Classification, Data Analysis and Knowledge Organisation*, 2002, 311-318.

[9] A. Hardy, L. Blasutig, "Application des tests de permutation au critère des Hypervolumes en classification automatique". *Rapport de recherche, FUNDP - Université de Namur, Namur, Belgique*, 2006.

[10] A. Hardy, "Validation of a clustering structure: determination of the number of clusters", in *E. Diday, M. Noirhomme (Eds.). Symbolic data analysis and the SODAS 2 software, Wiley*, 2007.

[11] A.F. Karr, "Point processes and their statistical inference". *Marcel Dekker, New York*, 1991.

[12] T. Kubushishi, "On some Applications of Point Process Theory in Cluster Analysis and Pattern Recognition", *Thèse de doctorat, FUNDP-Université de Namur, Namur, Belgique*, 1996.

[13] J.-Y. Pirçon, "La classification et les processus de Poisson pour de nouvelles méthodes monothétiques de partitionnement", *Thèse de doctorat, FUNDP - Université de Namur, Namur, Belgique*, 2004.

[14] J.P Rasson, J.-Y. Pirçon, P. Lallemand, S. Adans, "Unsupervised divisive classification", *In: E. Diday, M. Noirhomme (Eds.), Symbolic data analysis and the SODAS 2 software, Wiley*, 2007.

Intégration d'information biologique dans le traitement de données Xomiques

F. Husson[1], M. de Tayrac[2], M. Aubry[2], J. Mosser[2], S. Lê[1]

1. Agrocampus, IRMAR, 65 rue de Saint-Brieuc, 35042 Rennes
2. Équipe de relation transcriptionnelle et oncogénèse, Université de Rennes I, 2 av L. Bernard, 35043 Rennes
(Francois.Husson, Sebastien.Le)@agrocampus-rennes.fr

Mots clés : analyse factorielle multiple, données Xomiques, information biologique supplémentaire.

Résumé : nous proposons dans cette présentation une méthodologie pour analyser des données Xomiques provenant de plusieurs sources d'information : deux types de mesures (sur les gènes et la transcription de l'ADN) et une information biologique supplémentaire sur les fonctions des gènes. Dans un premier temps, nous sélectionnons les gènes, puis nous utilisons l'AFM pour traiter simultanément et de façon équilibrée les deux types de mesures, enfin nous intégrons l'information biologique supplémentaire grâce aux groupes de variables supplémentaires en AFM. Un exemple sur les tumeurs du cerveau est détaillé.

1 Introduction

Dans le cadre des données Xomiques, il est de plus en plus fréquent que les données proviennent de sources différentes. On dispose notamment d'une meilleure connaissance des fonctions de chaque gène qu'il peut être intéressant d'intégrer dans l'analyse statistique de ces données.

2 Matériels et méthodes

Les données. Quarante-cinq patients atteints d'une tumeur au cerveau ([1]) sont classés selon quatre types de tumeurs différentes : oligodendrogliome (O), astrocytome (A), mixed oligo-astrocytome (OA) et glioblastome (GBM), ce dernier étant le cancer de grade le plus élevé. Chaque tumeur a été analysée au niveau du génome (gène) et du transcriptome (CGH). L'information biologique sur les fonctions des gènes est obtenue grâce à une ontologie de gènes (disponible à l'adresse suivante :
`http://www.geneontology.org/GO.usage.shtml#truePathRule`).

Les données Xomiques dont nous disposons proviennent de deux types de mesures sur l'altération des gènes et d'informations biologiques sur la fonction de chaque gène. L'altération des gènes est mesurée au niveau du génome et au niveau du transcriptome (par hybridation génomique comparative, CGH, qui consiste à comparer un chromosome d'un tissu sain avec celui d'un tissu potentiellement malade). Ces mesures sont effectuées sur des puces ADN ce qui permet d'avoir une information sur de nombreux gènes (environ 40000 sur une puce).

Sélection des variables. Le nombre de gènes étant extrêmement important, il est indispensable de les sélectionner. En effet, la plupart d'entre eux n'interviennent nullement dans la maladie étudiée mais par défaut tous sont mesurés.

Pour cette sélection, nous utilisons l'analyse de variance à un facteur (le facteur type de tumeur) pour expliquer chaque variable (chaque gène). Nous conservons les variables pour lesquelles la probabilité critique associée au test F est suffisamment petite.

Notons que cette approche est justifiée par le fait qu'un gène apporte une information unidimensionnelle mais que la multiplicité des gènes permet une approche multidimensionnelle puisque les différents gènes n'opposent pas les mêmes types de tumeurs.

Recherche d'une structure commune. La prise en compte simultanée des deux sources d'information (gènes et CGH) pour différencier les types de tumeurs est possible grâce à des méthodes multi-tableaux telles que l'analyse canonique ([2]) et l'analyse factorielle multiple (AFM, [3]). Un des avantages de l'AFM dans le cas de tableaux de grandes tailles est d'uniquement diagonaliser la matrice des covariances tandis que l'analyse canonique inverse cette même matrice. Outre la difficulté d'inverser des matrices de grandes tailles, il s'ensuit une instabilité des axes lorsque les variables sont très corrélées. Un autre avantage de l'AFM est que cette méthode fournit des aides à l'interprétation graphiques qui permettent une comparaison directe des structures communes entre les groupes de variables.

Intégration d'information biologique. Une information supplémentaire peut être apportée ici : elle concerne les fonctions des gènes. Plusieurs gènes vont définir une fonction ou un processus biologique, et il est intéressant de savoir si cette fonction intervient de façon différente d'un type de tumeur à l'autre. D'un point de vue statistique, une fonction est un groupe de variables qui sera pris en compte comme un élément supplémentaire dans l'AFM.

3 Résultats

Les résultats ont été obtenus grâce au package d'analyse de données `FactoMineR` ([4]).

Pour la sélection de variables, nous avons utilisé un seuil de 0.20 permettant d'éliminer des gènes dont on est sûr qu'ils apportent très peu d'information. Pour le tableau de gènes (resp. CGH), nous conservons 358 (resp. 68) variables.

L'AFM fournit une représentation des tumeurs dans laquelle chaque tumeur est représentée par un point d'une couleur caractérisant son type (Fig. 1 gauche). Le type de tumeur est une variable qualitative qui est projetée en tant qu'élément supplémentaire sur le plan de l'ACP. Les tumeurs des différents types sont bien différenciées, notamment les tumeurs de haut grade (GBM). Ce type de tumeur est très caractéristique de l'axe 1, tandis que les autres types de tumeurs se différencient principalement sur l'axe 2.

Les coordonnées des groupes de variables CGH et gène sur l'axe 1 sont élevées (Fig. 2 gauche), ce qui montre que les 2 groupes ont participé activement à la différenciation des individus sur l'axe 1, et par suite, à la différenciation des types de tumeurs sur l'axe 1. En revanche, la coordonnée du groupe "gène" sur l'axe 2 est faible, tandis que celle du groupe "CGH" est élevée. Cela signifie que le groupe de variables CGH est au moins bidimensionnelle et différencie les individus (et les types de tumeurs) sur l'axe 2 également. Les données de CGH sont donc plus riches car plus multidimensionnelles. On peut de plus représenter la variable qualitative type de tumeur comme un groupe de variables illustratif

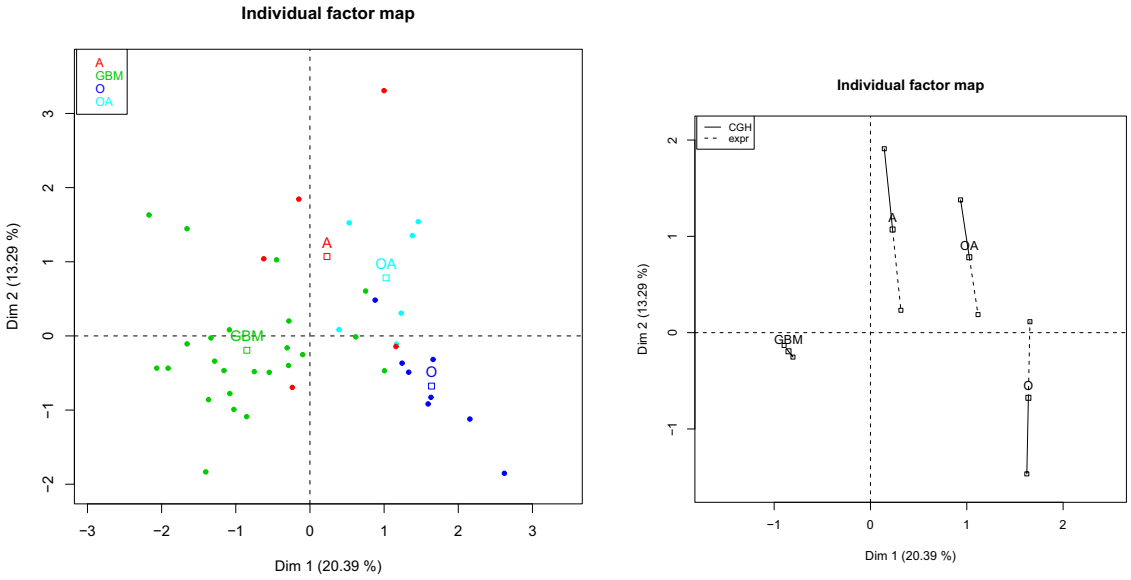


Figure 1: graphe de gauche : nuage des individus; graphe de droite : représentation des modalités et de leurs représentations partielles

(limité ici à une seule variable). Ce groupe a une forte coordonnée sur le premier axe et une relativement forte coordonnée sur le deuxième axe, ce qui confirme l'impression visuelle que les types de tumeurs sont bien différenciés sur le plan des individus de l'AFM.

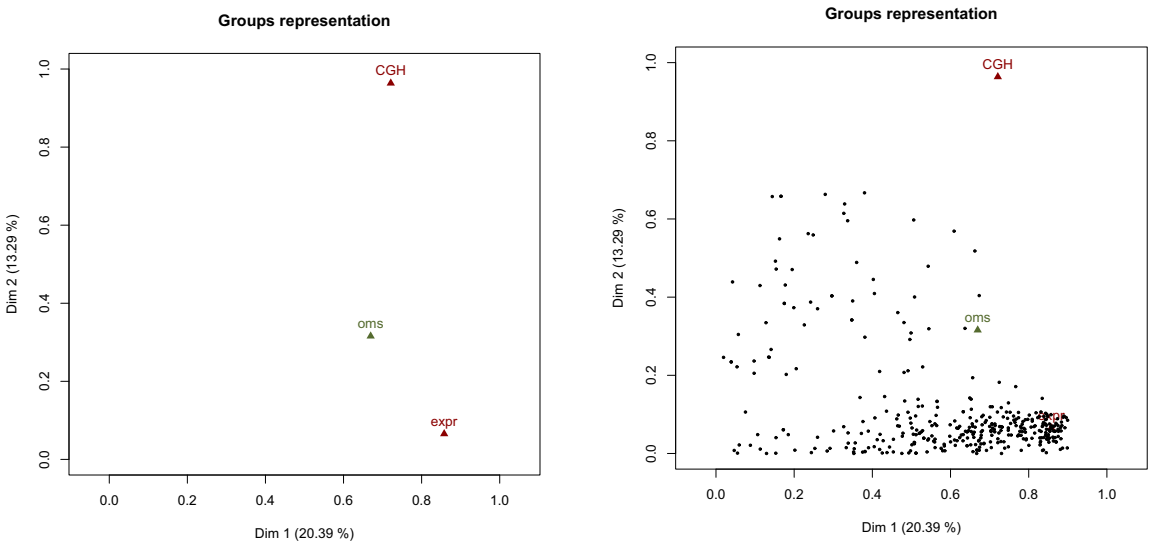


Figure 2: graphe de gauche : nuage des groupes de variables; graphe de droite : projection de l'information biologique supplémentaire

Si on s'intéresse plus particulièrement à la variable qualitative "type de tumeur", on peut visualiser les points partiels de ces modalités. A chaque type de tumeur sont associés deux point partiels : le type de tumeur vu exclusivement par les variables "CGH" et le type de tumeur vu exclusivement par les variables "gène". Le graphe de droite de la figure 1 confirme que les données de CGH permettent de différencier les types de tumeurs

sur l'axe 2 contrairement aux données "gène". En effet, les coordonnée partielles de CGH prennent des valeurs très différentes sur l'axe 2 (très faible pour O, très élevée pour A) tandis que les points partiels "gène" ont tous une coordonnée proche de 0 sur l'axe 2.

L'ajout d'information biologique supplémentaire (Fig. 2 droite) est précieuse pour interpréter les axes. De nombreuses fonctions biologiques (une fonction = un groupe de gènes) ont une forte coordonnée sur l'axe 1, ce qui signifie que de nombreuses fonctions sont atteintes pour les GBM par rapport aux autres types de tumeurs. Ce type de tumeur étant d'un grade plus avancé que les autres, il est attendu que de nombreuses fonctions biologiques soient touchées. Le tableau 1 facilite la lecture du graphe de droite de la figure 2 en fournissant les coordonnées de chaque fonction sur le premier axe triées par coordonnée décroissante, ce qui permet de mettre en évidence les fonctions qui séparent le plus les GBM des autres types de tumeurs. Le tableau 2 trie les fonctions selon leur coordonnée sur le deuxième axe, permettant ainsi de savoir quelles sont les fonctions qui sont différemment touchées entre les types de tumeurs O, A et OA.

GO terms / Modules	Genes	dim 1
Cell motility	20	0.871
Cofactor biosynthesis	3	0.865
Coenzyme metabolism	3	0.865
DNA metabolism	10	0.857
Negative regulation of cellular physiological process	37	0.854
Nucleobase, nucleoside, nucleotide and nucleic acid metabolism	59	0.849
Cofactor metabolism	4	0.847
Chromosome organization and biogenesis	7	0.844

Table 1: information biologique supplémentaire : les fonctions sont triées par coordonnées décroissantes sur le premier axe. *Le nombre de gènes présents dans chaque fonction et sa coordonnée sur la première dimension sont fournis.*

GO terms / Modules	Genes	dim 2
Nucleobase, nucleoside, nucleotide and nucleic acid metabolism	17	0.549
Cellular metabolism	36	0.627
Primary metabolism	32	0.590

Table 2: information biologique supplémentaire : les fonctions sont triées par coordonnées décroissantes sur le deuxième axe. *Le nombre de gènes présents dans chaque fonction et sa coordonnée sur la deuxième dimension sont fournis.*

[1] M. Bredel, C. Bredel, D. Juric, D. Harsh, G.R. Vogel, L.D. Recht, B.I. Sikic, "Functional network analysis reveals extended gliomagenesis pathway maps and three novel MYC-interacting genes in human gliomas", *C*ancer research, 65, 2005, 8679-8689.

[2] J.D. Carroll, "A generalization of canonical correlation analysis to three or more sets of variables.", *Proc. 76th Amer. Psych. Assoc.*, 1968.

[3] B. Escofier et J. Pagès, "Analyses factorielles simples et multiples.", *Dunod*, 1998.

[4] F. Husson, S. Lê et J. Mazet, "FactoMineR: Factor Analysis and Data Mining with R.", 2007, `http://factominer.free.fr`.

Une approche incrémentale d'une méthode de classification non supervisée par nuages d'insectes volants

J. Lavergne[1], H. Azzag[2], C. Guinot[1,3] et G. Venturini[1]

1. Laboratoire d'Informatique de l'Université de Tours, 37200 Tours, France
2. Laboratoire d'Informatique de l'Université Paris-Nord, 93430 Villetaneuse, France
3. CE.R.I.E.S, 92521 Neuilly-Sur-Seine, France
1. {julien.lavergne,venturini}@univ-tours.fr, http://www.antsearch.univ-tours.fr/webrtic
2. hanane.azzag@lipn.univ-paris13.fr, http://www-lipn.univ-paris13.fr/A3/
3. christiane.guinot@ceries-lab.com, http://www.ceries.com

Mots clés : approches inspirées du vivant, classification, flux de données.

1 Introduction

Nous nous intéressons dans cet article au problème de la classification incrémentale d'un flux de données (arrivant séquentiellement). Le volume de données considéré est tel qu'il est impossible de traiter toutes les données ensemble. Par ailleurs il se peut également que la classification évolue au cours du temps par l'apparition ou la disparition de classes. Nous proposons dans cet article une adaptation incrémentale d'un algorithme de classification par nuages d'insectes volants [12].

2 Modèle biologique et algorithme initial

Le vivant a souvent été une source d'inspiration pour répondre à des problèmes de classification non supervisée (i.e. algorithmes génétiques [9, 8], algorithmes biomimétiques [11, 10, 2, 1]). Ces derniers présentent plusieurs avantages : ils fonctionnent de manière distribuée et généralement sans classification initiale des données. Ils acceptent aussi bien des données numériques, symboliques que textuelles et se prêtent bien à un fonctionnement incrémental. Dans notre cas, nous nous intéressons à "l'intelligence en essaim" [6] et plus particulièrement aux modes de déplacement de certains animaux (i.e. regroupements d'oiseaux ou de poissons sous forme de nuages). L'algorithme étudié dans [12] est tel que les insectes volants représentent les données à regrouper (voir des approches similaires dans [13, 7]). Nous disposons pour cet algorithme d'un ensemble de données non étiquetées et d'une mesure de similarité entre ces données sans autre hypothèse préalable. Le but fixé est de regrouper ces données en classes. Chaque insecte représente une donnée et se déplace dans un environnement 2D. Pour calculer son déplacement, un insecte est donc caractérisé par ses coordonnées, un vecteur vitesse et un ensemble de règles locales simples. Ces dernières, communes à tous les individus, tendent à établir une distance entre insectes qui dépend de la similarité entre les données. Les insectes, en se déplaçant, vont se rencontrer et suivant les règles décider de se rapprocher ou de s'éloigner et d'aller ou non dans la même direction. L'expert du domaine peut ainsi visualiser dynamiquement au cours des itérations le regroupement des données sous la forme de nuages. L'algorithme donne en sortie un partitionnement en classes des données de l'ensemble initial. Pour

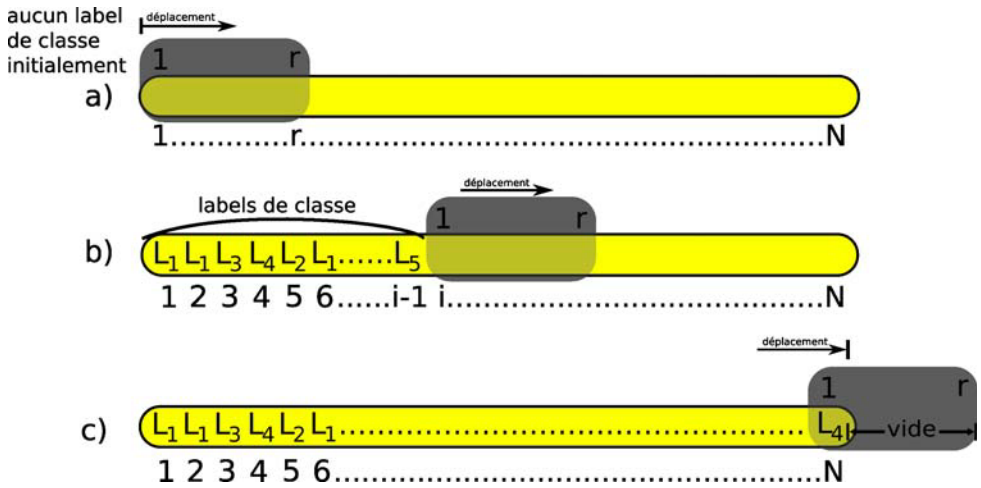


FIG. 1 – Illustration du principe de la fenêtre glissante : a) initialisation, b) propagation des labels entre données similaires et c) toutes les données ont été classées.

cela, un critère d'arrêt basé sur l'entropie spatiale est utilisé. L'algorithme a été validé sur des bases artificielles [1] et réelles [5] pour lesquelles une classification existe (cette information n'étant pas communiquée à l'algorithme).

Cet algorithme possède un aspect incrémental inexploité. En effet, on peut remarquer que l'on peut facilement enlever ou ajouter des données sans nécessiter de redémarrage de l'algorithme. Nous proposons dans la section suivante une extension incrémentale de cet algorithme afin de pouvoir classifier un flux de données de taille quelconque.

3 Approche incrémentale

Nous considérons maintenant un flux de N données, N pouvant prendre une valeur très grande (1 million par exemple) ou même infinie. Comme cela a déjà été proposé en fouille de flux de données, nous utilisons le principe d'une fenêtre temporelle glissante [3, 4] de largeur fixée r (voir figure 1). A chaque itération, nous décalons la fenêtre sur l'ensemble des données en procédant respectivement à la suppression de la plus ancienne donnée, au décalage d'un cran des autres données et à l'ajout d'une nouvelle donnée dans la fenêtre. Chaque fois qu'une donnée sort de la fenêtre, notre algorithme va lui attribuer un label de classe. Aucun autre post-traitement n'est nécessaire. Une nouvelle donnée arrivant fait partie d'une classe unique et va éventuellement intégrer les classes existantes.

Pour ce faire, des modifications supplémentaires ont été apportées à l'algorithme étudié dans [12]. Tout d'abord, cet algorithme va utiliser uniquement un nuage composé des r données de la fenêtre. Nous insérons la donnée arrivant dans la fenêtre à une position aléatoire dans l'environnement afin qu'elle puisse s'intégrer aux groupes existants (ou créer une nouvelle classe). Elle sera simulée pendant les r itérations où elle est présente dans la fenêtre. Enfin, nous arrêtons l'algorithme dès que l'on atteint la fin du flux de données (fenêtre vide). Pour donner un label de classe à chaque donnée, nous ne pouvons pas réaliser une classification finale sur l'ensemble des données. Nous proposons pour cela un algorithme de classification nommé *algorithme des labels* dont le principe est le suivant : chaque insecte porte un label de classe initialisé avec son numéro/indice de donnée. A chaque itération, chaque insecte parcourt son voisinage local à la recherche de l'insecte ayant le label de valeur minimum. L'insecte prend cette valeur de label si sa similarité avec le voisin en question est supérieure à un seuil de similarité Sim_{Seuil} (défini

r	Bases	C_R	T_{exec}	C_T	$C_{T_{\frac{N}{20}}}$	P_R
200	waveform	3	34,14 [1,25]	533,60 [28,41]	1,00 [0,00]	0,42 [0,00]
	10000-ta	5	50,08 [0,07]	15,20 [3,31]	5,00 [0,00]	1,00 [0,00]
	letter-recognition	26	127,07 [0,94]	9370,60 [35,39]	1,00 [0,00]	0,58 [0,00]
	25000-ta	5	129,90 [2,25]	26,60 [3,01]	5,00 [0,00]	1,00 [0,00]
	60000-ta	5	314,71 [5,37]	56,20 [8,63]	5,00 [0,00]	1,00 [0,00]
	100000-ta	5	529,06 [7,16]	90,00 [8,29]	5,00 [0,00]	1,00 [0,00]
	1000000-ta	5	5323,52 [3,01]	839,60 [6,62]	5,00 [0,00]	1,00 [0,00]
300	waveform	3	64,13 [0,05]	183,20 [16,70]	1,00 [0,00]	0,37 [0,00]
	10000-ta	5	113,08 [1,49]	5,40 [0,49]	5,00 [0,00]	1,00 [0,00]
	letter-recognition	26	250,17 [0,13]	6443,40 [59,34]	1,00 [0,00]	0,43 [0,00]
	25000-ta	5	290,41 [0,31]	5,40 [0,49]	5,00 [0,00]	1,00 [0,00]
	60000-ta	5	700,15 [3,10]	6,00 [0,89]	5,00 [0,00]	1,00 [0,00]
	100000-ta	5	1166,59 [5,05]	5,80 [0,75]	5,00 [0,00]	1,00 [0,00]
	1000000-ta	5	11653,16 [41,71]	15,20 [3,19]	5,00 [0,00]	1,00 [0,00]
500	waveform	3	152,81 [0,58]	84,20 [8,42]	1,00 [0,00]	0,35 [0,00]
	10000-ta	5	293,68 [0,77]	5,00 [0,00]	5,00 [0,00]	1,00 [0,00]
	letter-recognition	26	629,13 [0,47]	3205,00 [42,83]	1,00 [0,00]	0,23 [0,00]
	25000-ta	5	755,12 [2,87]	5,00 [0,00]	5,00 [0,00]	1,00 [0,00]
	60000-ta	5	1832,12 [2,42]	5,00 [0,00]	5,00 [0,00]	1,00 [0,00]
	100000-ta	5	3068,21 [2,02]	5,00 [0,00]	5,00 [0,00]	1,00 [0,00]
	1000000-ta	5	31101,70 [163,37]	5,00 [0,00]	5,00 [0,00]	1,00 [0,00]

TAB. 1 – r est la taille de la fenêtre glissante, C_R le nombre de classes réelles, T_{Exec} les temps d'exécution en seconde, C_T le nombre de classes trouvées, $C_{T_{\frac{N}{20}}}$ le nombre de classes trouvées majoritaires (i.e. $\geq \frac{N}{20}$) et P_R la pureté de classification.

dans [12]). Cet algorithme permet la continuation d'une classe dans le temps mais aussi la création de nouvelles classes, deux propriétés nécessaires pour classer un flux de données.

4 Résultats et Conclusions

Nous avons réalisé cette première étude sur des ensembles de données numériques, artificielles et réelles ayant de 5000 à 1000000 de données. Nous avons construit des jeux de données ({10000, 25000, 60000, 100000, 1000000}-ta) générés avec une loi uniforme (i.e. $C_R = 5$, sans recouvrement). Nous pouvons ainsi évaluer au mieux les capacités de notre approche. Les bases réelles (waveform avec 5000 données, letter-recognition avec 20000 données) proviennent du UCI Repository of Machine Learning [5]. Toutes les bases ont été mélangées aléatoirement.

La table 1 présente les résultats obtenus. Nous remarquons que T_{Exec} augmente et C_T diminue, et ce, en fonction de la taille r de la fenêtre : un label de classe commun se propageant alors plus facilement sur un plus grand nombre de données au cours d'une itération. Nous retrouvons globalement les classifications pour la plupart des bases artificielles testées en des temps d'exécution corrects (voir $r = 200$). Nous pouvons également souligner le fait que $C_{T_{\frac{N}{20}}}$ est très proche de C_R pour les bases artificielles. Ce n'est pas le cas pour les bases réelles où les difficultés des jeux de données sélectionnés empêchent une bonne classification. De plus, nous pouvons remarquer que l'algorithme des labels produit, dans tous les cas, un nombre important de classes pour toutes les bases. Si l'on compare $C_{T_{\frac{N}{20}}}$ avec C_T, nous pouvons dire que notre algorithme propage les labels de classe sur de grands sous-ensembles de données. La pureté P_R (égale à 1 pour toutes les

bases artificielles) qui représente le pourcentage de données bien classées nous renseigne sur le fait que ces sous-ensembles contiennent bien des données similaires.

Les résultats obtenus sont encourageants en terme de classification. Cependant, les résultats sur les bases réelles ainsi que le grand nombre de petites classes créées nous poussent à améliorer notre approche en ce sens. De plus nous aimerions pouvoir proposer le calcul d'un seuil de similarité dynamique. A terme, nous allons proposer une étude comparative.

Références

[1] H. Azzag. *Classification hiérarchique par des fourmis artificielles : applications à la fouille de données et de textes pour le Web*. PhD thesis, Université de Tours, 2005.

[2] H. Azzag, D. Ratsimba, D. D. Costa, C. Guinot, and G. Venturini. On building maps of web pages with cellular automata. In *BICC 2006 : IFIP Conference on Biologically Inspired Cooperative Computing*, pages 33–42, Santiago, Chile, August 20-26 2006.

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *PODS '02 : Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16, New York, NY, USA, 2002. ACM Press.

[4] B. Babcock, M. Datar, R. Motwani, and L. O'Callaghan. Maintaining variance and k-medians over data stream windows. In *PODS '03 : Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 234–243, New York, NY, USA, 2003. ACM Press.

[5] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[6] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence : From Natural to Artificial Systems*. Oxford University Press, New York, 1999.

[7] X. Cui, J. Gao, and T. E. Potok. A flocking based algorithm for document clustering analysis. *Journal of Systems Architecture*, 52(8-9) :505–515, 2006.

[8] E. Falkenauer. A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation*, 2(2) :123–144, 1994.

[9] D. Jones and M. Beltramo. Solving partitioning problems with genetic algorithms. In R. Belew and L. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 442–449, San Diego, CA, 1991. Morgan Kaufmann.

[10] P. Kuntz, P. Layzell, and D. Snyers. A colony of ant-like agents for partitioning in vlsi technology. In P. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*, pages 417–424, 1997.

[11] E. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In D. Cliff, P. Husbands, J. Meyer, and S. W., editors, *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 501–508. MIT Press, Cambridge, Massachusetts, 1994.

[12] F. Picarougne, H. Azzag, G. Venturini, and C. Guinot. On data clustering with a flock of artificial agents. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, pages 777–778, Boca Raton, Florida, USA, 2004. IEEE Computer Society.

[13] G. Proctor and C. Winter. Information flocking : Data visualisation in virtual worlds using emergent behaviours. In J.-C. Heudin, editor, *Proc. 1st Int. Conf. Virtual Worlds, VW*, volume 1434, pages 168–176. Springer-Verlag, 1998.

Construction d'arbres à partir de relations d'intermédiarité, Application au stemma codicum

M. Le Pouliquen [1,2], J.P. Barthélemy [1,3]

1. Département LUSSI TAMCIC, UMR CNRS 2872
ENST Bretagne, BP 832, 29285 Brest Cedex
2. Université de Bretagne Occidentale IUP GMP
6 av. Le Gorgeu - CS93837 29238 Brest Cedex 3
3. CAMS, UMR CNRS 8557, Ecole des Hautes Etudes en Sciences Sociales
54 bvd Raspail, 75270, Paris cedex 06
(marc.lepouliquen, jp.barthelemy)@enst-bretagne.fr

Mots clés : intermédiarité, arbre, stemma codicum

1 Introduction

La notion d'intermédiarité est une notion assez naturelle dans de nombreux domaines :
- un lieu B est intermédiaire entre A et C s'il se trouve sur un chemin menant de A à C.
- 1914 est intermédiaire entre 1870 et 1939.
- le vert est entre le jaune et le bleu.
- Paul est entre Jean et Louis si Paul est le fils de Louis et le père de Jean.
- Un ensemble B est entre les ensembles A et C si par exemple $A \subset B \subset C$ ou $C \subset B \subset A$...

En premier, nous allons tenter dans cet article de modéliser cette relation dans le cadre de l'intermédiarité de textes. Nous nous positionnons dans la problématique de l'édition critique. L'éditeur doit tenter de reconstituer au mieux, à partir des différents manuscrits conservés, l'œuvre telle que l'auteur l'a voulue. Le corpus de manuscrits est constitué de nombreuses copies de l'œuvre faites les unes sur les autres par des copistes et pourquoi pas de l'œuvre elle-même. Pour se faire, il apparaît intéressant de dresser un arbre généalogique de ces manuscrits appelé « stemma codicum » (cf. fig 1). Sur la figure, on constate que le manuscrit F est intermédiaire entre les manuscrits G et I, c.à.d. que le manuscrit G a été copié à partir du manuscrit F qui lui-même a été copié sur le I. C'est cette notion d'intermédiarité par la copie que l'on souhaite modéliser.

Dans un deuxième temps, nous nous intéressons aux possibilités que l'on a de reconstruire un arbre à partir d'un certain nombre de relations d'intermédiarité entre les sommets. On peut se demander si un ensemble de relation d'intermédiarité est modélisable par à un arbre et si oui, comment peut-on le reconstruire ?

Pour finir, nous détaillons les algorithmes que nous avons testé sur un corpus réel. Nous regardons dans quelles mesures ils permettent d'améliorer les techniques de classification de manuscrits avant de conclure.

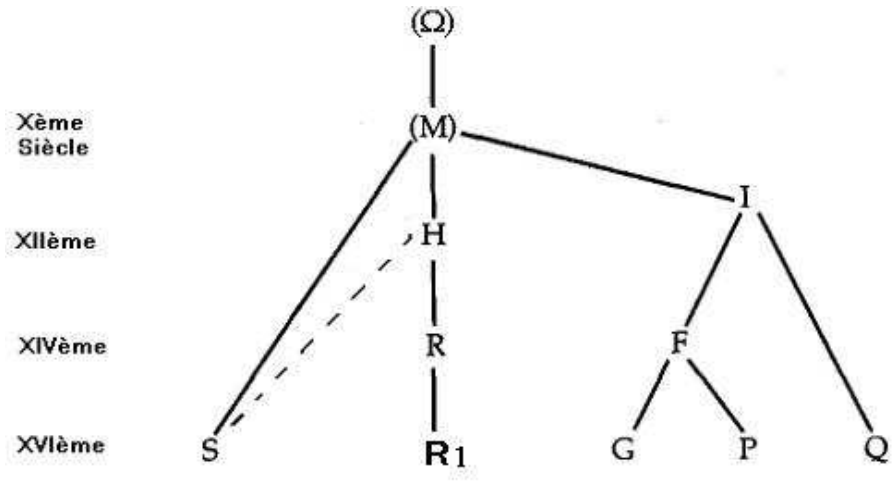


FIG. 1 – **Stemma codicum** chaque lettre correspond à un manuscrit, les parenthèses indiquent les manuscrits perdus ou supposés.

2 Différentes modélisations de l'intermédiarité au niveau des manuscrits

C'est à Don Quentin [7] que l'on doit l'idée d'utiliser la notion d'intermédiarité afin de dresser le stemma. En effet, il se propose de reconstituer des petites chaînes de trois manuscrits dont l'un est l'intermédiaire des deux autres puis, d'assembler ces petites chaînes afin d'inférer l'arbre complet. En s'inspirant de cette démarche, nous allons essayer de caractériser l'intermédiarité.

Pour définir l'intermédiarité, nous utilisons quatre caractérisations différentes, celle de la géométrie, celle liée à une structure définie par une relation binaire ou par un score et celle de la théorie des ensembles.

2.1 Intermédiarité en géométrie

Des nombreuses caractérisations géométriques, celle qui nous intéresse est celle introduite par Menger [5] sous le nom de relation métrique d'intermédiarité de la façon suivante :

Définition 1 *Une relation ternaire B sur une ensemble E est dite relation métrique d'intermédiarité s'il existe une métrique d sur E telle que :*

$$(a, b, c) \in B \Leftrightarrow d(a, b) + d(b, c) = d(a, c)$$

A partir de maintenant, pour exprimer le fait que b est intermédiaire entre a et c, on se contentera d'écrire (a, b, c) au lieu de $(a, b, c) \in B$.

Il faut désormais définir une métrique qui permet de comparer les manuscrits de telle sorte que l'intermédiarité au sens de la copie corresponde à celle de la métrique. Pour cela, prenons un exemple. Soient les 3 phrases suivantes correspondants aux trois mêmes phrases de différents manuscrits copiés les uns sur les autres.

A =« Voici une phrase courte inventée pour l'exemple »
B =« Voici une phrase inventée pour cet exemple »
C =« Voici une phrase crée pour cet exemple »

La logique veut que la phrase B soit intermédiaire au sens de la copie entre A et C. En effet, le copiste de B a omis « courte » et a modifié « l' » en « cet » et le copiste de C a remplacé « inventée » par « créée ». C'est en revanche peu probable que C soit l'intermédiaire, car le copiste de C a supprimé « inventée » qui est réintroduit par le copiste suivant. Soit la distance d définie par le nombre de mots insérés, supprimés ou substitués entre 2 textes alignés (une sorte de distance d'édition (Levenstein [3]) au niveau des mots). Dans notre cas, les calculs de d donnent :

A	Voici une phrase	courte	inventée	pour	l'	exemple
B	Voici une phrase		inventée	pour	cet	exemple
C	Voici une phrase		créée	pour	cet	exemple
$d(A,B)=2$		$d(B,C)=1$		$d(A,C)=3$		

TAB. 1 – Alignement des trois phrases pour compter le nombre de variantes

On constate en effet que (A, B, C) puisque $d(A, C) = d(A, B) + d(B, C)$ et l'on n'a pas (A, C, B).

2.2 Intermédiarité définie par une relation binaire

Nous pouvons aussi présenter une relation ternaire avec une relation d'ordre.

Définition 2 *Une relation ternaire B que nous appelons intermédiarité d'ordre de la façon suivante : (a, b, c) si et seulement si $a \leq b \leq c$ ou $c \leq b \leq a$. Il faut donc définir une relation d'ordre entre les manuscrits au sens de :*

$$M1 \leq M2 \Leftrightarrow M1 \text{ a été copié sur } M2 \text{ ou est égal à } M2.$$

Malheureusement, étant donné deux manuscrits, on n'a aucune façon de savoir si l'un a été copié sur l'autre ou vice versa à l'examen du texte. On peut se douter que deux manuscrits trop dissemblables ne sont pas des copies l'un de l'autre. Mais les copistes étant plus ou moins précis, les manuscrits copiés sont plus ou moins proches et leurs proximités ne traduisent pas forcément une filiation directe. Il n'y a donc pas de modélisation envisageable de la relation binaire et donc de l'intermédiarité qu'elle induirait.

2.3 Intermédiarité définie par un score

Le but est de construire un indice qui permet de détecter si un manuscrit est intermédiaire entre deux autres au sens de la copie. On observe en premier lieu que trois stemmas différents peuvent être construits conformément à *« B est intermédiaire entre A et C »* (cf. fig 2).

Pour construire le stemma à partir des différents manuscrits de la tradition textuelle, on part du constat suivant, à savoir que toutes les copies qui contiennent, aux mêmes endroits, les mêmes fautes, ont été faites les unes sur les autres et donc dérivent toutes d'une copie où ces fautes existaient. Ces fautes sont appelées variantes ou leçons. Dans

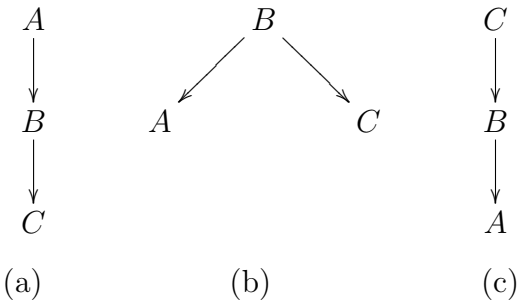


FIG. 2 – Stemmas possibles si *« B est intermédiaire entre A et C »*

les trois cas de la figure 2 , il y a un moyen de déterminer l'enchaînement des manuscrits à partir des variantes mais pas d'orienter le sens de la copie. En effet pour que B soit intermédiaire entre A et C, il suffit que A et C s'accordent tour à tour avec B au niveau des variantes mais surtout qu'ils ne s'accordent jamais contre lui. Reprenons notre exemple précédent :

A=« Voici une phrase courte inventée pour l'exemple »
B=« Voici une phrase inventée pour cet exemple »
C=« Voici une phrase créée pour cet exemple »

On a ici 3 variantes : *courtes*, *inventée/créée* et *l'/cet*, résumées dans le tableau suivant :

N° du lieu variant	Variante	Phrases associées	Variante	Phrases associées
1	courte	A	$\{\emptyset\}$	B,C
2	inventée	A,B	créée	C
3	l'	A	cet	B,C

Pour les variantes 1 et 3, on voit que B s'accorde avec C. Pour la variante 2, B s'accorde avec C et surtout A et C ne s'accordent jamais contre B. B est donc bien intermédiaire entre A et C. On peut construire un indice en divisant le nombre de lieux variants où ils ne s'accordent pas par celui où B s'accorde avec l'un des deux. On détermine ainsi si un manuscrit est « plus ou moins intermédiaire » (s'il est nul, il est intermédiaire). Cela correspond bien à l'indice recherché.

$$\text{Ici } Indice_B = \frac{0}{3} = 0,\ Indice_A = \frac{2}{1} = 2 \text{ et } Indice_C = \frac{1}{2} = 0,5$$

2.4 Intermédiarité en théorie des ensembles

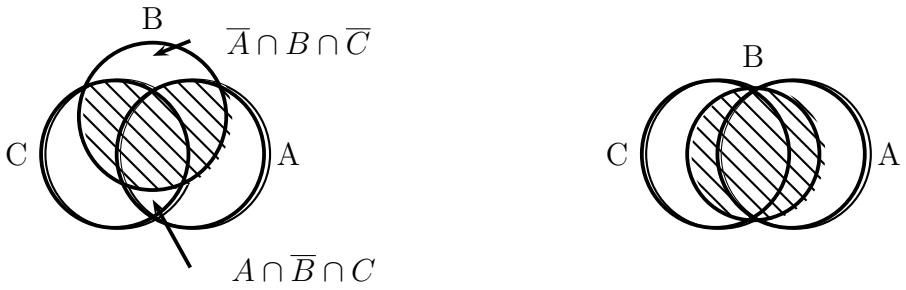
Nous allons ici nous intéresser à la relation d'intermédiarité introduite par Restle[8]. Cette relation définit la notion d'intermédiarité au niveau des ensembles.

Définition 3 *Soient trois ensembles A,B et C. On considère que B est intermédiaire entre A et C ssi :*

(i) $A \cap \overline{B} \cap C = \emptyset$

(ii) $\overline{A} \cap B \cap \overline{C} = \emptyset$

Cela correspond à la figure 3.



B n'est pas intermédiaire entre A et C B est intermédiaire entre A et C

FIG. 3 – Intermédiarité au sens de Restle.

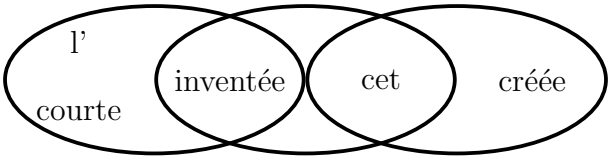
Dans le cas de nos manuscrits, nous pouvons très bien les associer à un ensemble qui les caractérise les uns par rapport aux autres, l'ensemble des variantes. Reprenons à nouveau notre exemple précédent :

A=« Voici une phrase courte inventée pour l'exemple »

B=« Voici une phrase inventée pour cet exemple »

C=« Voici une phrase créée pour cet exemple »

L'ensemble A est constitué des variantes { courte, inventée , l' }, B de { inventée , cet } et C de { créée , cet }. On peut alors le modéliser par :



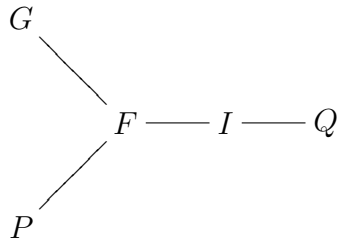
Cela correspond à nouveau au fait que B est entre A et C. Si l'on retrouve la même variante plusieurs fois dans le texte, on devra peut-être les indicer afin de les identifier et de ne pas les mélanger.

3 Utilisation de l'intermédiarité pour la représentation du stemma par un arbre

3.1 Introduction

Dans le paragraphe précédent, nous avons tenté de modéliser la notion d'intermédiarité sur un corpus de manuscrits. Le résultat que l'on obtient est un ensemble de manuscrits et un certain nombre de relations ternaires entre ces manuscrits.

Supposons que le résultat soit le suivant : { F,G,I,P,Q } avec les relations (G, F, I), (G, F, P), (G, F, Q), (F, I, Q), (G, I, Q), (P, I, Q) et toutes les symétriques, alors l'arbre correspondant est :



(Arbre non orienté associé aux données)

On peut constater qu'il n'y a ici qu'un seul arbre non orienté répondant aux données. Nous pouvons nous demander si l'on peut toujours représenter des relations ternaires par un arbre ?

3.2 Un peu de théorie

Pour la représentation du stemma, nous utilisons la notion d'arbre ainsi que toutes les notations qui lui sont liées définies par Barthélemy et Guénoche [1]. Les manuscrits sont représentés par les sommets et les arêtes sont les relations de filiation. Dans un article, Defays [2] établit les conditions nécessaires et suffisantes pour pouvoir représenter par un arbre un ensemble de relations d'intermédiarité.

Définition 4 *Soit Co un ensemble fini et ( ) une relation ternaire sur A. ( ) est une relation ternaire d'intermédiarité ssi* $\forall a, b, c \in Co$, *les axiomes (i) et (ii) sont satisfaits.*

(i) (a, b, c) *et* $(b, a, c) \Leftrightarrow a = b$

(ii) si (a, b, c) *alors* (c, b, a)

Théorème 1 *Soit ( ) une relation ternaire d'intermédiarité sur Co et les axiomes suivants.*

(i) $\forall a, b, c, d \in Co\ (a, b, c) \Rightarrow\ (a, b, d)$ *ou* (d, b, c)

(ii) $\forall a, b \in Co$ *Soit* $\exists c$ *tel que* (c, b, a) *sinon* $\forall d\ (a, b, d)$ *ou* (b, a, d)

(iii) $\forall a, b, c \in Co\ \Rightarrow\ (a, b, c)$ *ou* (b, a, c) *ou* (a, c, b)

Si (i), (ii), (iii) sont satisfaits, une représentation par un arbre linéaire (aussi appelé « chaîne ») est possible. Si (i), (ii) sont satisfaits, une représentation par un arbre est possible avec comme correspondance un sommet de l'arbre pour un élement de Co. Si (i) est uniquement satisfait, une représentation par un arbre est possible avec plus de sommets que d'éléments de Co.

Dans le cadre de nos manuscrits, il suffit de vérifier le (i) pour savoir si notre corpus Co est représentable par un arbre. Cela nous impose un algorithme avec une complexité en n^4 pour vérifier cette faisabilité. Ensuite, il faut réaliser un algorithme de reconstruction d'arbre à partir de nos relations.

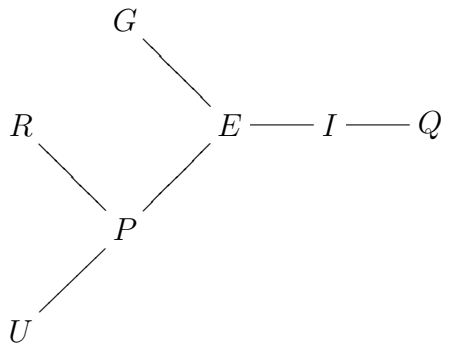
3.3 Algorithme et exemples

Algorithme 1 *Soit Co l'ensemble des manuscrits et B l'ensemble des relations d'intermédiarités*

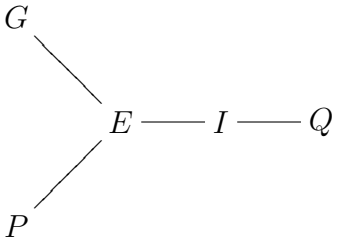
1. *On vérifie la compatibilité de B avec un arbre, il faut que le (i) du Th 1 soit satisfait.*
2. *On détermine ensuite l'ensemble F des manuscrits qui ne sont pas des intermédiaires.*
3. *Pour chaque sommet i de F, on recherche le nœud k tel que l'on n'a pas de j intermédiaire entre i et k. On conserve alors les arêtes ik.*
4. *On enlève alors les manuscrits F de Co et les relations contenants des manuscrits de F de B*
5. *On réitère alors les étapes précédentes à partir de l'étape 2 jusqu'à ce que B soit vide.*
6. *Il nous reste alors N sommets dans Co.*
   - *Si N=1, on a un arbre que l'on peut reconstruire avec l'ensemble des arêtes conservées.*
   - *Si N=2 et qu'une relation d'intermédiarité contient les 2 sommets, on les relie et on peut aussi de reconstruire notre arbre.*
   - *Sinon, il faut rajouter des sommets et des arêtes ou des relations si l'on veut reconstruire un arbre.*

Afin de mieux comprendre l'algorithme, voici 3 exemples qui prennent en compte les 3 sous-cas précédents. Le premier exemple correspond au cas N=1, le deuxième au cas N=2 et le dernier au cas où il faut rajouter des arêtes.

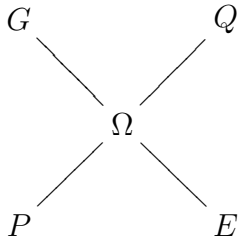
Exemple n°1 : Soient l'ensembles des manuscrits Co={ E,G,I,P,Q,R,U } avec les relations suivantes : (R,P,U) , (R,P,E) , (R,P,G) , (R,P,I) , (R,P,Q) , (U,P,E) , (U,P,G) , (U,P,I) , (U,P,Q) , (R,E,G) , (R,E,I) , (R,E,Q) , (U,E,G) , (U,E,I) , (U,E,Q) , (P,E,G) , (P,E,I) , (P,E,Q) , (G,E,I) , (G,E,Q) , (R,I,Q) , (U,I,Q) , (P,I,Q) , (G,I,Q) , (E,I,Q) et toutes les symétriques. On a alors F={ G,Q,R,U } car ils ne sont pas intermédiaires et l'on conserve les arêtes suivantes $G-E$, $Q-I$, $R-P$ et $U-P$. Après avoir enlevé les manuscrits de F, on obtient au 2ème tour Co={ E,I,P } avec la seule relation (P,E,I) et sa symétrique. Donc F={ I,P } tandis que les arêtes sont $I-E$ et $P-E$.
Enfin, au 3ème et dernier tour, on a Co={ E} sans relation.
En reconstituant l'arbre on obtient :



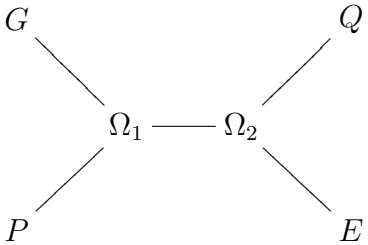
Exemple n°2 : Soient l'ensemble des manuscrits Co={ E,G,I,P,Q}, les relations suivantes : (G, E, I), (G, E, P), (G, E, Q), (E, I, Q), (G, I, Q), (P, I, Q) et toutes les symétriques. On a ici l'ensemble des intermédiaires F={ G,P,Q} et les arêtes associées $G - E$, $P - E$, $Q - I$.
Après avoir enlevé les manuscrits de F, on obtient au 2ème tour Co={ E,I} sans relation. Comme E et I se retrouvent dans la relation (E, I, Q), on les relie et on peut alors construire l'arbre suivant :



Exemple n°3 : Soit l'ensemble des manuscrits Co={E,G,P,Q} sans relation d'intermédiarité : On peut alors envisager les arbres suivants :



Arbre avec un nœud intermédiaire

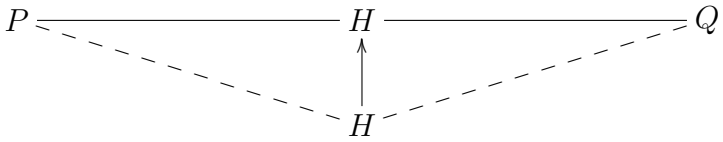


Arbre avec deux nœuds intermédiaires

Nous avons choisi 2 arbres quelconques, mais ce ne sont pas les seules possibilités. On constate aussi que seul, dans ce dernier exemple le (ii) du Th 1, n'est pas satisfait. Si l'on veut malgré tout inférer notre stemma, il va falloir compléter l'arbre par des sommets supplémentaires correspondants aux manuscrits disparus. jusqu'à ce que le (ii) soit satisfait. Trois méthodes sont envisagées selon la modélisation de l'intermédiarité choisie.

- Pour la relation métrique d'intermédiarité, on dispose d'une distance. Rien n'empêche alors d'utiliser les algorithmes de construction d'arbre à partir de distance additive comme ADTREE de Sattah et Tversky [10], les Groupements de Luong [4] ou NJ de Saitou et Nei [9]...
- Dans le cas où l'intermédiarité est définie par un score, une première méthode a été visée. Elle consiste à augmenter l'indice d'intermédiarité minimum afin de faire en sorte que les manuscrits presque intermédiaires deviennent intermédiaire.

Exemple : Ici H est presque intermédiaire entre P et Q, on suppose qu'il est réellement manuscrit intermédiaire.



– Enfin, pour l'intermédiarité en théorie des ensembles, on peut regrouper les manuscrits qui ont la plus grande intersection (le plus grand nombre de variantes communes) et considérer qu'ils sont issus d'un même manuscrit perdu que l'on rajoute.

Dans les 3 cas, une fois le corpus complété, il faut recalculer les relations d'intermédiarité, et relancer l'algorithme 1.

4 Applications

4.1 Méthodes

Précédemment, nous avons envisagé trois méthodes de reconstruction d'arbres à partir de corpus. Le schéma (cf. fig 4) nous résume les différentes possibilités. Trois types d'arbres sont proposés dans le schéma et décrits par Barthélemy et Guénoche [1]. Nous utilisons surtout les arbres hiérarchiques pour leurs ressemblances avec les stemmae codicum classiques.

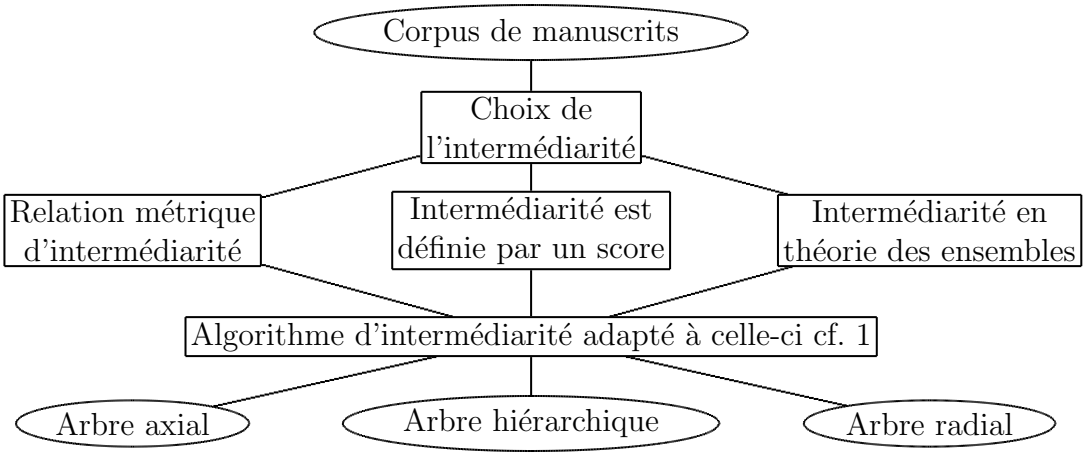


FIG. 4 – Méthodes de reconstruction d'arbres par l'intermédiarité

4.2 Application au corpus de textes : Tertullianus

Afin de réaliser les premières expérimentations, nous avons utilisé le corpus Tertullianus, De Pallio de Marie Turcan [11]. En plus de la collation[1] des textes sous forme d'apparat critique[2] (cf. fig 5), Marie Turcan nous propose un stemma (cf. fig 6). Nous avons utilisé les différentes méthodes proposées précédemment pour reconstruire le stemma que

[1]Dans le travail philologique, la collation consiste à relever les variantes d'un texte afin de permettre l'établissement de celui-ci. Ces variantes proviennent généralement de l'auteur lui-même, de copistes ou d'éditeurs.

[2]L'ensemble constitué par les indications de variantes et les éventuelles notes associées constitue ce que l'on appelle «l'apparat critique» du texte.

nous comparons avec celui proposé. Afin de permettre une comparaison plus aisée, nous nous sommes limités aux manuscrits suivants : Co={ F,G,L,N,R1,R2,R3,S,V,X}

Ligne 1 *post* uiri *add.* affrice [-ca *N*] uiri *NFX* ||
Ligne 3 annonae *LatPC Rig* : annona et *mssR Salm* ||
Ligne 5 uobis *NR* : nobis *FX* || olim : enim *N* ||
Ligne 6 subteminis : -tegminis *NR* || concilio *Iun Salm* : consilio *mssR* || mensurae R^3 *Salm* : mensura e *X* mensura et $N^{pc}FR^1$ ||
Ligne 7 crura : cura N^{ac} || nec intra : ne intra *X* || inuerecundae R^3 *Salm mss* : uerec- R^1 ||
Ligne 8 parae *testes* : parce V^{pc} *edd. praet. Bu* ||

FIG. 5 – **Exemple d'apparat critique** de Marie Turcan

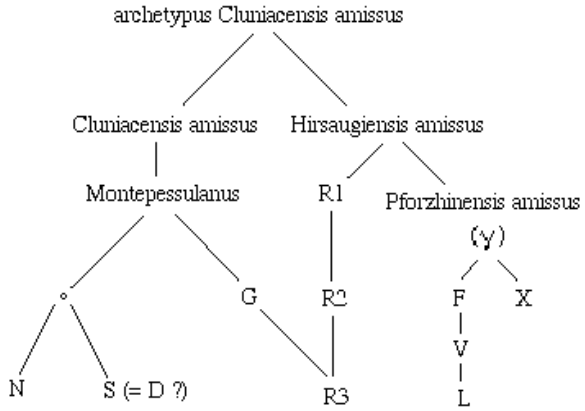


FIG. 6 – **Stemma codicum** Adapté du stemma établi par J.-C. Fredouille

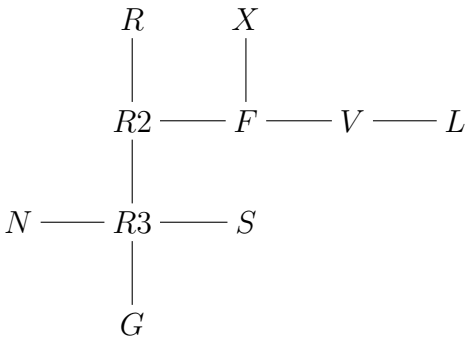
Après une première itération de l'algorithme avec les scores, il s'avère qu'aucun manuscrit n'est un intermédiaire. Nous décidons alors d'augmenter l'indice d'intermédiarité minimum. Après quelques essais, les relations suivantes apparaissent :

$$(L,F,X),(V,F,X),(F,V,L),(L,V,X),(G,R2,R1),(R3,R2,R1),(G,R3,R1),(G,R3,R2)$$

ce qui correspond aux 2 chaînes suivantes :

$$G \text{ —— } R3 \text{ —— } R2 \text{ —— } R1 \qquad\qquad L \text{ —— } V \text{ —— } F \text{ —— } X$$

Et après trois itérations, le (ii) du Th 1 est satisfait et l'on obtient finalement l'arbre suivant :



Le résultat obtenu n'est pas convaincant. Si l'augmentation de l'indice d'intermédiarité est intéressant au début, rapidement l'algorithme trouve des intermédiarités partout, même où il n'y en a pas (Ex. $(R2,F,X)$). C'est donc à l'éditeur de choisir les limites de ce qu'il estime raisonnable lors de la construction du stemma.

Dans le cas de l'utilisation de l'algorithme associé à la relation métrique d'intermédiarité, une seule relation apparaît, c'est (F,V,L). Finalement, c'est la méthode NJ qui permet

d'inférer l'arbre en ne concevant de F,V et L que le manuscrit le plus proche de tous les autres, ici le F. Au vu du peu d'intermédiarité détectée au cours de l'algorithme, c'est

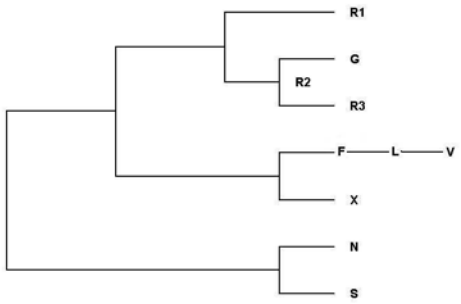


FIG. 7 – **Arbre obtenu par l'intermédiarité, NJ et TreeView [12]**

surtout NJ qui permet d'obtenir l'arbre dans cet exemple. L'arbre obtenu est en revanche relativement proche du stemma, ce qui laisse présager l'intérêt de la méthode.

La troisième méthode n'ayant pas encore été testée sur ce corpus, les résultats sont en attente.

5 Conclusion

Les méthodes d'intermédiarité constituent un outil indispensable dans la classification de tous les documents. Il est toujours intéressant de savoir qu'un objet s'intercale entre deux autres. Dans le cadre de documents textuels, le classement peut s'effectuer selon plusieurs axes : les thèmes, les auteurs, la grammaire, le style, les dates... Dans tous ces cas, il faut définir l'intermédiarité pour positionner notre document parmi les autres.

Dans l'édition critique, et plus précisément dans la reconstitution de l'histoire d'un texte et sa schématisation (Stemma codicum), nous avons tenté de modéliser la notion de « copie intermédiaire ». On constate que cette modélisation permet de retrouver les intermédiaires dans le cas d'un corpus réel, mais actuellement, seul un expert (l'éditeur) sait quand interrompre le programme pour dénaturer la relation ternaire obtenue.

La reconstruction de l'arbre est un autre problème. Soit on ne dispose pas d'assez de relations pour inférer totalement l'arbre (c'est le problème des manuscrits perdus qui sont parfois trop nombreux). Soit le nombre de relations à vérifier est tellement important que le programme se trouve ralenti et l'algorithme difficile à mettre en place dans le cas d'un corpus plus important (> 100 mns).

C'est sans doute vers la diminution du nombre de relations ternaires que se trouve une solution. En effet, La suppression de relations pour obtenir une sorte de base minimum (« re-déployable » par transitivité [6] et symétrie) permettrait d'améliorer l'algorithme et de simplifier le problème.

Un autre point d'intérêt se situe dans l'orientation des relations. Non seulement elle permettrait d'orienter le stemma et donc de déterminer la racine de notre arbre (ici le manuscrit original encore appelé « archétype »), mais en plus, elle simplifierait énormément la reconstruction de l'arbre.

Enfin, c'est peut être l'intérêt majeur de cette modélisation, elle permet de visualiser la contamination[3] d'un corpus. Si l'on examine la figure 1, on constate un trait en pointillés

[3]Contamination : copie d'un manuscrit sur plusieurs modèles (syn : corruption, hybridation)

entre les manuscrits H et S. Cela signifie que le manuscrit S a été copié à partir de (M) mais aussi à partir de H. Notre stemma n'est plus un arbre (il y a un cyle M,S,H) que l'on observe par les intermédiarités (M, S, H), (S, H, M) et (H, M, S).

Références

[1] J. P. Barthélemy et A. Guénoche, "Les Arbres et les Représentations des Proximités", *Masson*, 1988

[2] D. Defays, "Tree representation of ternary relations", *Journal of Mathematical Psychology*, 19, 1979, 208-218

[3] V. I. Levenshtein, "Binary Codes capable of correcting deletions, insertions, and reversals", *Soviet Physics - Doklady*, 10, 1966, 707-710

[4] X. Luong, "Méthodes d'analyse arborée. Algorithmes. Applications", *Thèse de doctorat, Paris V*, 1988

[5] K. Menger, "Untersuchungen über allgemeine Metrick", *Mathematische Annalen*,100, 1928, 75-163

[6] E. Pitcher et M. F. Smiley, "Transitivities of Betweenness ", *Transactions of the American Mathematical Society*, 52, 1942, 95-114

[7] H. Quentin, "Essais de critique textuelle", *Picard*, 1926

[8] F. Restle, "A metric and an ordering on sets", *Psychometrika* 24, 1959, 207-220.

[9] N. Saitou et M. Nei, "The neighbor-joining method : a new method for reconstructing phylogenetic trees", *Mol Biol Evol* 4, 1987, 406–425.

[10] S. Sattah et A. Tversky, "Additive similarity trees", *Psychometrika* 42, 1977, 319–345.

[11] M. Turcan, "Tertullien, De pallio. Introduction, édition critique, traduction française, commentaire et index, mis en ligne en mars 2006", *à paraître dans la collection Sources Chrétiennes, Editions du Cerf, fin* 2007

[12] Roderic D. M. Page, "Logiciel Treeview, A phylogenetic tree viewer"

FactoMineR, une librairie de fonctions R en analyse des données pour l'enseignement et la recherche

S. Lê, J. Josse, F. Husson

Agrocampus, IRMAR, 65 rue de Saint-Brieuc, 35042 Rennes
(Sebastien.Le, Julie.Josse, Francois.Husson)@agrocampus-rennes.fr

Mots clés : analyse factorielle, groupes de variables, hiérarchie sur les variables, groupes d'individus, interface graphique.

Résumé : la librairie de fonctions `FactoMineR` permet le traitement de données multidimensionnelles. Ce package prend en compte différents types de variables (qualitative et/ou quantitative), différents types de structure sur les données (structure sur les individus ou les variables ou hiérarchie sur les variables). En plus des principales méthodes d'analyse des données (analyse en composantes principales, analyse factorielle des correspondances, analyse des correspondances multiples), des méthodes avancées (analyse factorielle multiple, analyse procrustéenne généralisée, analyse factorielle multiple hiérarchique, analyse factorielle multiple duale) sont également implémentées dans cette librairie. `FactoMineR` fournit des indicateurs et des sorties graphiques soignées et facilement exportables. Une interface graphique permet de plus de rendre accessible cette librairie de fonctions à des non utilisateurs du langage R.

1 Introduction

Nous présentons une librairie de fonctions dédiée à l'analyse de données multidimensionnelles et disponible sous le logiciel gratuit R. Cette librairie s'appelle FactoMineR ([3]). Les raisons qui nous ont amenées à développer ce package sont les suivantes :

- rendre disponibles et conviviales les méthodes classiques d'analyse des données dans un logiciel libre (facilitant ainsi l'accès aux méthodes d'analyse des données pour les étudiants en stage);
- diffuser des méthodologies avancées.

Les méthodes classiques implémentées sont l'Analyse en Composantes Principales (ACP), l'Analyse Factorielle des Correspondances (AFC), l'Analyse des Correspondances Multiples (ACM), la description de groupes d'individus (ou de classes). Les méthodes avancées succinctement présentées ci-après sont : l'Analyse Factorielle Multiple (AFM), l'Analyse Procrustéenne Généralisée (APG), l'Analyse Factorielle Multiple Hiérarchique (AFMH), l'Analyse Factorielle Multiple Duale (AFMD) et l'Analyse Factorielle de Données Mixtes (AFDM).

Il nous a semblé important de faciliter l'utilisation des méthodologies proposées par l'intermédiaire d'un menu déroulant. Ainsi, des utilisateurs non familiers à l'environnement R, pourront facilement paramétrer les fonctions par simples clicks et obtenir les résultats des différentes analyses (sur les individus, les variables, les groupes de variables) ainsi que des aides à l'interprétation (qualités de représentation, contributions). Par ailleurs, `FactoMineR` permet d'obtenir des graphiques soignés et facilement exportables.

2 Présentation de quelques méthodes avancées

Dans le package `FactoMineR`, il est possible de prendre en compte différentes structures sur le jeu de données. Les données peuvent être organisées en groupes d'individus, en groupes de variables, ou selon une hiérarchie sur les variables.

Groupes de variables : le point de vue de l'AFM. Plusieurs méthodes sont disponibles pour analyser des données structurées en groupes de variables (l'AFM, [1] ou l'APG, [2]). L'AFM permet de traiter des données avec des variables quantitatives et/ou qualitatives, contrairement à l'APG qui ne permet d'analyser que des variables quantitatives. L'AFM permet de plus de représenter des groupes de variables supplémentaires afin d'aider à l'interprétation.

Hiérarchie sur les variables : l'analyse factorielle multiple hiérarchique (AFMH). Dans de nombreux jeux de données, les variables sont structurées selon une hiérarchie entre des groupes ou sous-groupes de variables. Ceci est classique dans le traitement d'enquêtes structurées en plusieurs thèmes et sous-thèmes. Analyser de telles données impliquent d'équilibrer le rôle de chaque thème, mais également de chaque sous-thème à l'intérieur de chaque thème. Pour cela, il est nécessaire de considérer une hiérarchie sur les variables. L'AFMH ([4]) considère une telle structure sur les variables dans une analyse globale qui équilibre les groupes de variables à chaque niveau de hiérarchie.

Groupes d'individus : l'analyse factorielle multiple duale (AFMD). L'AFMD ([5]) est une extension de l'AFM quand les individus sont structurés en groupes. Le cœur de la méthode est une analyse factorielle interne, en référence à l'analyse des correspondances interne, pour laquelle les données sont systématiquement centrées par groupe. Cette analyse fournit une représentation superposée des L nuages de variables associés aux L groupes d'individus ainsi qu'une représentation des matrices de corrélations associées chacune à un groupe d'individus.

3 Interface graphique de `FactoMineR`

Il est possible de rajouter facilement un menu déroulant `FactoMineR` à un environnement déjà existant nommé `Rcmdr`. Pour cela, il suffit de lancer, dans R, la ligne de commande suivante :

```
> source("http://factominer.free.fr/install-facto.r")
```

L'interface graphique correspondant à `FactoMineR` est très conviviale et permet de sauvegarder l'ensemble des résultats dans un fichier `*.csv`, de construire des graphiques directement présentables, mais également de récupérer le code de la fonction permettant de générer les analyses (ceci est utile si on souhaite relancer les analyses).

À titre d'exemple, nous présentons le menu de l'ACP (Fig. 1).

La fenêtre principale de l'ACP permet de sélectionner les variables actives (par défaut, toutes sont actives). Différents boutons permettent d'ajouter des variables supplémentaires (qualitatives ou quantitatives), de sélectionner des individus supplémentaires, de choisir les sorties et de paramétrer les graphes.

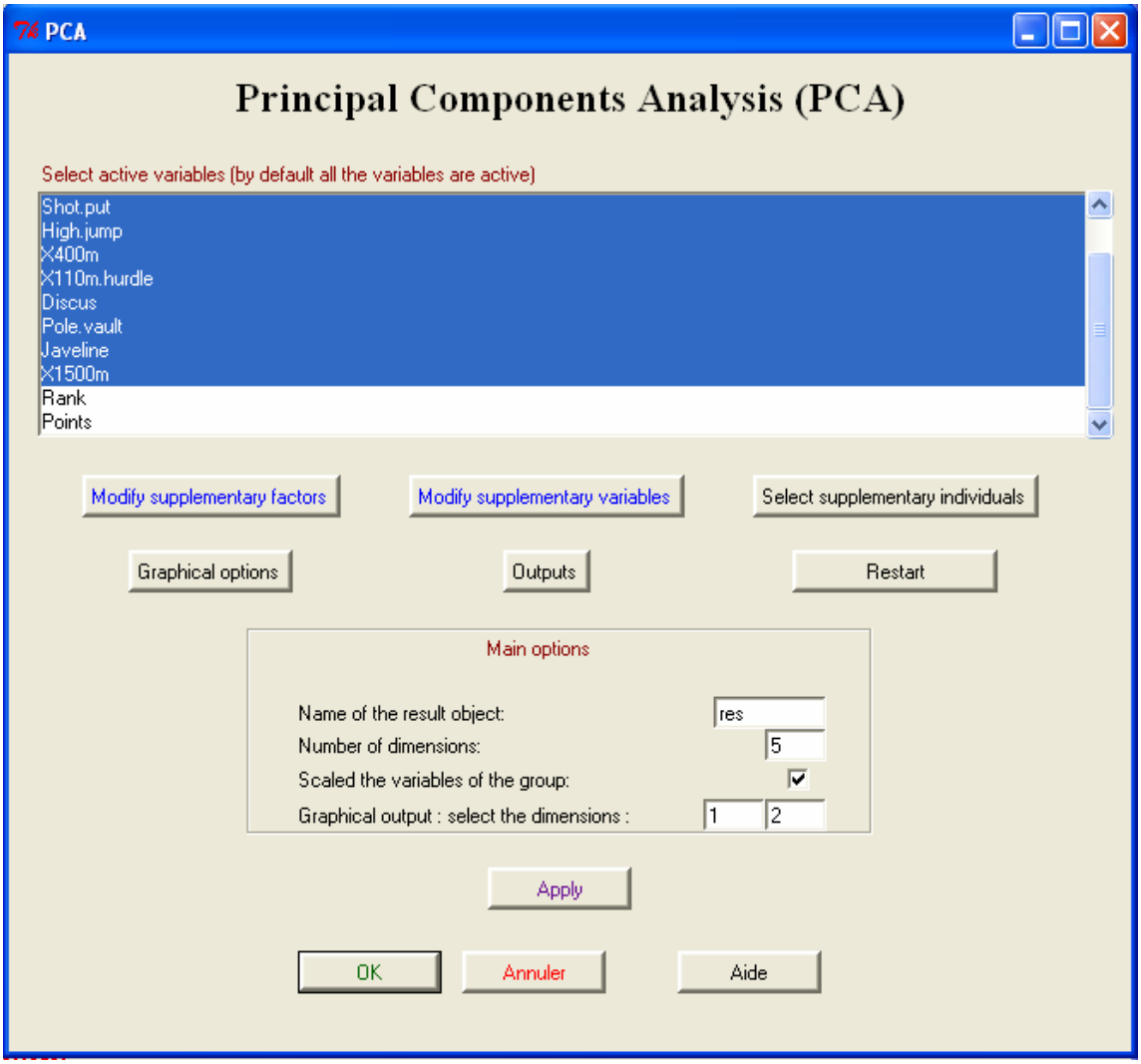


Figure 1: Fenêtre principale de l'ACP

Les options graphiques (Fig. 2) concernent le graphe des individus et le graphe des variables. Pour les individus, il est possible de représenter les individus actifs, les individus illustratifs et les modalités des variables qualitatives illustratives; il est également possible de choisir les éléments que l'on souhaite représenter. Les individus peuvent être coloriés en fonction d'une variable qualitative (les variables qualitatives sont proposées dans une liste).

Pour les variables, il est possible de les représenter en fonction de leur qualité de représentation. Les variables actives et illustratives sont de couleurs différentes.

Le site `http://factominer.free.fr/` donne des exemples pour illustrer les différentes méthodes disponibles dans la librairie.

[1] B. Escofier et J. Pagès, "Analyses factorielles simples et multiples.", *Dunod*, 1998.
[2] J. C. Gower, "Generalized Procrustes Analysis.", *Psychometrika*, 40, 1975, 33-51.

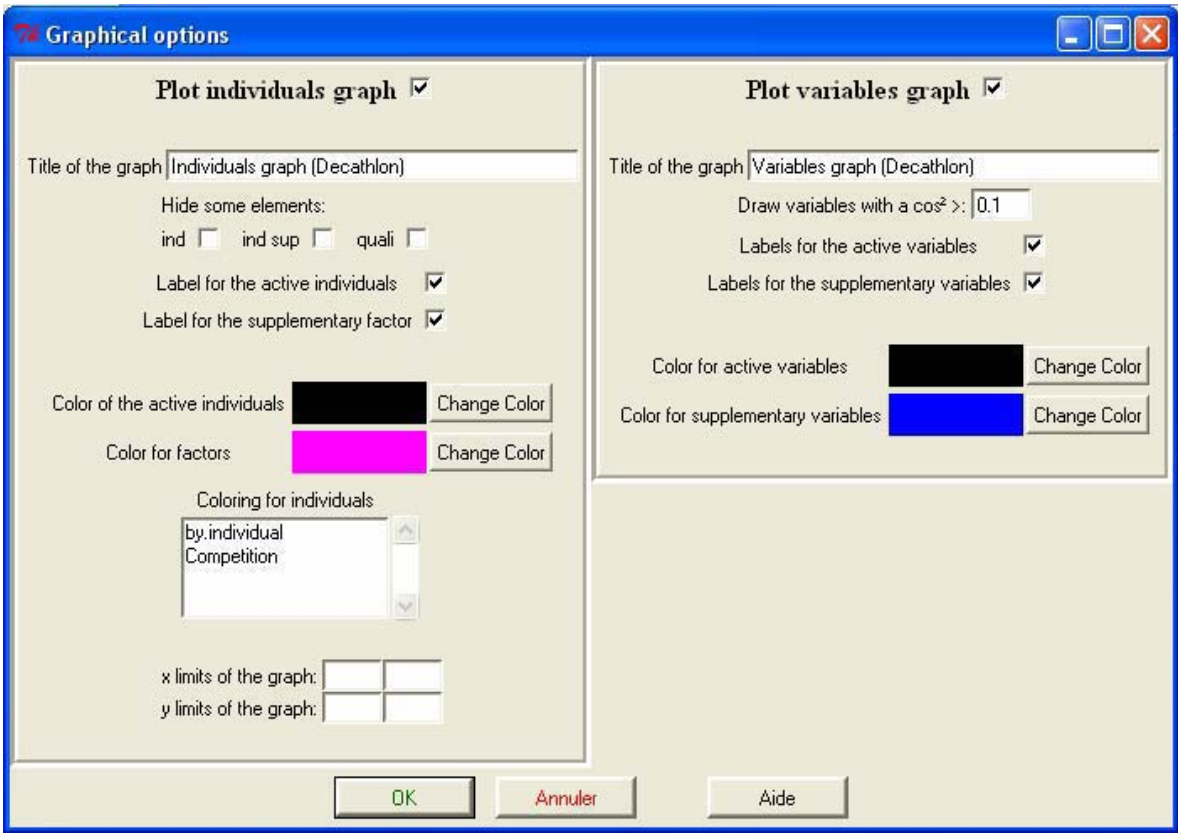


Figure 2: Fenêtre des options graphiques de l'ACP

[3] F. Husson, S. Lê et J. Mazet, "FactoMineR: Factor Analysis and Data Mining with R.", 2007, `http://factominer.free.fr`.

[4] S. Le Dien et J. Pagès, "Hierarchical Multiple Factor Analysis: application to the comparison of sensory profiles.", *Food Quality and Preference*, 14, 2003, 397-403.

[5] S. Lê et J. Pagès, "DMFA: Dual Multiple Factor Analysis.", *12th International Conference on Applied Stochastic Models and Data Analysis*, 2007.

[6] R Development Core Team, "R: A Language and Environment for Statistical Computing.", 2005, Vienna, Austria. `http://www.R-project.org`, 2006.

Médianes et unanimité dans les treillis

Bruno Leclerc

CAMS, EHESS, 54 bd Raspail, 75270 Paris cedex 06
leclerc@ehess..fr

Mots clés : Arbres, graphes et classification, Optimisation en classification.

Étant donné un espace métrique (E, d) et un p-uple (*profil*) $\pi = (x_1, ..., x_p)$ d'éléments de E, une *médiane* est un élément m de E minimisant l'*éloignement* $e(m) = \sum_i d(m,x_i)$. La *procédure médiane* est la méthode d'aggrégation qui associe à tout profil de E sa ou ses médianes. Elle a été considérée dans de nombreux problèmes de consensus ou de centralité correspondant à des domaines d'application et des métrique très divers (Barthélemy et Monjardet 1981, 1988). On s'intéresse ici au cas où E est muni d'une structure latticielle (treillis, ou parfois demi-treillis), et où d est une distance de plus court chemins dans le graphe (valué ou non) de couverture du treillis E.

Rappelons quelques situations où des problèmes d'agrégation dans des treillis de types variés sont apparus :

- treillis distributifs des parties d'un ensemble donné X, des tournois sur X, des dissimilarités sur X, de produits de chaînes, …
- demi-treillis à médianes des cliques d'un graphe donné, des hiérarchies sur X, des préordres totaux sur X, …
- demi-treillis distributifs des parties de X à au plus k éléments, …
- treillis (géometrique) des partitions de X, des partitions sous contraintes de connexité dans un certain graphe sur X, …
- treillis semimodulaires inférieurement des ultramétriques sur X, …
- géométries convexes des familles de Moore sur X, des intervalles d'un ordre total sur X, des sous-arbres d'un arbre donné, d'autres familles de convexes, demi-treillis des ordres sur X, …
- autre treillis : préordres sur X, treillis permutoèdre des ordres sur X, …

La question abordée dans cet exposé est la suivante. On considère une médiane m d'un profil π, et un élément x de E tel que l'on a $x \leq x_i$, pour tout $i = 1,\ldots, p$, où $\leq$ est l'ordre du treillis E. En d'autres termes, on a $x \leq \wedge_i x_i$. Est-on alors assuré de retrouver cette propriété pour la médiane m, c'est à dire d'avoir encore $x \leq m$? Si cela est vrai pour tout profil et toute médiane, on dit que la procédure médiane vérifie la propriété d'unanimité (ou de Pareto) dans l'espace métrique (E, d) considéré. Le cas contraire peut être vu comme paradoxal puisqu'une propriété partagée par tous les éléments du profil ne se retrouve alors pas dans l'élément m censé résumer ce profil.

En fait, divers résultats et contre-exemples permettront de montrer que, si la propriété de Pareto est vérifiée dans de nombreux cas par la procédure médiane, elle n'est pas générale et dépend largement de la structure du treillis et de la métrique considérées.

[1] J.-P. Barthélemy, B. Leclerc, "The median procedure for partitions", in I.J. Cox, P. Hansen, and B. Julesz, eds., *Partitioning data sets, DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 19, 1995, 3-34.
[2] J.P. Barthélemy, B. Monjardet, "The median procedure in cluster analysis and social choice theory", *Mathematical Social Sciences* 1, 1981, 235-268.
[3] J.P. Barthélemy, B. Monjardet, "The median procedure in data analysis : new results and open problems", in: H.H. Bock, ed., *Classification and Related Methods of Data Analysis*, North-Holland, Amsterdam, 1988, 309-316.
[4] B. Leclerc, "Medians and majorities in semimodular lattices", *SIAM J. on Discrete Math.* 3, 1990, 266-276.
[5] B. Leclerc, "Lattice valuations, medians and majorities", *Discrete Math.* 111, 1993, 345-356.
[6] B. Leclerc, "Medians for weight metrics in the covering graphs of semilattices", *Discrete Applied Math.* 49, 1994, 281-297.
[7] B. Leclerc, "The median procedure in the semilattice of orders", *Discrete Applied Math* 127, 2003, 241-269.
[8] Li Jinlu, "Singular points and an upper bound of medians in upper semimodular semilattices", preprint, Dept of Mathematics, Shawnee State University, 1996.
[9] B. Monjardet, "Théorie et applications de la médiane dans les treillis distributifs finis", *Annals of Discrete Mathematics* 9, 1980, 87-91.
[10] S. Régnier, "Sur quelques aspects mathématiques des problèmes de classification automatique", *ICC Bull.* 4, 1965, 175-191, repr. *Math. Sci. hum.* 82, 1983, 13-29.

Sur les différentes expressions formelles d'une hiérarchie binaire symétrique ou implicative

I.C. Lerman

IRISA, Campus Universitaire de Beaulieu, 35042 Rennes Cédex
lerman@irisa.fr

Mots clés : classification ascendante hiérarchique, arbre implicatif, hiérarchie binaire.

La Classification Ascendante Hiérarchique (CAH) est un outil puissant d'analyse des données. Elle a connu un développement fulgurant ces trente dernières années. Face à un tableau des données décrivant un ensemble $\mathcal{O}$ d'objets élémentaires (resp., $\mathcal{C}$ de catégories) par un ensemble $\mathcal{V}$ de variables (attributs ou descripteurs), la CAH permet l'organisation en classes et sous classes de proximité aussi bien de l'ensemble $\mathcal{O}$ des objets (resp., $\mathcal{C}$ de catégories) que de l'ensemble $\mathcal{V}$ des variables descriptives. L'analyse des données qui en résulte suppose la reconnaissance de classes ou sous-classes "significatives" issues de l'une ou de l'autre des des deux classifications hiérarchiques duales. Elle suppose également la mise en correspondance de ces deux dernières.

Entre parties disjointes de l'ensemble E à traiter, on établit un indice de dissimilarité δ dépendant de la description fournie par le tableau des données. δ a un caractère parfaitement symétrique par rapport aux deux arguments à comparer.

La notion intuitive de hiérarchie implicative est apparue au début des années 90 ([2]). Sa construction est en tout point analogue à celle d'une CAH ; mais à une différence fondamentale près que nous allons préciser. Si $\mathcal{A}$ désigne l'ensemble à organiser ($\mathcal{A}$ est généralement un ensemble d'attributs booléens), $\mathcal{A}$ est muni d'un indice de similarité orientée (un indice d'implication) σ qui est ensuite étendu à la comparaison de deux segments ordonnés et disjoints de $\mathcal{A}$.

L'organisation synthétique des résultats que la CAH procure est un arbre indicé de classifications qui s'emboîtent sur E. Chaque classification occupe un niveau de l'arbre. L'interprétation de l'expert repose pour l'essentiel sur une indexation de la suite croissante des niveaux de l'arbre au moyen d'une section commençante de la suite des entiers. Le principe de la construction de l'arbre est binaire : une classe apparaissant à un niveau donné, résulte de la fusion de deux sous classes filles qui sont déjà apparues à des niveaux inférieurs. Ces deux dernières sont repérées comme étant les plus proches au sens de l'indice δ. Il arrive souvent qu'on propose une indexation numérique des niveaux de l'arbre à partir d'un indice de stratification qui correspond à la valeur de δ pour les deux sous classes filles dont la jonction se produit au niveau concerné. Nous nous limitons quant à nous à une indexation ordinale parce que, pour une part, comme nous venons de le mentionner, c'est cette indexation qui joue le rôle le plus important dans l'interprétation de l'expert. D'autre part, cette forme ordinale d'indexation permet une plus grande clarté formelle dans le passage entre une hiérarchie binaire symétrique et un arbre implicatif.

Il existe différentes expressions formelles de la structure dégagée par une CAH classique : hiérarchie de parties (indicée ou non), arbre de classification (indicé ou non),

distance ultramétrique.

La formalisation proposée dans ([3]) et ([1]) de la structure dégagée par une hiérarchie implicative s'apparente à celle d'une hiérarchie binaire non indicée de parties. Dans ([4]) nous avons repris la formalisation avec un éclairage nouveau d'une façon argumentée et nourrie en faisant le parallèle le plus étroit possible avec la notion de classification hiérarchique binaire. Ainsi, on se rend clairement compte de ce qui change en passant du cas symétrique au cas orienté. C'est précisément l'objet de notre présentation.

À cette fin et de façon systématique le cas classique est repris en rappelant ou en précisant des définitions telles que celles de hiérarchie binaire (non indicée ou indicée), d'arbre de classification binaire (non indicé ou indicé) ou de distance ultramétrique. On supposera - comme évoqué ci-dessus - que l'indexation ou la valeur de la distance ultramétrique décrit une section commençante de l'intervalle $\mathbb{N}$ des entiers. Nous introduisons une formalisation - nouvelle à notre connaissance - en termes de hiérarchie de fourches binaires. En effet, une telle expression formelle est particulièrement adaptée au transfert entre la notion de hiérarchie binaire symétrique et la notion de hiérarchie implicative ; la notion de fourche binaire symétrique donnant une notion de fourche binaire orientée.

Toujours dans le cas symétrique, nous précisons les règles de passage entre les différentes expressions formelles. Ainsi, on a une équivalence de représentation entre une hiérarchie binaire (non indicée) de parties et une hiérarchie de fourches binaires symétriques. On a également une équivalence de représentation entre une hiérarchie binaire indicée, un arbre binaire de classification indicé et la distance ultramétrique associée. Cependant, la question se pose de faire correspondre de façon compatible, à une hiérarchie binaire non indicée de parties d'un ensemble E, que nous notons ici $\mathcal{H}_b(E)$, un arbre de classification binaire indicé ordinalement sur E. Il y a plusieurs solutions à ce problème. Elles sont énumérables. Nous proposons précisément un algorithme pour réaliser - de proche en proche - l'une quelconque des solutions possibles. Cet algorithme réalise un "accrochage" de ce que nous appelons les "chaînes complètes" de la hiérarchie binaire. Une telle chaîne complète est une suite orientée par inclusion de parties de $\mathcal{H}_b(E)$, telle qu'il n'existe pas de parties de $\mathcal{H}_b(E)$, strictement comprise entre deux parties consécutives et telle que le premier (resp., le dernier) élément est un singleton (resp., la partie pleine E).

Dans ces conditions, la transposition de la formalisation dans le cas implicatif ou orienté se déduit de façon claire et naturelle. Comme exprimé ci-dessus, ce qui se transporte le plus directement est la notion de hiérarchie de fourches binaires. Ces dernières sont symétriques dans le cas classique. Elles deviennent orientées dans le cas implicatif. De cette dernière (hiérarchie de fourches orientées) dérive un ordre total sur l'ensemble organisé que nous désignerons comme ci-dessus par $\mathcal{A}$. Le correspondant de la hiérarchie binaire de parties de E est une hiérarchie binaire d'intervalles de $\mathcal{A}$, lequel est totalement ordonné. La notion d'arbre binaire indicé (resp., non indicé), de classifications devient une notion d'arbre binaire indicé (resp., non indicé) orienté de classifications ; chaque jonction entre deux classes devient orientée de gauche à droite. La caractérisation d'une distance ultramétrique est maintenant relative à un triangle orienté de la forme (x, y, z) où le plus petit des vecteurs est soit $\vec{xy}$, soit $\vec{yz}$; les deux autres étant égaux. On a finalement le tableau de correspondance :

Cas symétrique	**Cas orienté**
Fourche binaire symétrique	Fourche binaire orientée
Hiérarchie de fourches binaires symétriques	Hiérarchie de fourches binaires orientées
Hiérarchie binaire indicée (resp.,non indicée) de parties	Hiérarchie binaire indicée (resp.,non indicée) d'intervalles
Chaînes complète de parties	Chaînes complète d'intervalles
Distance ultramétrique	Distance ultramétrique orientée

Nous donnons ci-dessous les graphiques de deux arbres binaires où le premier est symétrique et le second est orienté. Nous avons choisi les mêmes associations pour chacun des deux arbres. Cependant, elles sont orientées pour l'arbre implicatif.

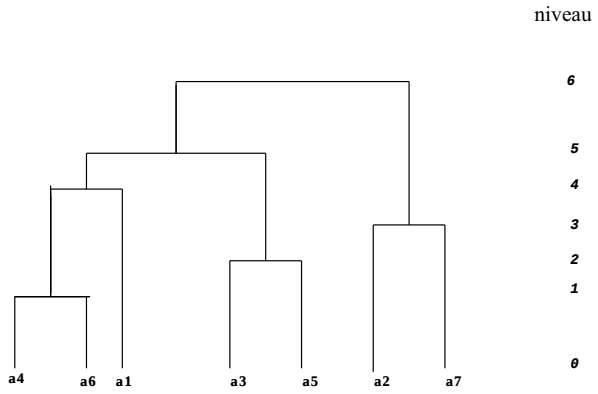


Figure 1: Arbre symétrique de classification binaire

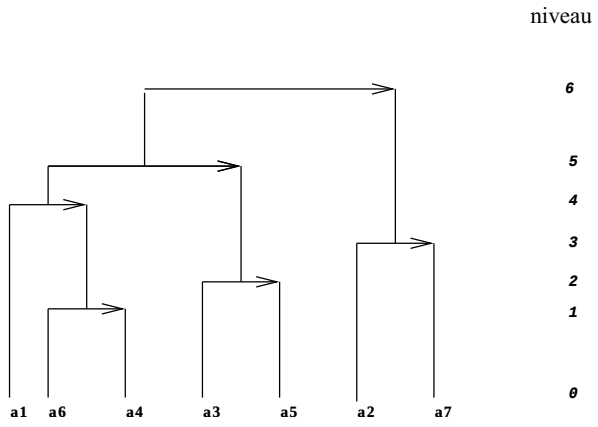


Figure 2: Arbre de classification binaire implicative

Remerciements

Nous remercions Pascale Kuntz (Professeur à l'École Polytechnique de l'Université de Nantes) pour l'échange que nous avons pu avoir autour de [4] et qui nous ont permis de préciser certains aspects.

Références

[1] R. Gras, P. Kuntz, "Discovering r-rules with a directed hierarchy", *Soft Computing* (5): March 2006, 453-460.

[2] R. Gras, A. Larher, "L'implication statistique, une nouvelle méthode d'analyse des données", *Mathématiques et Sciences Humaines* (120), 1993, 5-31

[3] P. Kuntz, "Classification hiérarchique orientée en ASI", In R. Gras, F. Spagnolo and J. David editors, *Troisièmes Rencontres Internationales - A.S.I. Analyse Statistique Implicative*, Università degli Studi di Palermo, 2005, 53-62

[4] I.C. Lerman, "Analyse logique, combinatoire et statistique de la construction d'une hiérarchie implicative ; niveaux et noeuds significatifs", *Publication Interne Irisa* n^0 1827, novembre 2006, 45 pages

Visualisation de clusters dans les espaces de grande dimension

S. Lespinats[1], B. Fertil[2] et J. Hérault[3]

1. INSERM unité U722 et Université Denis Diderot – Paris 7, Faculté de médecine, site Xavier Bichat, 16 rue Henri Huchard, 75870 Paris cedex 18, France
2. UMR INSERM unité 678 - Université Pierre et Marie Curie - Paris 6, boulevard de l'hôpital, 75634 Paris. France
3. Institut National Polytechnique de Grenoble. Laboratoire des images et des signaux. 46, avenue Felix Viallet, 38031 Grenoble cedex
(lespinats@ bichat.inserm.fr, fertil@imed.jussieu.fr, jeanny.herault@inpg.fr.

Mots clés : clusters, grande dimension, rangs de voisinage, visualisation.

Visualisation de données de grande dimension

Quel que soit le domaine d'activité, il est courant de chercher à analyser des données de grande dimension (c'est-à-dire des données décrites par un grand nombre de paramètres). Or, s'il est aisé de visualiser des données de deux dimensions, l'exploration de données de plus grande dimension est évidemment moins immédiate. On fait alors souvent appel à des méthodes de réduction de dimension qui peuvent se baser sur des projections linéaires (comme l'analyse en composantes principales [2, 11]) ou non-linéaires (comme l'analyse en composantes curvilignes [4]).

Pourtant, les espaces de grande dimension possèdent des propriétés particulières (regroupées sous le nom de « fléau de la dimension ») qu'il convient de ne pas ignorer [1]. Parmi ces propriétés (souvent déconcertantes pour notre intuition adaptée aux espaces de deux ou trois dimensions) nous citerons en particulier la « concentration de la mesure » : la différence relative entre les « courtes » et les « grandes » distances se réduit rapidement pour tendre vers 0 lorsque la dimension de l'espace augmente. Cette propriété pose un véritable problème aux méthodes de visualisation classiques, en effet, celles-ci s'appuient en général sur les distances entre données (ou sur les produits scalaires, ce qui revient presque au même).

Nous avons présenté précédemment une méthode de visualisation des données (baptisée DD-HDS pour Data-Driven High Dimensional Scaling) adaptée à ce contexte difficile [7, 8]. Notre méthode se distingue par une fonction pondération qui 1) est de forme sigmoïde s'adaptant à l'histogramme des distances de façon à réellement avantager la représentation des distances courtes malgré la concentration de la mesure et 2) s'appuie sur les distances d'origine ET sur les distances dans la représentation, ce qui permet de pénaliser à la fois les « faux voisinages » ET les « déchirements ». Bien que cette méthode ait montré une efficacité réelle pour la représentation de données de grande dimension, nous avons pu mettre en évidence des jeux de données pour lesquels des clusters manifestes étaient mal séparés dans la représentation.

Les rangs de voisinage

Pour mieux préserver les clusters dans les représentations, nous proposons une méthode de réduction de dimension qui s'appuie sur les « rangs de voisinage ». Nous définissons les rangs de voisinage de la manière suivante :

Une relation entre une donnée et son plus proche voisin est une relation de rang 1.
Une relation entre une donnée et son 2ème plus proche voisin est une relation de rang 2.
etc
Par convention, une relation d'une donnée à elle-même est une relation de rang 0.

RankVisu

La plupart des méthodes de visualisation classiques sont destinées à préserver les distances entre données en donnant l'avantage à la représentation des distances courtes (par exemple, [4, 7, 9, 10]). A cause de la concentration de la mesure, les distances courtes et les distances longues sont en général du même ordre de grandeur dans le cas de données de grande dimension [1]. L'agglomération des clusters sur la représentation n'est alors pas avantagée. C'est pourquoi, au lieu de s'intéresser aux distances, la méthode que nous proposons ici (baptisée RankVisu) s'attache à conserver les rangs de voisinage sans tenir compte des distances dont ils sont issus.

Plusieurs arguments peuvent être avancés pour justifier l'utilisation d'un tel critère :

1) Les rangs de voisinages supportent souvent l'information permettant d'affecter les données aux différents clusters : les méthodes de « k plus proches voisins » utilisent ce même critère [3]. C'est cette propriété qui explique que RankVisu soit aussi efficace dans la mise en valeur des clusters.
2) L'utilisation des rangs rend moins sensible aux valeurs aberrantes et permet de faire face aux cas où les clusters ne sont pas isotropes (on pourrait alors parler d'une visualisation « non paramétrique », c'est-à-dire d'une visualisation ne faisant pas d'hypothèse sur la loi de distribution des données dans les clusters).
3) Les rangs de voisinages ne sont aucunement affectés par la dimension des données. En effet, quelle que soit la dimension il existe un plus proche voisin, un $2^{ème}$ plus proche, etc. Nous échappons ainsi aux problèmes de fléau de la dimension que rencontre les méthodes de visualisation basées sur les distances.

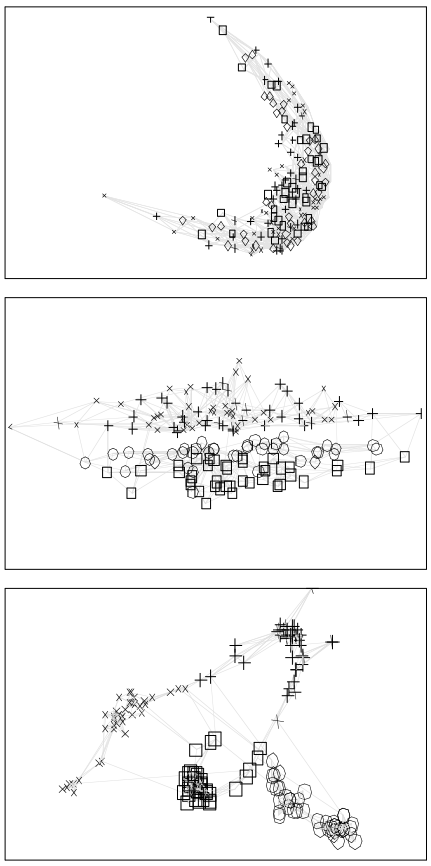
Les résultats que nous obtiendrons ainsi seront à mettre en regard de ceux que l'on peut obtenir par non-metric Multi Dimensional Scaling (ou non-metric MDS) [5, 6]. En effet, cette technique a pour but de conserver dans la représentation l'ordre entre l'ensemble des distances plutôt que les distances elles-mêmes. Ce critère et celui de RankVisu ne sont donc pas très éloignés.

Résultats

Les visualisations de deux jeux de données sont proposées. Le premier est constitué de données simulées de façon à comporter quatre clusters dans un espace de dix dimensions. Une difficulté supplémentaire est apportée par le fait que les clusters sont non isotropes et à égale distance les uns des autres (les distances inter-clusters sont réglées de manière à ce qu'elles soient proches des distances intra-cluster moyenne). Pour obtenir un tel jeu de données, celles-ci sont générées de la façon suivante : chaque classe est centrée sur un sommet d'une pyramide (3D) à quatre faces (longueur des arrêtes = 1). Toutes les données ont donc les mêmes coordonnées sur les trois premières variables. Les coordonnées diffèrent sur les sept autres dimensions : sur les variables 4 à 9, les coordonnées sont tirées au hasard dans une loi normale centrée et dont la variance est 0.5 tandis que sur la $10^{ème}$ variable, les coordonnées sont tirées dans une loi centrée réduite (ce qui engendre l'anisotropie des classes).

Les difficultés induites ici (grande dimension, égales distances entre les clusters, même ordre de grandeur des distances inter-clusters et intra-clusters, anisotropie des clusters) rendent délicate la représentation de ce jeu de données.

Trois représentations ont été obtenues par trois méthodes (figure 1). La première méthode est le Sammon's mapping [9] (qui se base sur la conservation des distances). La deuxième méthode, nommée non-metric MDS, est due à Kruskal [5, 6]; elle vise à conserver l'ordonnancement des distances entre l'ensemble des données. RankVisu constitue la troisième alternative. RankVisu a pour objectif la conservation des rangs de voisinage. On observe que les clusters ne sont pas rendus d'une façon satisfaisante ni par le Sammon's



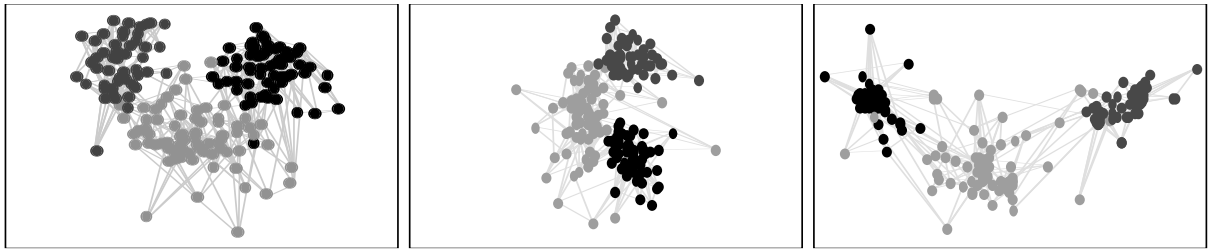
(A) (B) (C)

Figure 1: Visualisation bidimensionnelle du premier jeu de données. Trois méthodes ont été utilisées : (A) Sammon's mapping, (B) non-metric MDS, (C) RankVisu. Quatre clusters ont été générés (représentés par quatre signes différents : x, +, □ et o). Les relations de voisinage sont également exprimées : sur les représentations, chaque donnée est liée à ses cinq plus proches voisins dans l'espace d'origine (liens matérialisés par les segments gris).

mapping (où les clusters sont totalement mélangés et où les séparations entre clusters ne sont pas exprimées par des vides) ni par non-metric MDS (où les clusters sont mélangés deux à deux). En revanche, RankVisu permet de regrouper entre elles les données d'un même cluster et de séparer les clusters. Ainsi, le lecteur est capable de visualiser les clusters sur un plan.

On peut expliquer ces résultats par le fait que, même si les données sont séparables en 4 clusters (voire figure 1, graphe C), ceux-ci sont mal exprimés par les méthodes classiques. En effet, les distances inter- et intra- cluters sont du même ordre, ainsi la conservation des distances les plus courtes ne permet pas de respecter l'intégrité des clusters (figure 1, graphe A) et le rangement des distances n'apporte pas non plus de solution sur ce point (figure 1, graphe B).

Le deuxième jeu de données est un benchmark classique : les « wine data ». Il s'agit de mesures chimiques sur un ensemble de vins produit par trois viticulteurs italiens. Chaque individu est un vin, l'ensemble des vins d'un même viticulteur forme une classe. Les données ont treize dimensions (pour les treize mesures effectuées par vin). Les représentations dans des espaces à deux dimensions (figure 2) permettent toutes de séparer les classes, les données appartiennent donc clairement à trois clusters correspondants aux trois classes. Pourtant, RankVisu permet de nettement amplifier les différences. En effet, il devient alors évident que le jeu de données comprend trois clusters distincts et nous visualisons les liens qui les relient.



(A) (B) (C)

Figure 2: Visualisation bidimensionnelle des wine data (A) Sammon's mapping, (B) non-metric MDS, (C) RankVisu. Les trois niveaux de gris expriment l'appartenance aux trois classes, les segments gris relient chaque donnée à ses cinq plus proches voisins.

Conclusion

Si l'utilisation des rangs de voisinage pour la visualisation des données fait perdre la topologie selon laquelle les données sont organisées, elle permet en revanche de mettre en lumière les liens qui les relient. Ainsi les clusters apparaissent. Notez cependant que RankVisu est une méthode purement descriptive et n'a donc pas pour vocation la classification des données. Ce point peut être considéré comme une faiblesse de la méthode aussi bien que comme sa force principale. En effet, Rank Visu ne fait aucune hypothèse que ce soit sur la distribution des données, la « forme » des clusters ou le nombre de classes.

RankVisu semble être une méthode capable de mettre en valeur la présence de clusters et pourrait donc être utilisée comme analyse préliminaire afin de guider des procédures de classification. On peut même envisager son utilisation comme un prétraitement des procédures de clustering classiques, en particulier dans le cas de données pour lesquelles les distances sont soumises à caution (par exemple dans le cas de distances subjectives (données psychophysiques, …), ou dans le cas de données peuplant l'espace selon des densités variables).

Bibliographie

[1] C.C. Aggarwal, A. Hinneburg, D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space", in J. V. Bussche and V. Vianu, Eds. *Lecture Notes In Computer Science, ser. 1973*, (Berlin, Germany, Springer-Verlag, 2001), 420–434.

[2] J.P. Benzécri, "Analyse des données", Dunod Paris Bruxelles Montreal, 1973.

[3] B.V. Dasarathy, "Nearest Neighbor (NN) Norms: NN pattern Classification Techniques", (IEEE Computer Society Press, Los Alamitos, California 1990).

[4] P. Demartines,J. Hérault, "Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets", *IEEE Transactions on Neural Networks*, vol. 8, no. 1, 1997, 148-154.

[5] J.B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis", *Psychometrika*, 29, 1964, 1-27.

[6] J.B. Kruskal, "Non-metric multidimensional scaling: a numerical method", *Psychometrika*, 29, 1964, 115-129

[7] S. Lespinats, M. Verleysen, A. Giron, B. Fertil, "DD-HDS: a tool for visualization and exploration of highdimensional data", *IEEE transactions on Neural Networks*, in press.

[8] S. Lespinats, A. Giron, B. Fertil, "Compression et classification de données de grande dimension." *12èmes Rencontres de la Société Francophone de classification, SFC05*, 2005.

[9] J.W. Sammon, "A nonlinear mapping for data structure analysis", *IEEE Transactions on Computers*, vol. C-18, no. 5, 1969, 401-409.

[10]J.B. Tenenbaum, V. de Silva, J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science*, 290, 2000, 2319-2323.

[11] W.S. Torgerson, "Multidimensional scaling: 1. Theory and method", *Psychometrika*, 17, 1952, 401-419.

Clustering de nuages de points stéréoscopiques : une comparaison de différents paradigmes

Nicolas Loménie, François-Xavier Jollois

Université Paris Descartes, Laboratoire CRIP5
45 rue des Saints-Pères, 75006 Paris
(jollois, Nicolas.Lomenie)@math-info.univ-paris5.fr

Mots clés : classification automatique, reconnaissance de formes, nuages de points stéréoscopiques

A l'heure actuelle, plusieurs plateformes accessibles permettent d'obtenir rapidement des nuages de points 3D stéréoscopique ou laser représentant une scène visuelle (la caméra Triclops de la compagnie Point Grey Research). L'enjeu notamment en robotique mobile est d'être capable d'interpréter ces nuages de points de façon robuste et rapide en terme d'objets ou d'obstacles. Il s'agit d'un problème de clustering dans lequel la vérité terrain qualitative est visuellement disponible et où le niveau d'interprétation en terme de nombre de clusters est également visuellement évaluable.

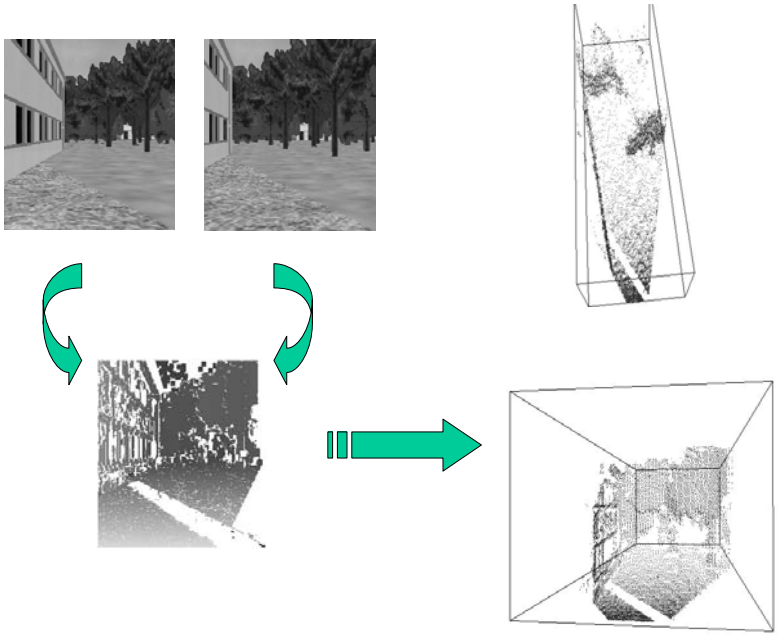


Illustration 1: Exemple d'acquisition de nuage de points 3D stéréoscopique. A partir de deux vues décalées d'une même scène, on obtient une carte de disparités puis un nuage de points 3D.

Nous comparons dans cette étude diverses techniques de clustering en précisant à chaque fois les performances et le cadre théorique sous-jacent. Les nuages de points auxquels nous nous intéressons sont stéréoscopiques (voir Ilustration 1) et les clusters à détecter sont donc caractérisés par une forte inhomogénéité en :

- densité (notamment avec l'éloignement au capteur);
- taille;
- topologie;

Dans la littérature liée à la vision par ordinateur, on trouve un algorithme appelé UFP-ONC [5] (Unsupervised Fuzzy Partition – Optimal Number of Classes) inspiré du Fuzzy C-Means de J. Bezdek [2] qui permet de traiter ces caractéristiques des clusters tout en fournissant un nombre optimal de cluster selon un critère de densité moyenne. Nous l'avons utilisé et testé avec succès sur un ensemble de scènes stéréoscopiques natuelles ou plus structurées avec des résultats exploitables pour des problèmes de navigation autonome en robotique mobile.

Parallèlement, dans la littérature liée à la fouille de données, l'utilisation des modèles de mélange dans la classification est devenue une approche classique et très puissante (voir par exemple [1] et [3]). En traitant la classification sous cette approche, nous cherchons à maximiser la vraisemblance du modèle par rapport aux données. Pour ce faire, il existe deux approches : vraisemblance et vraisemblance classifiante. Pour la première, il est courant d'utiliser l'algorithme EM [4], composé de deux étapes : Estimation et Maximisation. Celui-ci est très populaire pour l'estimation de paramètres. Ainsi, de nombreux logiciels sont basées sur cette approche, comme Mclust-EMclust, Emmix, Mixmod ou AutoClass. Pour la seconde approche, il existe une variante de EM, dénommée CEM, dans laquelle est ajoutée une étape de Classification entre les deux étapes E et M. EM travaille sur une partition floue et CEM sur une partition dure.

Dans le cas de modèles Gaussien, chaque classe est représentée par un vecteur moyenne et une matrice de variance-covariance. Cette dernière détermine l'aspect géométrique de la classe. A partir de [1] et [3], il est possible de paramétrer cette matrice pour contraindre ou non les classes sur leur forme (par exemple ellipsoïdale ou sphérique), sur leur orientation ou sur leur volume. Par ailleurs, Hathaway [6] a montré que l'algorithme EM était équivalent à fuzzy c-means (FCM) pour le cas d'un modèle de mélange gaussien contraint.

Nous pouvons considérer le problème du nombre de classes avec une perspective bayésienne. Ainsi, nous pouvons utiliser les critères AIC, BIC, CS et ICL. Les trois premiers sont basés sur une approximation de la vraisemblance intégrée et le dernier sur une approximation de la vraisemblance classifiante intégrée. Ceux-ci pénalisent l'adéquation du modèle aux données avec la complexité de ce modèle. La pénalisation diffère entre ces critères.

Dans cette édude, nous comparons donc les résultats obtenus en terme de vérité terrain, de temps de calcul et de robustesse dans un cadre spécifique de vision par ordinateur avec l'état de l'art des algorithmes et outils disponibles actuellement dans la communauté des fouilleurs de données.

[1] Banfield, J. D. and Raftery, A. E. (1993), Model-based Gaussian and non-Gaussian Clustering, *Biometrics*, 49, 803–821,
[2] J.C. Bezdec, Fuzzy mathematics in pattern classification, Cornell University, Ithaca, NY
[3] Celeux, G. and Govaert, G. (1995), Gaussian Parcimonious ClusteringMethods, *Patt. Rec.*, 28, 781–793, 1995.
[4] Dempster, A. and Laird, N. and Rubin, D. (1977), Mixture Densities, Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statitical Society*, 39, 1, 1–38, 1977.
[5] I. Gath and A.B. Geva, Unsupervised optimal fuzzy clustering , *Pattern Analysis and Machine Intelligence*, 1989, vol. 11(7), pp. 773-781, july 1989
[6] Hathaway, J., Another interpretation of the EM algorithm for mixture distribution, *Journal of Statisticals and Probability Letters*, 4:155-176, 1986

Classification et analyse textuelle : l'approche topologique

S. Mellet[1], X. Luong[1], D. Longrée[1,2] et J.P. Barthélemy[3]

1. Laboratoire BCL, Université Nice Sophia-Antipolis, CNRS ; MSH de Nice, 98 bd E. Herriot, F-06200 NICE
2. Laboratoire LASLA, Université de Liège, Quai Roosevelt 1B, B-4000 LIEGE
3. Laboratoire TAMCIC, ENST Bretagne, CNRS, BP 832, F-29285 Brest Cédex
(mellet, luong, longree@unice.fr ; jp.barthelemy@enst-bretagne.fr)

Mots clés : analyse textuelle, classification automatique, topologie textuelle, motifs.

0. Introduction

Contrairement à ce que laisseraient supposer les méthodes traditionnelles de classification automatique des textes qui prennent appui sur de purs dénombrements d'occurrences, un texte n'est pas un sac dans lequel seraient rassemblées en vrac ses unités constitutives. Un texte est à tout le moins une chaîne linéaire, donc un espace ordonné. Or qui dit espace dit aussi lieux : lieux privilégiés de concentration de l'information, lieux de martèlement et de redondance, lieux de pause informative, lieux de transition et zones de rupture, entre deux développements thématiques différents par exemple. Autant de « topoï » intrinsèques, inhérents à tout déploiement textuel. Tout cela est bien connu de toutes les disciplines qui ont pour objet d'étude le texte, mais on avait cru pouvoir l'occulter dans les traitements statistiques, ou du moins ne récupérer qu'a posteriori cette structuration linéaire.

La reconnaissance et la prise en compte de l'existence dans les textes de ces lieux différenciés nous ont donc conduits à développer d'autres méthodes de classification en postulant qu'un texte est d'abord un ensemble (E) d'unités linguistiques qui ne sont pas indépendantes les unes des autres, qui est muni d'une structure ou, plus exactement, de plusieurs structures imbriquées dont l'union constitue cet ensemble. Ces structures dessinent des sous-ensembles de (E), délimités par des zones frontières dont les propriétés sont souvent très intéressantes à étudier. Il devient alors tentant de comparer le texte à un espace topologique dont les formes pourraient être analysées quantitativement et qualitativement, à des fins classificatoires notamment.

1. Rappel sur les méthodes traditionnelles de classification des textes

Les textes sont ramenés à un ensemble d'unités (par ex. l'ensemble des formes graphiques qui les constituent, ou l'ensemble des lemmes, c'est-à-dire des entrées de dictionnaire auxquelles ils font appel, ou encore l'ensemble des catégories grammaticales qui y sont représentées, etc.) et ces unités d'analyse sont considérées globalement et en vrac ; elles donnent lieu à des dénombrements qui permettent de constituer de classiques tableaux de contingence à partir desquels des AFC, des analyses arborées ou des analyses en composantes principales permettent de représenter les similarités des profils textuels. Il est à noter cependant que les matrices issues du dénombrement des catégories grammaticales présentent des spécificités par rapport aux matrices lexicales qui nécessitent un aménagement des analyses traditionnelles (obligation de travailler sur les fréquences et non pas en termes de présence / absence ; prise en compte de quelques colonnes creuses et néanmoins significatives ; cf. X. Luong & S. Mellet [10]). Ces méthodes, traditionnelles en statistique linguistique, donnent globalement d'assez bons résultats, mais ne permettent pas des classifications fines prenant en compte la structuration interne des textes. Or cette structuration est parfois déterminante pour faire émerger les classifications génériques, *i.e.*

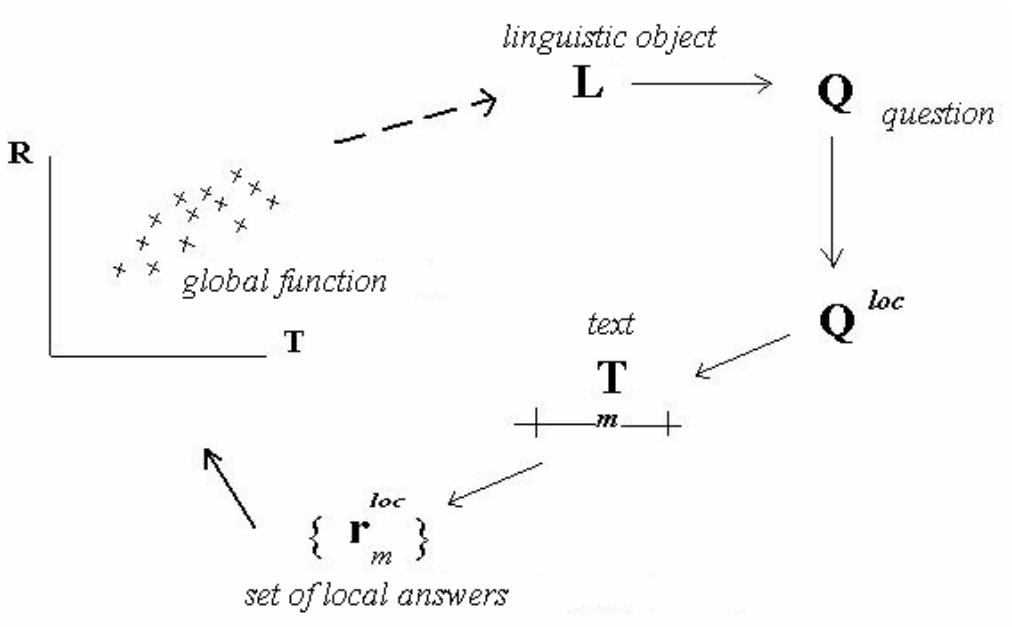
liées au genre et au sous-genre dont relève chaque texte (cf. Luong, Longrée & Mellet [6] ; Longrée & Mellet [7]).

2. La classification textuelle à partir des études de voisinage

Nous considérons donc qu'un texte est une structure linéaire constituée d'un ensemble d'événements linguistiques (occurrence d'un mot, d'un syntagme, d'une catégorie grammaticale, etc.) qui, chacun à leur tour, peuvent être considérés comme des points remarquables de la chaîne textuelle. Mais ce faisant, l'analyste ne saurait détacher l'unité observée de son contexte immédiat, c'est-à-dire de la portion textuelle jugée pertinente pour l'analyse et qui comprend un certain nombre d'autres mots (ou, plus généralement, d'autres événements linguistiques) situés avant et après lui. La pertinence de la taille du contexte varie selon les faits étudiés, la taille globale du texte, l'objectif de la recherche, etc. L'empan contextuel doit donc être défini par le linguiste en fonction de ces différents paramètres ; si l'ensemble des paramètres n'est pas maîtrisable, le choix d'une taille arbitraire ajustée par essais/erreurs peut apparaître légitime et donner d'excellents résultats.

Ce nombre [arbitrairement] fixé d'éléments x_i de (E) entourant le point x (*i.e.* l'occurrence étudiée) est considéré comme un ***voisinage*** de x. Et puisque l'on peut, selon les besoins de l'étude, faire varier ce nombre et donc la taille des contextes ainsi définis, on considère que chaque point x se trouve muni d'une ***famille de voisinages***.

On applique alors un modèle d'interrogation sur ces familles de voisinage qui peut être résumé au moyen du schéma suivant :



Ce modèle permet à la fois de caractériser localement chaque voisinage et de fournir une fonction numérique globale pour l'espace étudié.

Soit en effet L l'objet linguistique à étudier dans un texte T pour caractériser celui-ci par rapport à d'autres textes T_1, T_2, T_n. Cet objet est soumis à une question Q (par exemple, quels sont les cooccurrents privilégiés de L ?) ; la question globale Q se décline en une ou plusieurs questions locales (Q^{loc}) portant sur les voisinages de L à un point m du texte T (par exemple, pour chaque occurrence de L dans le texte, quelles sont les catégories grammaticales contenues dans un voisinage de 10 mots avant et après L ? Combien de substantifs ? Combien d'adjectifs ? Combien de verbes ? etc.). Les procédures d'exploration automatique du texte fournissent une série de réponses sous forme numérique $\{r_m^{loc}\}$. Il est alors possible de

produire un graphe dessinant un nuage de points dont chacun correspond à une réponse locale et dont l'ensemble constitue la fonction globale associée à L, laquelle peut ensuite être sollicitée pour déterminer certaines propriétés topologiques du texte sous étude et le comparer à d'autres textes.

L'étude des voisinages est donc susceptible de soutenir en partie une classification des textes fondée sur une approche topologique de l'analyse textuelle. Une version longue de cet exposé pourra en fournir un exemple concret emprunté aux travaux de Luong, Longrée et Mellet, dans lesquels l'objet linguistique étudié est le temps verbal privilégié de la narration, à savoir le parfait, équivalent du passé simple français : selon que le voisinage de chaque occurrence de parfait dans un texte compte majoritairement d'autres formes de parfaits ou au contraire une variété de temps incluant aussi des imparfaits, des plus-que-parfaits et des présents de narration, le profil de chaque texte diffère et fournit une classification intéressante. La distribution des voisinages selon les différentes parties du texte est aussi prise en compte.

On note cependant que, dans cet exemple, on appréhende les textes à travers un seul paramètre. Or il semblerait judicieux de pouvoir associer plusieurs paramètres perçus comme autant de dimensions du texte. Tel est l'objectif de la méthode que nous développons actuellement : la classification à partir des motifs du texte.

3. La classification textuelle à partir des motifs récurrents

On appelle ici « motif » l'association récurrente de n éléments de l'ensemble (E) muni de sa structure linéaire qui donne une pertinence aux relations de successivité et de contiguïté. Ainsi, si l'ensemble (E) est composé de x occurrences des éléments A, B, C, D, E, F, un premier motif pourra être la récurrence du groupe linéairement ordonné ABD, un autre motif pourra être la récurrence du groupe AA.

Apparemment simple, cette définition soulève pourtant un certain nombre de difficultés dès que le linguiste cherche à la concrétiser : nous n'aborderons cependant pas ici ces problèmes et nous nous contenterons d'illustrer notre propos par un exemple qui met en jeu les divers types de propositions – principales et subordonnées – de divers textes narratifs latins, en admettant que la phrase est une unité pertinente pour la détection des motifs liés à ce type d'unités linguistiques. L'objectif est d'obtenir une classification des textes du corpus en fonction d'une part de la fréquence des motifs dominants, d'autre part de leur combinaison mutuelle et de leur distribution au fil de chaque texte.

Dans ce second exemple, les propositions qui nous intéressent sont d'une part la proposition principale, d'autre part certaines subordonnées qui contribuent à poser un cadre circonstanciel à l'action, tels les ablatifs absolus (*i.e.* une forme de proposition participiale) et les subordonnées en *cum* + subjonctif (*i.e*. l'équivalent approximatif des propositions en *alors que*, *comme*). L'organisation de toutes ces propositions structure en effet la narration et les stylisticiens ont depuis longtemps montré qu'elles contribuaient à caractériser le style d'un auteur : certains en effet les utilisent avec parcimonie, d'autres ne reculent pas devant leur accumulation. Certains préfèrent placer les subordonnées en début de phrase, pour poser d'abord le cadre de l'action principale, d'autres au contraire pratiquent volontiers la « relance syntaxique » en fin de phrase, rajoutant après coup des éléments informatifs secondaires. Sont ainsi traditionnellement définies la phrase « à structure narrative type » avec complément circonstanciel initial et la phrase « à rallonge » ; on observe bien sûr des usages mixtes, combinant l'une et l'autre des structures. Les différents motifs, leur fréquence respective et leur combinaison caractérisent donc le style d'un auteur ; leur distribution pourrait aussi caractériser les différentes parties d'une œuvre (les parties introductives ou de commentaires entre les passages narratifs offrant un cadre plus accueillant aux accumulations de subordonnées, par exemple).

Chaque texte du corpus se trouve donc caractérisé par un ensemble de descripteurs qui rendent compte, sous forme numérique, de l'usage (en fréquence et en distribution locale et globale) des motifs au sein de chaque texte du corpus. L'ensemble de ces descripteurs donne le profil de chaque texte et ces profils sont ensuite intégrés à une matrice rectangulaire dont les textes fournissent l'intitulé des lignes et dont les descripteurs de profil fournissent les colonnes. A partir de là peuvent être appliquées les méthodes classiques de calcul de distance ; pour notre part nous adoptons le calcul de distance et la représentation arborée proposée par Luong & Barthélémy [2], qui a l'avantage de constituer des classes tout en imposant simultanément une contrainte locale de proximité.

Il est à noter que cette analyse est, à toutes ses étapes, multidimensionnelle. Le profil affecté à chaque texte et intégré à la matrice de calcul des distances est un profil complexe qui répercute la diversité des paramètres nécessaires pour rendre compte d'un texte dans sa richesse et sa singularité. La représentation graphique qui en est issue ne correspond pas, bien sûr, à la topologie interne des textes, mais à une topologie externe qui est l'image de la structuration – ou, plutôt, d'une structuration possible – du corpus.

Références bibliographiques

[1] Barthélemy J.-P. & Guénoche A., *Les arbres et les représentations des proximités*. Paris : Masson, 1988.

[2] Barthélemy J.P. & Luong X., « Sur la topologie d'un arbre phylogénétique : aspects théoriques, algorithmiques et applications à l'analyse de données textuelles », *Mathématiques et Sciences Humaines* 100, 1987, 57-80.

[3] Barthélemy J.-P. & Luong X., « Représenter les données textuelles par des arbres », in *JADT 1998*, Actes des 4èmes Journées Internationales d'analyse de données textuelles, Univ. de Nice : UMR 6039, 1987, 49-70.

[4] Lebart L., « Validité des visualisations de données textuelles », in G. Purnelle, C. Fairon & A. Dister (éds), *JADT 2004*, 7èmes Journées internationales d'Analyse statistique des Données Textuelles. UCL : Presses universitaires de Louvain, vol. 2, 2004, 708-715.

[5] Longrée D. & Luong X., « Temps verbaux et linéarité du texte : recherches sur les distances dans un corpus de textes latins lemmatisés », *Corpus* 2, 2003, 119-140.

[6] Longrée D., Luong X. & Mellet S., « Temps verbaux, axe syntagmatique, topologie textuelle : analyse d'un corpus lemmatisé », in G. Purnelle, C. Fairon & A. Dister (éds), *JADT 2004*, 7èmes Journées internationales d'Analyse statistique des Données Textuelles. UCL : Presses universitaires de Louvain, vol. 2, 2004, 743-752.

[7] Longrée D.& Mellet S., « Temps verbaux et prose historique latine : à la recherche de nouvelles méthodes d'analyse statistique », in Actes du 13ème Colloque international de Linguistique latine (ICLL 13, Bruxelles 2005), sous presse.

[8] Longrée D., Mellet S. & Luong X., « Distance intertextuelle et classement des textes d'après leur structure : méthodes de découpage et analyses arborées », in J.M. Viprey (éd.), *JADT 06,* 8èmes Journées internationales d'Analyse statistique des Données Textuelles. Besançon : Presses universitaires de Franche-Comté, vol. 2, 2006, 643-654.

[9] Luong X., Juillard M., Mellet S. & Longrée D., « The concept of text topology. Some applications to language corpora », *Literary and Linguistic Computing*, (à paraître).

[10] Luong X. & Mellet S., « Mesures de distance grammaticale entre les textes », *Corpus* 2, 2003, 141-166.

[11] Piérard S. & Bestgen Y., « A la pêche aux marqueurs linguistiques de la structure des discours », in J.M. Viprey (éd.), *JADT 06*, 8èmes Journées internationales d'Analyse statistique des Données Textuelles. Besançon : Presses universitaires de Franche-Comté, vol. 2, 2006, 749-758.

Sélection de modèles prévisionnels par analyse de données symboliques

O. Merroun[1,3], E. Diday[1], A. Dessertaine[2], P. Rigaux[3], E. Eliezer[4]

1. CEREMADE Place du Maréchal De Lattre De Tassigny
75775 PARIS CEDEX 16 - FRANCE
2. EDF 1, avenue du général de Gaulle 92140 Clamart- FRANCE
3. LAMSADE Place du Maréchal De Lattre De Tassigny
75775 PARIS CEDEX 16 – FRANCE
4. SYROKKO 6, rue Ambroise Jacquin 95190 Fontenay-en-Parisis- FRANCE

omar.merroun@gmail.com, diday@ceremade.dauphine.fr, philippe.rigaux@dauphine.fr, alain.dessertaine@edf.fr, eliezer@syrokko.com

Mots clés : analyse des données symboliques, prévision.

Résumé

La consommation électrique dépend souvent fortement de la température constatée. Les modèles de prévision de consommation électrique, particulièrement ceux utilisés pour prévoir les consommations globales horaires ou demi-horaires sur un horizon de 9 jours, sont de types additifs et non linéaires. Ils prennent en compte de manière non linéaire l'influence de la température, ainsi que des phénomènes saisonniers et calendaires, auxquels s'ajoute une modélisation de type ARIMA des erreurs. L'apprentissage des modèles se fait à températures constatées. Les prévisions de température quant à elles s'appuient sur les estimations fournies par Météo France, et plus particulièrement sur la moyenne de 51 différents scénarios de températures.

On se propose de traiter la problématique suivante : pouvons-nous déterminer parmi ces 51 scénarios celui où la moyenne de ceux qui donneront le plus précisément possible la température, dans le but de minimiser dans nos prévisions de consommation les risques liés aux aléas de prévisions de températures fournies par Météo France ?

Dans un premier temps, nous nous limiterons à simplement détecter, sur une période donnée, le modèle qui prévoit avec le plus de précision possible la température.

On dispose de 20 jours du mois de juillet 2005, de 51 scénarios d'évolution possible pour les 9 jours suivants. On connaît également les températures observées durant la même période. Pour trouver les scénarios les plus intéressants on va représenter les données dans un formalisme symbolique, et se servir des outils fournis par l'Analyse des Données Symboliques.

À partir du tableau de données des prévisions des températures, on a calculé la « qualité » de la prévision par la formule suivante : qualité = température prédite/ température constatée. Cet indicateur nous permet d'estimer la précision des scénarios pour chaque jour du mois de juillet.

Chaque jour du mois de juillet (jour de départ) correspond à 9 prévisions pour les 9 prochains jours (« horizons » du jour de départ), pour un scénario. Donc chaque jour de départ est décrit par 9 valeurs de qualité de prévision pour chaque scénario. (Tableau ci-dessous) :

Date de départ « j »	Horizon « h »	Date cible « p »	scénario 0	scénario 1	scénario 2	…	scénario 50
…	…	…	…	…	…	…	…
10/04/2005	1	11/04/2005	16,5	15,6	16,1	…	15,9
	2	12/04/2005	16,4	15,9	16,0	…	15,8
	…	…	…	…	…	…	…
	9	19/04/2005	16,3	15,7	16,5	…	17
11/04/2005	1	12/04/2005	16,7	15,8	16,7	…	16,9
	…	…	…	…	…	…	…
	9	20/04/2005					

Tableau 1, tableau de prévision de température des 51 scénarios

Par ailleurs, on dispose de l'historique des températures constatées, pour la période en question. Le Tableau 2 concerne la description des températures constatées à un jour *j*.

date	Température observée
…	…
10/04/2007	16,1
11/04/2007	15,9
…	…

Tableau 2, les températures constatées

Pour pouvoir comparer les performances des scénarios, on calcule le rapport suivant, qui exprime la qualité de la prévision :

$$qualité = \frac{température\quad prédite}{température\quad constatée}$$

Donc si la « qualité » vaut 1, cela veut dire que la prévision est bonne. On obtient un nouveau tableau de données (Tableau 3) en substituant, dans le Tableau 1, les valeurs de prévision de chaque scénario par le rapport exprimant la qualité de cette prévision.

Jour de départ X horizon	Date de départ « j »	Date cible « p »	scénario0	scénario 1	scénario 2	…	scénario 50
	…	…	…	…	…	…	…
10/04/2005x1	10/04/2005	11/04/2005	0.98	0.95	0.96	…	0.98
10/04/2005x2	10/04/2005	12/04/2005	0.97	0.96	0.98	…	0.97
…	…	…	…	…	…	…	…
10/04/2005x9	10/04/2005	19/04/2005	0.95	0.94	0.89	…	0.90
11/04/2005x1	11/04/2005	12/04/2005	0.99	0.98	0.96	…	0.97
11/04/2005x2	11/04/2005	13/04/2005	0.99	0.98	0.96	…	0.97
…	…	…	…	…	…	…	…
20/04/2005x9	20/04/2005	29/04/2005	0.97	0.97	0.97	…	0.97

Tableau 3, tableau de la qualité des prévisions de température des 51 scénarios

Pour pouvoir traiter et analyser ces données, on a transformé ce tableau de données classique, en un autre, *symbolique*, grâce au module DB2SO du logiciel public SODAS[1]. Cette transformation a engendré le Tableau 3.

Individu	Température constatée	scénario 0	scénario 1	scénario 2	…
11/07/2005	[15,10:19,20]	[0.87 :0.99]	[0.90 :1.11]	[0.91 :1.22]	…
12/07/2005	[15:19,20]	[0.89 :1.09]	[0.85 :1.04]	[0.89 :0.12]	…
13/07/2005	[15:18,30]	[0.97 :0.99]	[0.89 :1.29]	[0.91 :1.21]	…
14/07/2005	[15:18]	[0.88 :1.02]	[0.90 :1.09]	[0.92 :1.10]	…
…					

Tableau 4, matrice symbolique tableau de la qualité des prévisions de température des 51 scénarios

Chaque intervalle est obtenu en prenant le *min* et le *max* des valeurs de qualité des 9 jours pour un scénario considéré. Par exemple, l'intervalle correspondant au jour 13/07/2005 est [0.97 :0.99]. Ce qui signifie que la qualité minimum pour les 9 jours qui suivent est 0,97, de même la qualité maximum est 0,99 pour ces 9 jours.

Cette méthode nous permet de regrouper les informations qui décrivent 9 jours suivant un même jour de départ. À partir de ce tableau de données, on veut comparer les scénarios de prévision selon deux critères : la *variance nulle*, et la *valeur modale*.

Un scénario est d'une bonne qualité si la valeur modale correspond à la modalité (intervalle) qui contient la valeur 1 (la qualité de la prédiction est proche de la température observée). Par ailleurs, plus l'écart type est petit, plus la courbe se concentre autour de la classe modale, et donc plus le scénario est pertinent.

À partir de la matrice symbolique, on obtient successivement deux sorties grâce à un module amélioré de SODAS développé dans le cadre du projet SEVEN piloté par EDF Clamart (projet RNTL de l'appel 2005 http://www.rntl.org/projet/resume2005/seven.htm). La première, consiste à construire les histogrammes des intervalles de précision des prédictions pour chaque scénario. Chaque histogramme associé à chaque scénario est bâti sur un découpage en dix classes égales entre la plus petite et la plus grande des valeurs de tous les intervalles du tableau. Les fréquences des classes correspondent au cumul des fractions des intervalles qui les coupent, pour

[1] http://www.ceremade.dauphine.fr/%7Etouati/sodas-pagegarde.htm

chaque jour de départ. Tous les histogrammes sont décrits par les mêmes classes. Notamment, la classe 5 du scénario T0 correspond à une fréquence de 0.53571.

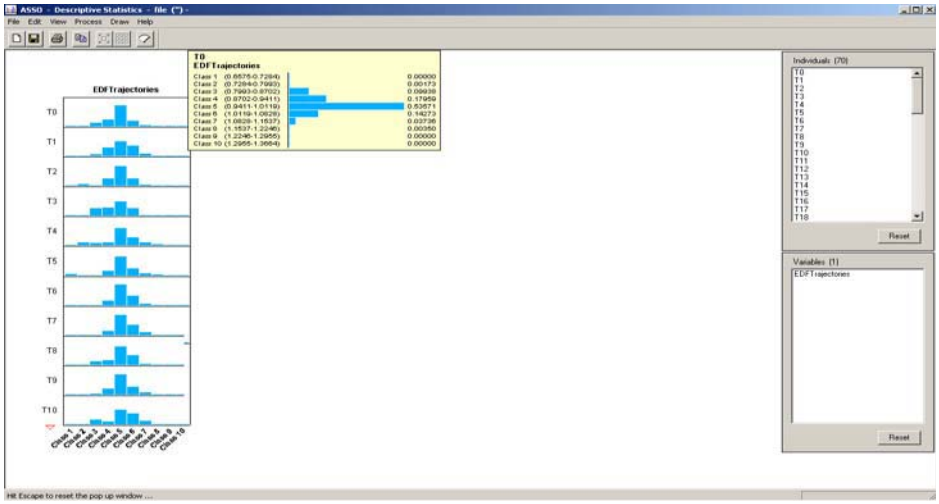


Figure 1 : environnent affiché par l'option « matrice des histogrammes »

La deuxième sortie affiche pour chaque classe la valeur minimale, moyenne et maximale des histogrammes de la Figure 1. La classe qui nous intéresse est la 5ième, car elle contient la valeur 1 (qui correspond à une température prédite égale à la température constatée). On constate aussi que les scénarios prédisent globalement avec plus de précision la bonne température puisque le mode est obtenu pour la classe 5 qui contient la valeur 1.

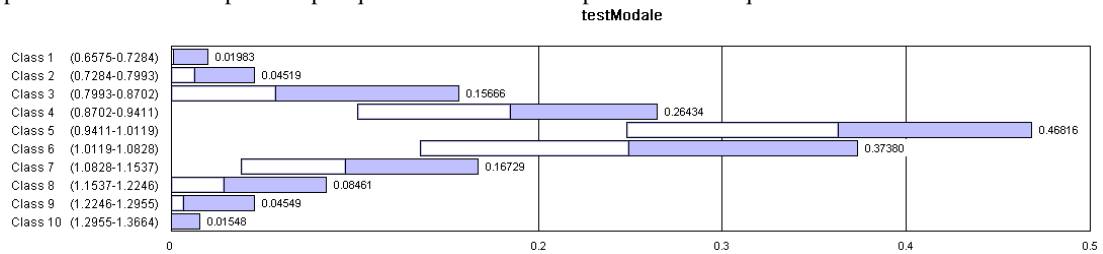


Figure 2 : l'histogramme minimum/moyenne/maximum représentant les 51 scénarios de prédiction

Afin d'ordonner les scénarios selon leur qualité, nous les trions selon l'ordre croissant de la valeur de la modalité 5 (5ième classe), qui correspond au mode des histogrammes de la Figure 2.

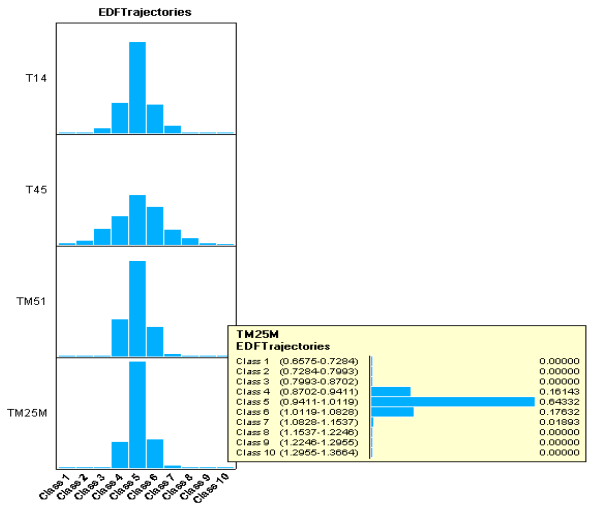


Figure 3 : comparaison d'un scénario T45 (le plus mauvais scénario), avec le meilleur scénario T14 parmi les 51, suivi du modèle moyen TM51, et du modèle TM25M. Ce dernier est le meilleur et le plus détaillé.

Dans la Figure 3, on retient les modèles les plus intéressants. On a d'abord le modèle TM25M, définit par la moyenne des 25 meilleurs scénarios, qui est le meilleur modèle. Il a la plus grande valeur modale (0.64332). On trouve ensuite le modèle TM51, correspondant à la moyenne de tous les modèles. Sa valeur modale est 0.5798. Il est suivi par le meilleur scénario, T14 (0.5558). Enfin le scénario le plus mauvais est le T45 (0.3051).

Finalement, on a composé d'autres modèles de prédiction basés sur les 51 scénarios, dans l'objectif de trouver un meilleur modèle. Partant de la liste triée des 51 scénarios, on calcule des moyennes de *n* meilleurs scénarios et *n* mauvais scénarios, où *n* varie de 5 à 51 avec des sauts de 5.

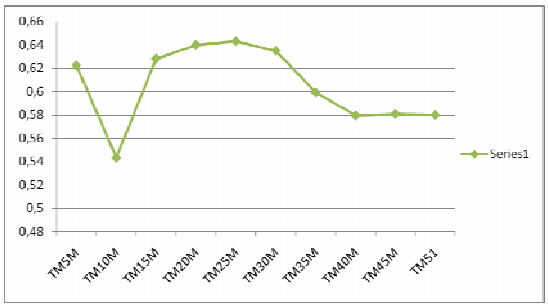


Figure 4 : évolution des *n* meilleurs scénarios

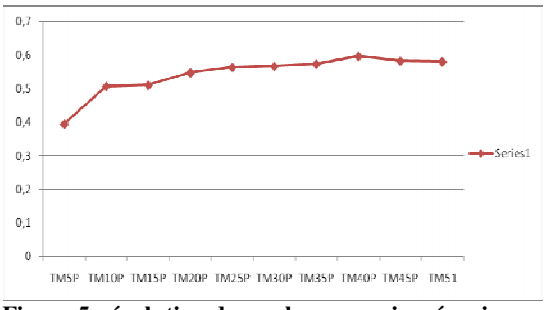


Figure 5 : évolution des *n* plus mauvais scénarios

Dans la Figure 4, on s'aperçoit que la prédiction s'améliore (à partir de la moyenne des dix meilleurs) à mesure que l'on ajoute des scénarios. La courbe s'améliore ainsi jusqu'à un pic correspondant au modèle des 25 meilleurs scénarios. Puis la courbe se met à décroître à mesure que la qualité des scénarios ajoutés se détériore. D'après la Figure 5, on constate que le modèle le plus mauvais est le TM5P, la moyenne des 5 pires scénarios. Au fur et à mesure qu'on ajoute d'autres scénarios, qui sont moins mauvais, on améliore sensiblement la qualité de prédiction.

Conclusion

Cette démarche, étant effectuée sur une période donnée, permet de trouver le meilleur scénario, ou modèle de prédiction, en se basant sur la valeur modale et sa position (par rapport au 1), et la variance de la distribution fréquentielle. On constate qu'il n'existe pas forcement un meilleur scénario, mais plutôt des combinaisons, notamment la moyenne, permettant d'améliorer la qualité de prédiction.
La prochaine étape de notre travail sera d'essayer de prévoir à une date déterminée, en connaissance du contexte (saisonnalité, performance passée des différents scénarios, comportement même de la température constatée …), la meilleure combinaison de scénarios, à utiliser pour les prochains 9 jours.

Références

A. Dessertaine, (2007) "Data Stream and Load Forecasting : some ideas for a research project at Electricité De France". Workshop on Data-Stream Analysis – Caserta (Italie)

A. Bruhns., G. Deurveilher, J.S. Roy, (2005) "A non linear regression model for mid-term load forecasting and improvements in seasonality", Power Systems Computation Conference 2005

C. Maté, J. Arroyo, A. Muñoz et A. Sarabia (2006) : "Smoothing methods for histogram-valued time series" - 26th International Symposium on Forecasting. Santander (España). 11-14 Juin 2006

E. Diday (2005) "Categorization in Symbolic Data Analysis". In handbook of categorization in cognitive science. Edited by H. Cohen and C. Lefebvre. Elsevier editor.
http://books.elsevier.com/elsevier/?isbn=0080446124

H.-H. Bock, E. Diday (2000) "Analysis of Symbolic Data for extracting statistical information from complex data". Springer Verlag, Heidelberg, ISBN 3-540-66619-2.

L. Billard, E. Diday (2006) "Symbolic Data Analysis: conceptual statistics and data Mining". Wiley. ISBN 0-470-09016-2

Classification de parcours de vie à l'aide de l'optimal matching*

Nicolas S. Müller, Matthias Studer, Gilbert Ritschard

Département d'économétrie, Université de Genève
nicolas.muller@metri.unige.ch

Mots clés : application, classification automatique, séquences.

Ce travail analyse les parcours de vie familiale en les considérant comme des séquences. Le but est de parvenir à observer le caractère temporel des parcours de vie en prenant en compte la durée entre chaque événement constitutif de ce parcours, mais aussi l'ordre dans lequel ils surviennent. Nous proposons d'appliquer aux données de l'enquête biographique rétrospective du Panel suisse de ménages une méthode d'analyse des séquences dans le but d'obtenir une typologie des parcours de vie du 20ème siècle. Celle-ci nous permettra ensuite de mieux approcher les changements qui ont pu intervenir dans leur structure.

1 Méthode

Afin de représenter les parcours de vie familiale sous forme de séquences, quatre événements ont été retenus. Il s'agit de l'âge au départ de chez les parents, l'âge au premier mariage, l'âge au premier enfant et l'âge au premier divorce. Ces événements sont considérés comme les étapes qui jalonnent la vie des individus. Ils sont comme des « bornes » qui délimitent les différents états qu'un individu traverse au cours de sa vie. La vie familiale d'un individu est donc représentée sous la forme d'une « séquence », une suite d'années de sa vie avec comme information l'état dans lequel il se trouve chaque année. A partir des quatre événements retenus, huit états distincts ont été définis. Ils sont représentés dans le tableau 1.

Les individus sont observés de l'âge de quinze ans à celui de trente ans. Ceux âgés de moins de trente ans au moment de l'enquête ont été éliminés afin de ne conserver que des séquences complètes et sans données manquantes. Le nombre total d'observations est de 4318. Dans le but de limiter le nombre d'états, il a été décidé de considérer tout divorce comme un état unique, tout en ayant conscience que le nombre de divorces avant l'âge de trente ans est très faible.

La méthode d'analyse de séquences que nous utilisons dans ce travail est celle dite d' « optimal matching ». L'algorithme retenu, connu aussi sous le nom d'alignement de séquences, a été développé à l'origine pour l'analyse rapide des protéines et des séquences d'ADN. Ce type de méthode a été conçu pour permettre la comparaison rapide de nombreuses séquences afin de trouver des correspondances parmi celles-ci. Les premiers algorithmes d'optimal matching sont apparus au début des années 70 et leur première utilisation dans les sciences sociales remonte à l'article d'Abbott et Forrest sur leur application à des données historiques [1]. On doit à Abbott de nombreux articles méthodologiques sur l'utilisation de ces méthodes dans les sciences sociales, et notamment en sociologie [2], [3], [4].

*Etude soutenue par le Fonds national suisse de la recherche (FNS) FN-100012-113998, et réalisée avec les données collectées dans le cadre du projet « Vivre en Suisse 1999-2020 », piloté par le Panel suisse de ménages et supporté par le FNS, l'Office fédéral de la statistique et l'Université de Neuchâtel.

TAB. 1 – Liste des états

	départ	mariage	enfant	divorce
0	non	non	non	non
1	oui	non	non	non
2	non	oui	oui/non	non
3	oui	oui	non	non
4	non	non	oui	non
5	oui	non	oui	non
6	oui	oui	oui	non
7	oui/non	oui/non	oui/non	oui

Concrètement, cette méthode évalue la « distance » entre une séquence A et une séquence B en calculant le nombre minimum d'opérations nécessaires pour passer de l'une à l'autre. Les opérations disponibles sont de deux types, soit l'insertion ou la suppression d'un état, soit la substitution d'un état par un autre. Un coût est attribué à ces deux types d'opération selon le type de données. Dans ce travail, les opérations de substitution ont été favorisées par rapport aux opérations d'insertion/suppression dans le but d'éviter la déformation du temps (allongement ou rétrécissement) qu'elles provoqueraient. Les coûts de substitution entre états sont définis par une matrice des coûts calculée en fonction des taux de transition observés dans les données. Ainsi, plus un passage d'un état à un autre est observé fréquemment, plus son coût est bas.

2 Résultats

L'application de l'optimal matching a permis le calcul d'une « distance » entre chaque individu, en fonction du nombre de transformations nécessaires pour passer d'une séquence à une autre. Le résultat se présente sous la forme d'une matrice symétrique de distances qui est ensuite utilisée dans une analyse de classification hiérarchique ascendante. La méthode de regroupement des cas est celle de Ward. Nous avons décidé de retenir une solution de classification à cinq classes. Une visualisation des résultats sous la forme d'histogrammes avec pour chaque année de vie la contribution de chaque groupe au nombre total d'individus (barres empilées à 100%) nous permet de distinguer les cinq groupes selon plusieurs caractéristiques. Les graphiques pour les groupes 1 et 2 sont reproduits plus loin (voir figure 1 et 2).

Le premier groupe (n = 952 ; 22% du total) contient une proportion remarquable d'individus ayant eu des enfants sans mariage et d'individus ayant rencontré un divorce. Le deuxième groupe (n = 1051 ; 24,3%) se caractérise par une période entre le départ de chez les parents et le premier mariage qui est courte. Les mariages sont relativement précoces : 50% des individus qui composent ce groupe sont mariés à l'âge de 22 ans, et 100% à 27 ans (l'âge médian au premier mariage de l'échantillon = 26 ans ; 19,2% mariés à 22 ans). Le troisième groupe (n = 1228 ; 28,4%), contient des individus au départ plutôt tardif ; les premiers départs ne commencent qu'à l'âge de 23 ans (l'âge médian au premier départ dans l'échantillon est de 22 ans). Le quatrième groupe (n = 872 ; 20,2%) se caractérise par des départs précoces suivis d'une période relativement longue avant les premiers mariages. A l'âge de 30 ans, seulement 30% des individus sont mariés, dont moins de la moitié avec des enfants. Le dernier groupe, (n = 215 ; 5%) est constitué d'individus qui ne quittent

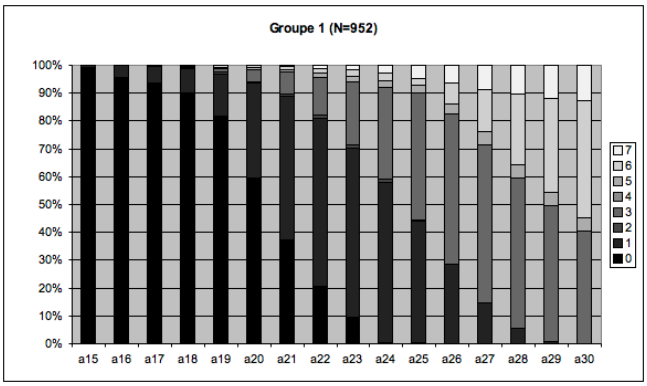


FIG. 1 – Groupe 1

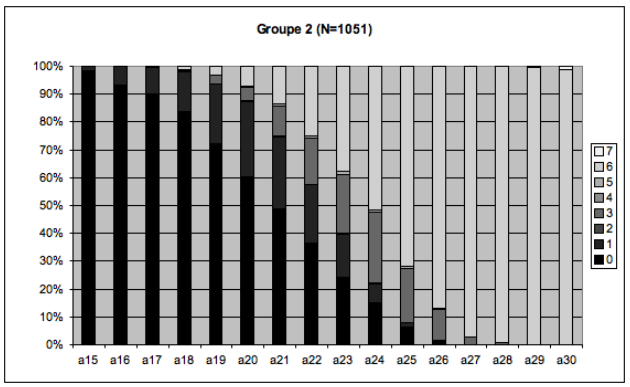


FIG. 2 – Groupe 2

pas le foyer parental. Il est intéressant de noter que ce groupe est réuni avec le groupe trois (départs tardifs) dans la solution de classification à quatre groupes. Des tests du χ^2 nous permettent de rejeter d'emblée l'hypothèse d'indépendance entre l'appartenance à l'un des groupes et les variables cohorte de naissance et genre. Les cohortes de naissance ont été définies selon la discrétisation optimale opérée par la procédure CHAID d'induction d'arbre dans le cas où la variable prédite est l'appartenance à un groupe et la variable prédictrice l'année de naissance. En fait on obtient ainsi 5 cohortes dont on a réuni la première et la dernière avec leur cohorte adjacente respective pour faciliter l'interprétation et éviter d'avoir des cohortes avec trop peu d'individus. Les trois cohortes résultantes sont définies selon les dates de naissance suivantes : de 1909 à 1947, de 1948 à 1956 et de 1957 à 1972.

Afin de capter de manière plus précise l'effet de cohorte et l'effet de genre sur l'appartenance à l'un des groupes découverts, une régression logistique a été faite pour chacun des cinq groupes. Plusieurs variables ont été introduites dans le modèle, comme la catégorie socio-professionnelle des parents, le niveau d'éducation des parents ou encore la langue dans laquelle a été rempli le questionnaire (et qui permet d'obtenir une approximation de la région linguistique dans laquelle habite l'individu). Les variables retenues pour chaque groupe sont la cohorte de naissance ainsi que le sexe, excepté pour le groupe 4. Dans celui-ci, le genre ne semble pas être un facteur favorisant l'appartenance ou non. Les coefficients des régressions logistiques sont présentés dans le tableau 2.

On observe qu'un individu a environ 2,5 fois plus de chances d'être dans le groupe 5 ou 2 fois plus de chances d'être dans le groupe 2 s'il est une femme plutôt qu'un homme. Un homme a quant à lui 2,3 fois plus de chances qu'une femme d'être dans le groupe 3

TAB. 2 – Régressions logistiques

	groupe 1	groupe 2	groupe 3	groupe 4	groupe 5
cohorte (1909-1947)	1	1	1	1	1
cohorte (1948-1956)	1.334 ***	0.839*	0.650 ***	2.309 ***	0.347 ***
cohorte (1957-1972)	1.256 **	0.506 ***	0.861*	3.127 ***	0.202 ***
homme	0.787 ***	0.517 ***	2.404 ***	1.082	0.485 ***
problèmes argent	0.814 **	-	-	-	-
constante	0.284 ***	0.564 ***	0.293 ***	0.120 ***	0.130 ***

*** Significatif au seuil de 1% ** Significatif au seuil de 5% * Significatif au seuil de 10%

plutôt qu'un autre. La variable de cohorte nous permet de voir l'évolution des chances d'appartenir à un groupe plutôt qu'à un autre ; ainsi, le groupe 2 et le groupe 3 montrent que les chances d'appartenir à ce groupe baissent dans les deux dernières cohortes par rapport à la première. Cet effet est encore plus marqué pour le groupe 5 (celui des individus ne quittant pas le foyer parental). Une variable supplémentaire a été introduite dans le modèle pour le groupe 1 ; ainsi, un individu n'ayant pas eu de problèmes d'argent dans sa jeunesse a moins de chances de se trouver dans le groupe 1.

3 Conclusion

Nous pouvons conclure que l'utilisation de l'optimal matching présente un intérêt certain pour l'analyse des parcours de vie. Couplée à une classification hiérarchique, elle a permis de mettre en évidence des groupes aux caractéristiques bien définies. Elle permet surtout d'aborder le parcours de vie dans sa totalité, en prenant en compte plusieurs événements, leur durée et leur chronologie et offre donc une perspective exploratoire intéressante pour l'analyse des séquences de manière générale.

Références

[1] A. Abbott, J. Forrest, « Optimal Matching Methods for Historical Sequences » *in Journal of Interdisciplinary History*, 26, 1986, pp. 471-494.

[2] A. Abbott, « A Primer on Sequence Methods » *in Organization Science*, Vol.1, No. 4, 1990, pp. 375-392.

[3] A. Abbott, A. Hrycak, « Measuring Resemblance in Sequence Data : An Optimal Matching Analysis of Musicians' Careers » *in The American Journal of Sociology*, Vol. 96, No. 1, Jul. 1990, pp. 144-185.

[4] A. Abbott, A. Tsay, « Sequence Analysis and Optimal Matching Methods in Sociology » *in Sociological Methods & Research*, Vol. 29, No. 1, Aug. 2000, pp. 3-33.

[5] J. B. Kruskal, «An overview of sequence comparison» *in David Sankof and Joseph B. Kruskal (eds), Time warps, string edits, and macromolecules. The theory and practice of sequence comparison*, Don Mills, Ontario : Adison-Wesley, 1983, pp. 1-44.

[6] S.B. Needlman, C. D. Wunsch, «A general method applicable to the search for similarities in the amino acid sequence of two proteins» *in Journal of Molecular Biology*, 48, 1970, pp. 443-453.

Étude de la classification des bactériophages

D. Nguyen[1], A. Boc[1], A. B. Diallo[1,2] et V. Makarenkov[1]

1. UQAM, CP8888, Succursale Centre-Ville, Montréal (Québec) Canada, H3C 3P8
2. McGill University, 3775 University Street, Montréal (Québec) Canada, H3A 2B4
(Nguyen.Van_Dung.2, Alix.Boc, Diallo.Banire, Makarenkov.Vladimir)@uqam.ca
Banire@mcb.mcgill.ca

Mots clés : classification arborescente, inférence phylogénétique, transfert horizontal de gène.

1. Introduction

L'évolution des bactériophages, qui sont des virus infectant les bactéries et les Archaea, est complexe à cause des mécanismes d'évolution réticulée comprenant le transfert horizontal de gènes (THG) et la recombinaison génétique. Une représentation phylogénétique sous forme de réseau est donc nécessaire pour interpréter l'histoire d'évolution des bactériophages [9]. Par ailleurs, la classification de ces micro-organismes présente intrinsèquement d'autres difficultés dues, d'une part, à la non-conservation de gènes au cours de leur évolution, et d'autre part, à l'hétérogénéité de leurs génomes.

Malgré leur abondance dans la biosphère [6], la classification des bactériophages n'est pas encore complètement établie. Il y a encore beaucoup de possibilités de l'affiner. Dans cet article, nous présentons une plate-forme d'inférence phylogénétique servant à tester les hypothèses sur l'évolution des phages, notamment : a) la reconstruction de l'arbre phylogénétique (i.e. classification) d'espèces de bactériophages, b) la détection et la validation des transferts horizontaux de gènes qui caractérisent leur évolution.

2. Méthodologie

Notre plate-forme d'inférence phylogénétique prend en entrée des données de séquences de protéines et retourne en sortie un *arbre phylogénétique d'espèces* reflétant l'histoire d'évolution des génomes des bactériophages ainsi que des *arbres de gènes* (i.e. des protéines) individuels représentant l'évolution de chacun des gènes considérés. Les différentes statistiques concernant les transferts horizontaux de gènes sont aussi rapportées (voir Figure 1). La méthodologie sous-jacente consiste en trois étapes : préparation de données extraites de la base de données GenBank de NCBI, inférence des arbres d'espèces et de gènes, et détection des THG.

En date de juillet 2006, nous avons recensé sur le site de NCBI, tout en s'assurant de la validité des références sur le site de ICTV (site ayant autorité officielle sur la taxonomie des virus), 163 génomes complets de bactériophages issus de 9 familles différentes dont une avec des annotations partielles (*unclassified*). Les données de séquences de protéines ont été extraites de la banque de données GenBank, en particulier, celles relatives aux VOG – *Viral Orthologous Groups* [1]. Les VOG sont des regroupements prédéfinis de protéines classés selon la fonction protéique à laquelle ils sont associés. Un VOG peut comprendre des séquences de plusieurs espèces différentes. Dans cette étude, 602 regroupements de VOG ont été considérés. L'étude phylogénétique des phages présente un défi double à cause de la grande variabilité à la fois dans la composition génétique et dans la taille des génomes. Le premier défi découle de la grande divergence des séquences de protéines [13]. Le second défi est dû aux tailles de génomes très variables, qui est d'ordre 2 de magnitude (le nombre de gènes codant en protéines varie de 8 à 381), en comparaison aux procaryotes (de ~400 à ~7 000 gènes) ou aux eucaryotes (de ~4 000 à ~60 000), qui sont d'ordre 1 de magnitude [9]. Bien que la meilleure façon de normaliser les génomes de ces micro-organismes en vue d'inférer leur histoire d'évolution reste un débat ouvert [12], la tendance actuelle est de combiner l'étude d'évolution du contenu de gènes et

l'analyse des alignements de chacune des protéines qui se retrouvent dans les génomes de plusieurs phages [9]. Les regroupements des protéines orthologues apportent des données nécessaires pour résoudre la première partie du problème. Reste à trouver le moyen de normaliser correctement.

Le point de départ de notre analyse consiste à estimer les distances entre génomes complets des espèces étudiées. Nous commençons par la construction d'une matrice binaire de présence et d'absence de gènes chez les espèces étudiées. Dans le cas des bactériophages cette matrice comprend 163 lignes (i.e. nombre d'espèces) et 602 colonnes (i.e. nombre de regroupements VOG) contenant des '1' (présence du gène dans le regroupement) et des '0' (absence du gène). Une matrice symétrique de distances inter-génomiques est ensuite calculée. Plusieurs types de distances ont été récemment utilisés pour mesurer la distance entre les génomes : le coefficient de corrélation standard [5], le coefficient de Jaccard [5], le coefficient de Maryland Bridge [12] et la Moyenne Pondérée [3]. Nous avons testé ces différents coefficients. Les résultats obtenus étaient très semblables, compte tenu qu'il n'y ait pas d'ordre *a priori* dans les regroupements VOG.

La matrice de dissimilarité entre les espèces sert ensuite à reconstruire, via l'algorithme Neighbor Joining (NJ) [14], l'arbre phylogénétique d'espèces. Parallèlement à l'algorithme NJ utilisant les distances inter-génomiques, l'approche d'inférence bayésienne, en utilisant le logiciel MrBayes [7], a été examinée. L'avantage de l'approche bayésienne est qu'elle peut traiter directement des caractères morphologiques '0' et '1' sans passer par le calcul de la matrice de distance. Elle suppose une distribution *a posteriori* des topologies d'arbres et utilise les méthodes Markov Chain Monte Carlo (MCMC), pour rechercher dans l'espace d'arbres et inférer la distribution *a posteriori* des topologies.

Pour chacune des deux approches, la validation statistique des topologies d'arbres obtenues a été effectuée. Dans le cas de l'algorithme NJ, le *bootstrap* a été utilisé : a) les données ont été aléatoirement échantillonnées avec remplacements afin de créer de multiple pseudo-données à partir des données d'origine : la matrice binaire d'espèces a été dupliquée en utilisant le programme SeqBoot (inclus dans le package PHYLIP [4]) en 100 copies ; b) avec les copies des matrices binaires, 100 matrices inter-génomiques ont été d'abord calculées, puis servies à reconstruire les arbres d'espèces avec NJ ; c) l'arbre de consensus suivant la règle de majorité étendue (≥50%) a été généré par le logiciel Consens (inclus dans PHYLIP [4]). Dans le cas de MrBayes, 2 millions de générations échantillonnées à toutes les 100 générations, avec 4 chaînes et 2 exécutions indépendantes ont été générées, créant ainsi 20 000 arbres. Un arbre de consensus a été produit à partir des 100 derniers arbres (*burning*=19 900) représentant 10 000 générations stationnaires. Les scores des branches représentent les probabilités *a posteriori* calculées à partir du consensus. En ce qui concerne l'inférence des arbres de gènes, ClustalW [15] a été utilisé pour aligner les séquences appartenant à chacun des 602 VOG. Comme dans le cas de l'arbre d'espèces, NJ et MrBayes ont été appliqués pour reconstruire les arbres de gènes : NJ utilise en entrée les matrices de distances, alors que MrBayes infère l'arbre directement à partir des séquences alignées. Les arbres phylogénétiques d'espèces inférés par NJ et MrBayes ont été très semblables. Toutefois, les scores de probabilités *a posteriori* fournis par MrBayes ont été généralement plus élevés (voir Figure 1) que les scores de bootstrap de NJ. Pour ces données donc l'approche bayésienne se montre plus efficace que celle basée sur les distances. De ce fait, l'approche bayésienne a été retenue pour la suite de nos travaux.

Finalement, la détection des THG a été effectuée, en utilisant le programme HGT Detection du package T-Rex [10], suivant la méthode de réconciliation topologique entre l'arbre de gènes et l'arbre d'espèces [11]. HGT Detection (voir le site www.trex.uqam.ca) prend en entrée un arbre d'espèces et un arbre de gènes pour le même ensemble espèces. Les THG sont ainsi calculés, en indiquant en sortie l'origine et la destination pour chacun des transferts inférés.

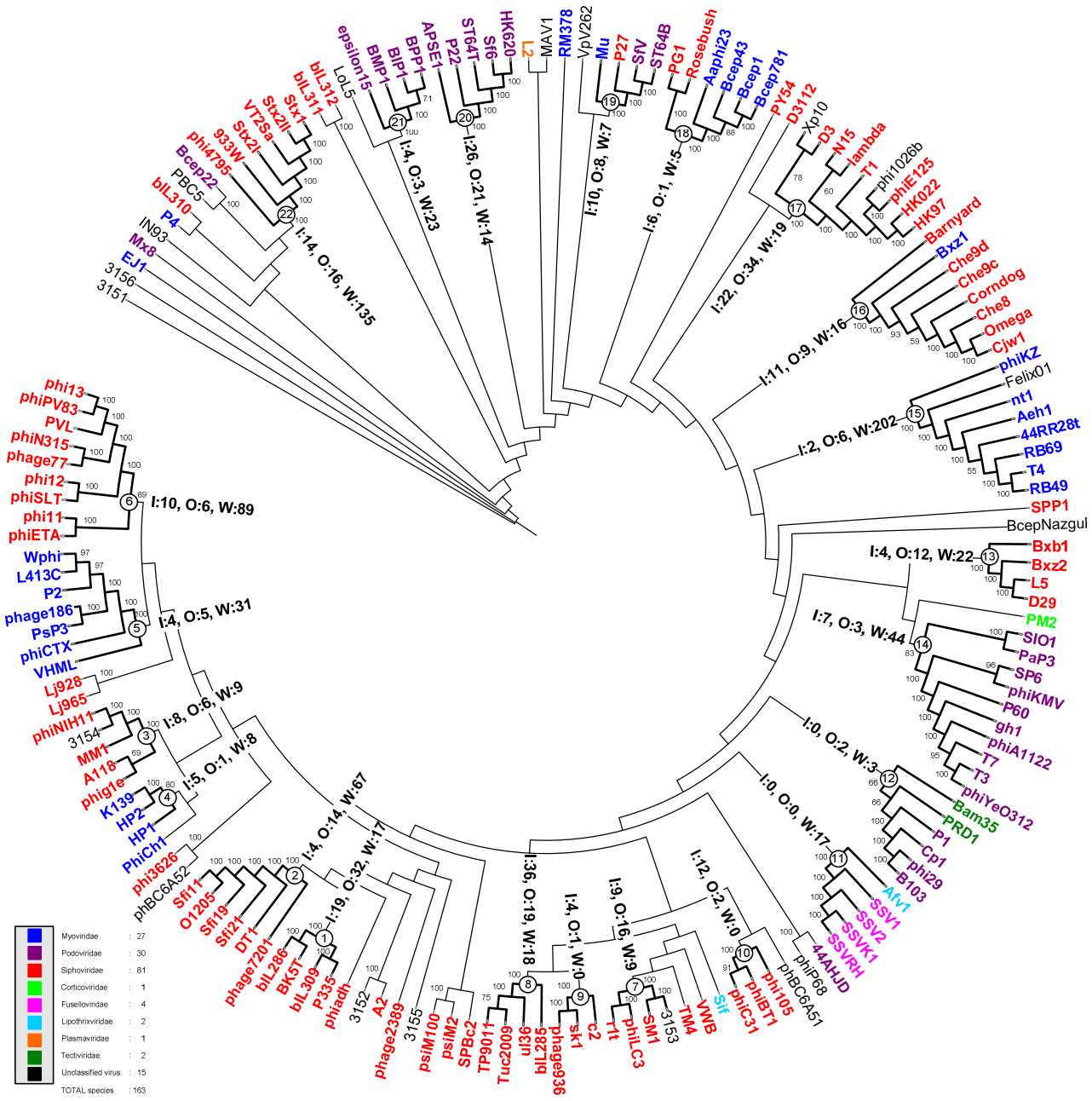


Figure 1 : Arbre phylogénétique d'espèces inféré par MrBayes [7] ; 12 des 22 groupes (représentés par des cercles) identifiés correspondent aux taxonomies de NCBI/ICTV. Pour chaque groupe, I (*In*) signifie le nombre de THG entrant dans le groupe, O (*Out*) le nombre de THG sortant du groupe et W (*Within*) le nombre de THG à l'intérieur de ce groupe. Figure générée avec l'outil iTol (*http://itol.embl.de/upload.cgi*).

3. Résultats et discussion

Figure 1 montre l'arbre phylogénétique de bactériophages ainsi que les différentes statistiques obtenues. De manière générale, l'arbre phylogénétique d'espèces, montrant un effet de chaîne, incorpore un grand nombre de signaux phylogénétiques capturés : au total, 122 phages, c-à-d 75% des génomes étudiés, ont été classés dans 22 groupes avec des scores de probabilités *a posteriori* supérieur à 55%. Ces groupes robustes contiennent entre 3 et 10 phages, avec une taille moyenne de clades de 6 espèces. Plusieurs familles d'espèces, 12 sur 22 groupes, référencées par NCBI/ICTV ont été retrouvées par notre analyse : *Siphoviridae* (groupes 1, 2, 6, 8, 9, 10, 13, 22), *Podoviridae* (groupes 14, 20, 21) et *Myoviridae* (groupe 4).

Au niveau des transferts, nous avons calculé les statistiques globales des THG intra (*Within*) et inter (*In/Out*) groupes. Plusieurs points sont remarquables : a) les groupes 2 à 6, 12 à 16, 18,

21 et 22 ont le nombre de transferts intra-groupes supérieur à ceux d'inter-groupes, alors que le reste des groupes a une tendance inverse, à l'exception cependant du groupe 11 qui ne donne ni reçoit de transferts, et des groupes 9 et 10 qui n'ont pas de transferts intra-groupes ; b) les groupes 1, 2, 5, 7, 12, 13, 15, 17 et 22 en donnent plus qu'ils en reçoivent, et inversement pour le reste ; c) les groupes qui en donnent beaucoup plus que la moyenne (informations non représentées sur la figure) sont les suivants : le groupe 1 au groupe 8 (23 transferts), le groupe 17 au groupe 20 (17 transferts) et le groupe 20 au groupe 17 (11 transferts). Les transferts entre les espèces hors groupes n'ont pas été comptabilisés dans cette étude.

Dans le futur proche, nous proposons de compléter cette analyse de l'évolution des bactériophages par une étape supplémentaire. Cette étape consistera à reconstruire les séquences de gènes ancestraux, en utilisant la méthode exposée dans [2], pour chacun des 602 VOG. Cette analyse permettra de déterminer le début de l'histoire évolutive de la fonction protéique associée au VOG en question. Une bonne reconstruction de la séquence protéique ancestrale peut nous aider dans différentes études telles que l'adaptation, le changement de comportement, la divergence fonctionnelle, etc. [8].

4. Bibliographie

[1] Y. Bao, S. Federhen, D. Leipe, V. Pham, S. Resenchuk, M. Rozanov, R. Tatusov, T. Tatusova, "NCBI Genomes Project", *Journal of Virology*, 78:7291-7298. 2004.

[2] A. B. Diallo, V. Makarenkov, M. Blanchette, *"Finding Maximum Likelihood Indel Scenarios"*, *Comparative Genomics*, 171-185. 2006.

[3] B. E. Dutilh, M. A. Huynen, W. J. Bruno, B. Snel, "The consistent phylogenetic signal in genome trees revealed by reducing the impact of noise", *J. Mol. Evol.* 58: 527-539. 2004.

[4] J. Felsenstein, *PHYLIP* (http://evolution.genetics.washington.edu/phylip.html - software download page and software manual) - *PHYLogeny Inference Package*. 2004.

[5] G. Glazko, A. Gordon, A. Mushegian, "The choice of optimal distance measure in genome-wide datasets", *T.P. Biol.* 61, 471–480. 2002.

[6] R. W. Hendrix, "Bacteriophages: evolution of the majority", *T.P. Biol.* 61, 471–480. 2002.

[7] J. P. Huelsenbeck, F. Ronquist, R. Nielsen, J. P. Bollback, "Bayesian inference of phylogeny and its impact on evolutionary biology", *Science* 294:2310–2314. 2001.

[8] N. Krishnan, H. Seligman, C. Stewart, A. Jason de Koning, D Pollock, "Ancestral sequence reconstruction in primate mitochondrial dna: Compositional bias and effect on functional inference", *Molecular Biology and Evolution*. 21 (10), 1871-1883. 2004.

[9] J. Liu, G. Glazko, A. Mushegian, "Protein repertoire of double-stranded DNA bacteriophages", *Virus Research*, 2006 Apr; 117(1):68-80. Epub 2006.

[10] V. Makarenkov, "T-Rex: reconstructing and visualizing phylogenetic trees and reticulation networks", *Bioinformatics* 17, 664-668. 2001.

[11] V. Makarenkov, A. Boc, C. F. Delwiche, A. B. Diallo, H. Philippe (2006). New efficient algorithm for modeling partial and complete gene transfer scenarios. Data Science and Classification, V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Ziberna (Eds.), IFCS 2006, Springer Verlag, pp. 341-349.

[12] B. Mirkin, E. Koonin, "A top-down method for building genome classification trees with linear binary hierarchies", DIMACS series in Discrete Math. & Theor. Computer Sci. 2003.

[13] F. Rohwer, R. Edwards, "The phage proteomic tree: a genome-based taxonomy for phage". J. Bacteriol. 184, 4529–4535. 2002.

[14] N. Saitou, M. Nei, "The Neighbor-Joining method: a new method for reconstructing phylogenetic trees", *Molecular Biology and Evolution*, 4:406-425. 1987.

[15] J. D. Thompson, D. G. Higgins and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice", *Nucleic Acids Res.*, 22:4673-4680. 1994.

un algorithme non itératif pour la classification d'observations bidimensionnelles

N. Paul, M. Terre et L. Fety

CNAM, 292, rue St-Martin, 75003 Paris
(nicolas.paul, michel.terre, luc.fety)@cnam.fr

Mots clés : apprentissage statistique, classification automatique

Dans cet article nous nous intéressons à la classification non supervisée d'observations à deux dimensions (observations complexes). Ces observations sont distribuées suivant une loi de mélange dont le nombre de composantes est supposé connu. Une récente présentation exhaustive des algorithmes de classification est disponible dans [1]. L'algorithme "Expectation-Maximization" (EM) [2] est la méthode la plus largement utilisée pour classer les observations et estimer les paramètres des différentes composantes. Si on dispose d'une représentation paramétrique des différentes composantes du mélange (exemple : mélange de Gaussiennes), l'algorithme EM permet de rechercher itérativement les paramètres des composantes qui maximisent la vraisemblance des observations. Le temps de convergence (nombre d'itérations) de l'algorithme EM peut être long, surtout si le nombre d'observations ou le nombre de composantes est élevé. De plus, rien ne garantit la convergence vers le maximum global de la vraisemblance. Les solutions pour éviter la convergence vers des maxima locaux consiste à affiner l'initialisation de l'algorithme [3] ou à utiliser des versions stochastiques de l'algorithme [4].

Nous proposons ici une nouvelle approche basée sur la minimisation d'un nouveau critère KP ("K Produits"). Ce critère a été introduit dans [5] pour classifier des données réelles (cas monodimensionnel). Il est ici généralisé au cas bidimensionnel. Soit $\{z_n\}_{n\in\{1\cdots N\}}$ un ensemble de N observations complexes distribué selon une loi de mélange à K composantes et soit $\mathbf{x} = (x_1, \cdots x_K)^t$ un vecteur quelconque de $\mathbb{C}^K$, nous définissons le critère KP comme étant la somme de tous les produits de K termes $\prod_{k=1}^{K} \|z_n - x_k\|^2$:

$$J : \mathbb{C}^K \to \mathbb{R}^+ : \ \mathbf{x} \to \sum_{n=1}^{N} \prod_{k=1}^{K} \|z_n - x_k\|^2 \tag{1}$$

Ce critère est clairement positif pour tout vecteur $\mathbf{x}$ de $\mathbb{C}^K$. Appelons $\mathbf{a} = (a_1, ..., a_K)^t$ le vecteur contenant les K moyennes des différentes composantes du mélanges. La première motivation, intuitive, pour définir J est le cas asymptotique où les variances des composantes sont toutes nulles. Dans ce cas, toutes les observations sont égales à l'un des a_k et $J(\mathbf{x})$ est minimal en $\mathbf{x} = \mathbf{a}$. Notre seconde et principale motivation est que, dans le cas général, le minimum de J peut être atteint par une simple résolution de système linéaire suivi de calcul des racines d'un polynôme d'ordre K :

Soit $\mathbf{z}_n \triangleq (z_n^{K-1}, z_n^{K-2} \cdots, 1)^t$, $\mathbf{z} \triangleq \sum_{n=1}^{N} \mathbf{z}_n^* z_n^K$ et $\mathbf{Z} \triangleq \sum_{n=1}^{N} \mathbf{z}_n^* \mathbf{z}_n^t$. $\mathbf{Z}$ est inversible si le nombre d'observations différentes est supérieure à K. Soit alors $\mathbf{y}_{min} = (y_{1,min}, ...y_{K,min})$ la solution de $\mathbf{Z}.\mathbf{y}_{min} = \mathbf{z}$ et soit $\mathbf{x}_{min}$ un vecteur de $\mathbb{C}^K$ contenant, dans un ordre quelconque, les K racines du polynôme $q_{\mathbf{y}_{min}}(\alpha) \triangleq \alpha^K - y_{1,min}\alpha^{K-1} - ... - y_{K-1,min}\alpha - y_{K,min}$,

TAB. 1 – algorithme KP : étapes et complexités

étape 1 : calcul d'un minimum de J
calcul de $\mathbf{Z}$ et $\mathbf{z}$: $\mathrm{O}(NK)$
calcul de $\mathbf{y}_{min}$, solution de $\mathbf{Z}.\mathbf{y}_{min} = \mathbf{z}$: $\mathrm{O}(K^2)$
calcul des racines $(x_{1,min}, \cdots, x_{K,min})$ de $q_{\mathbf{y}_{min}}(\alpha)$: $\mathrm{O}(K^2)$
étape 2 : association et estimation des moyennes
association de chaque z_n à la racine $x_{k,min}$ la plus proche : $\mathrm{O}(NK)$
calcul des K moyennes des groupes résultants : $\mathrm{O}(N)$

alors $\mathbf{x}_{min}$ est un minimum de J.

Le minimum de J est une estimation biaisée de $\mathbf{a}$. Par exemple, pour un mélange de $K = 2$ Gaussiennes complexes circulaires, équiprobables, de moyenne $-a$ et a et de variances $2\sigma_1^2$ et $2\sigma_2^2$, le minimum de J tend, lorsque le nombre d'observation tend vers l'infini, vers $\mathbf{x}_{min} = \left(-a\left(\sqrt{1+\lambda^2}-\lambda\right), a\left(\sqrt{1+\lambda^2}+\lambda\right)\right)^t$, avec $\lambda \stackrel{\Delta}{=} \frac{\sigma_2^2-\sigma_1^2}{a^2+\sigma_2^2+\sigma_1^2}$. Remarquons toutefois que l'estimation est non biaisée si les deux variances sont égales ($\sigma_2 = \sigma_1 \Rightarrow \lambda = 0$) et que le biais est négligeable si $\sigma_2^2 - \sigma_1^2 << a^2$.

L'algorithme complet d'estimation des modes consiste alors en deux étapes : lors de la première étape, le minimum de J, $\mathbf{x}_{min} = (x_{1,min}, ..., x_{K,min})^t$ est calculé et donne une première estimation grossière (biaisée) de l'ensemble des moyennes des composantes. Dans une seconde étape, chaque observation z_n est associée à la racines $x_{k,min}$ la plus proche. K ensembles sont ainsi formés, et la moyenne de chaque ensemble donne l'estimation finale des moyennes des composantes du mélange. Les deux étapes de l'algorithme sont détaillées dans la table 1. La complexité totale est en $\mathrm{O}(NK + K^2)$, ce qui est équivalent à $\mathrm{O}(NK)$ si le nombre d'observations N est supérieur au nombre de composantes K.

Les performances de notre algorithme ont été évaluées sur le mélange de quatre Gaussiennes complexes décrit dans la table 2 où le paramètre σ permet d'envisager plusieurs conditions de mélanges (chevauchement plus ou moins important des composantes). La figure 1 donne un exemple de 100 observations pour $\sigma = 0.2$. Les résultats de simulations sont présentés dans la figure 2 pour σ variant de 0 à 0.5. Pour chaque valeur de σ, 1000 simulations ont été réalisées. Pour chaque simulation, 100 observations ont été générées selon la loi de mélange. Les performances de KP sont comparées à celles de l'algorithme EM avec une initialisation uniforme dans la zone d'observation. Notre critère de performances est la précision sur l'estimation des quatre moyennes. Pour s'affranchir de l'ambiguïté de permutation du vecteur de moyennes estimées, nous définissons l'erreur d'estimation e_r comme étant la distance maximale entre le vecteur des moyennes et la meilleure des $K!$ permutations du vecteur des moyennes estimées :

$$e_r \stackrel{\Delta}{=} \min_{\text{permutations}} \mathrm{N}\left(\mathbf{a} - \text{permutations}(\hat{\mathbf{a}}_r)\right)$$

où $\mathrm{N}(\mathbf{x}) \stackrel{\Delta}{=} \max_{k\in\{1\cdots K\}} \|x_k\|$. Sur la figure 2 on constate que dès que σ n'est pas nul l'algorithme EM a une probabilité non nulle de converger vers une mauvaise solution ($p(e_r > 0.5) =$

TAB. 2 – scenario de simulation

moyennes	covariances	poids
1	$\begin{pmatrix} \sigma^2 & 0.75\sigma^2 \\ 0.75\sigma^2 & \sigma^2 \end{pmatrix}$	0.3
i	$\begin{pmatrix} \sigma^2 & 0 \\ 0 & \frac{\sigma^2}{4} \end{pmatrix}$	0.3
-1	$\begin{pmatrix} \frac{\sigma^2}{4} & 0 \\ 0 & \sigma^2 \end{pmatrix}$	0.2
$-i$	$\begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$	0.2

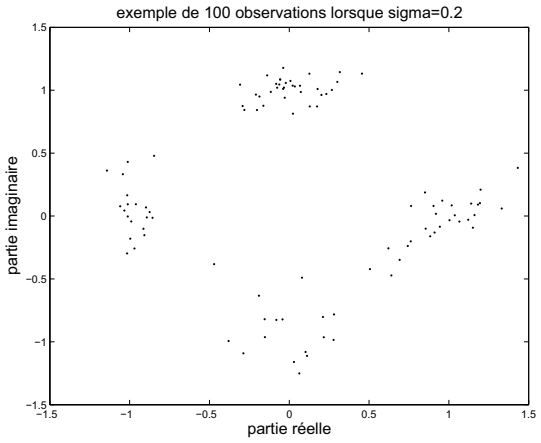


FIG. 1 – 100 observations du mélange de quatre gaussiennes décrit table 2 pour $\sigma = 0.2$

0.35 pour $\sigma = 0.1$). Par contre, l'algorithme KP reste précis pour $\sigma < 0.15$ ($p(e_r < 0.1) = 0.95$ pour $\sigma = 0.15$) et relativement précis pour $\sigma < 0.25$ ($p(e_r < 0.2) = 0.95$ pour $\sigma = 0.25$). L'extension de notre approche à des observations $\mathbf{z}_n$ de dimension D quelconque est maintenant envisagée, et nous étudions la minimisation du critère associant à un ensemble de K vecteurs $\{\mathbf{x}_k\}_{k\in\{1\cdots K\}}$ la somme des N produits de K termes $\prod_{k=1}^{K} ||\mathbf{z}_n - \mathbf{x}_k||^2_{\mathbb{R}^D}$.

Références

[1] Berkin P (2006) A Survey of clustering data mining techniques. Grouping Multidimensional Data : Recent Advances in Clustering, Ed. J. Kogan and C. Nicholas and M. Teboulle, Springer, pp. 25-71

[2] Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, B. 39, pp. 1-38

[3] McLachlan G, Peel D (2000) Finite Mixture Models. Wiley Series in probability and statistics, John Wiley and Sons

[4] Celeux G, Chauveau D, Diebolt J (1995) On stochastic version of the EM algorithm. INRIA research report no 2514, available : http ://www.inria.fr/rrrt/rr-2514.html

[5] Paul N, Terre M, Fety L (2007) A global algorithm for clustering univariate observations, Advances in Data Analysis and Classification, Springer ed., soumis (Mars 2007), disponible : http ://arxiv.org/PS_cache/physics/pdf/0703/0703281v1.pdf

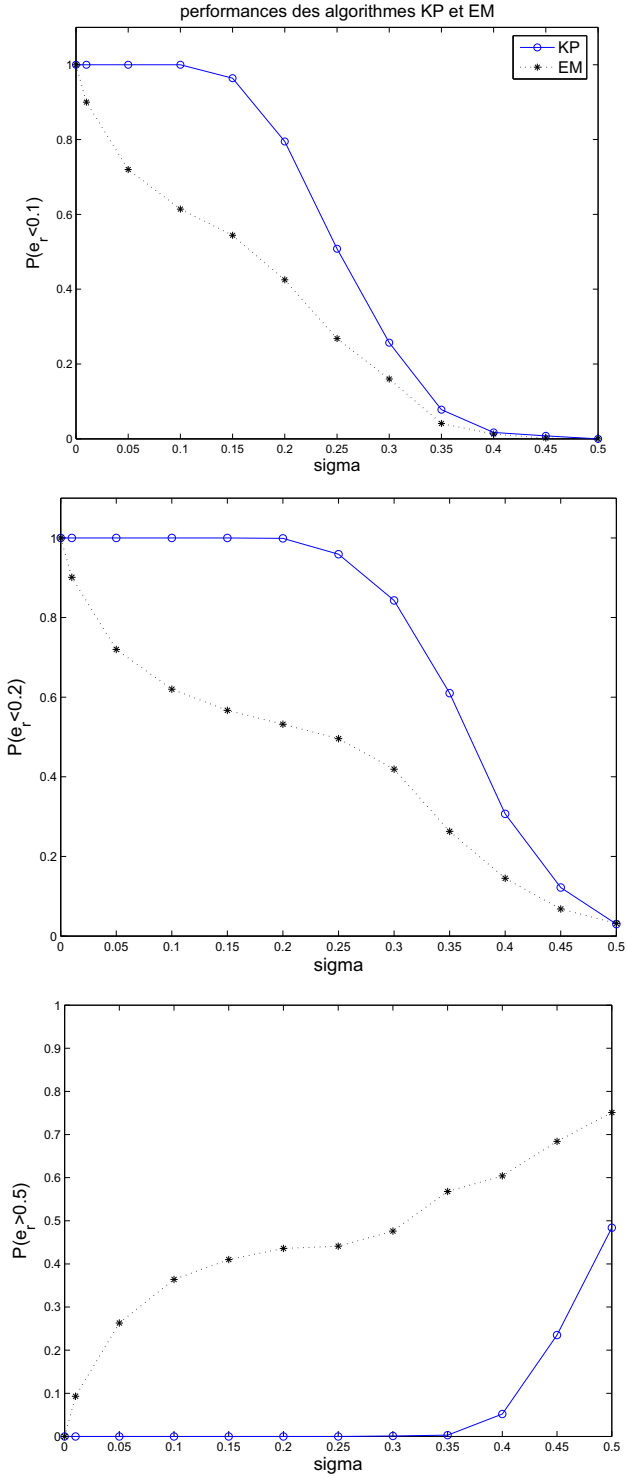


FIG. 2 – performances des algorithmes EM et KP sur un mélange de quatre gaussiennes complexes décrit dans la table 2 pour différentes valeurs de σ. Pour chaque σ, 1000 simulations ont été réalisées. Pour chaque simulation, 100 observations ont été générées. e_r est la distance maximale entre le vecteur contenant la meilleure permutation des modes estimés et le vrai vecteur de modes. Les critères de performances sont les probabilités que e_r soit inférieur à 0.1 (figure du haut), inférieur à 0.2 (figure du milieu) et supérieur à 0.5 (figure du bas).

Extraction de concepts et de relations en analyse relationnelle de concepts (ARC)

M.H. Rouane[2], M. Huchard[1], A. Napoli[2], P. Valtchev[3]

1. LIRMM, 161 rue Ada, 34392 Montpellier
2. LORIA, Campus Sciences, BP 239, 54506 Vandœuvre-lès-Nancy
3. Dépt. d'informatique, UQAM, CP 8888, succ. Centre-Ville, Montréal, Canada, H3C 3P8
rouanehm@loria.fr, huchard@lirmm.fr, napoli@loria.fr, valtchev.petko@uqam.ca

Mots clés : analyse formelle de concepts, analyse relationnelle de concepts, treillis de concepts, extraction de connaissances, traitement de relations

Résumé

Ce résumé introduit l'analyse relationnelle de concepts (ARC), qui est une extension de l'analyse formelle de concepts (AFC, en anglais *Formal concept analysis*), formalisée dans [4]. La théorie de l'AFC s'appuie sur les connexions de Galois pour concevoir un treillis de concepts (formels) à partir d'un tableau binaire de données. Sur la base du treillis de concepts peuvent en outre être extraits des motifs et des règles d'association, qui jouent un rôle prépondérant en découverte de connaissances dans les données. Ici, le formalisme de l'ARC s'appuie sur l'AFC et permet en plus l'extraction de relations entre concepts, où le domaine et le co-domaine d'une relation sont des concepts formels d'un treillis de concepts d'origine. Le processus de l'ARC s'appuie sur la construction progressive d'une famille de treillis de concepts et converge pour donner un treillis final, qui peut ensuite être traduit dans les termes d'un langage de représentation des connaissances comme une logique de description, pour pouvoir servir de support à une ontologie. l'ARC est une extension opérationnelle importante de l'AFC, en lien direct avec les préoccupation liées à la représentation des connaissances et à la formalisation du raisonnement.

1 Introduction

Cet article porte sur les processus d'*analyses formelle et relationnelle de concepts* (AFC et ARC), qui favorisent l'interopérabilité entre processus d'extraction et de représentation des connaissances, spécialement dans le cadre des applications liées au Web sémantique [3]. Les techniques et méthodes actuelles de fouille de données doivent être en mesure de traiter des données complexes, par exemple des données relationnelles, qui, après traitement, pourront être exploitées dans le cadre d'un système à base de connaissances (avec de langages de représentation des connaissances comme OWL ou SWRL).

Par rapport à cet objectif, nous nous intéressons à la classification de données relationnelles dans des structures comme des *concepts* (prédicats unaires), des *propriétés* fonctionnelles (attributs) et relationnelles (prédicats binaires). Le processus de l'AFC permet de construire un treillis de concepts à partir d'un tableau binaire de données, traitant par là essentiellement le cas des attributs binaires. Ici, nous étendons les possibilités de l'AFC en autorisant le traitement de propriétés relationnelles dans le cadre de l'analyse relationnelle de concepts. Le processus de ARC repose sur trois éléments originaux : un

modèle de données calqué sur le modèle entité-association, une méthode d'échelonnage conceptuelle paramétrable qui traduit les relations entre objets en propriétés ou en "attributs relationnels", et enfin une méthode itérative de construction de treillis de concepts où les concepts sont décrits à la fois par des attributs fonctionnels et relationnels.

Le processus d'extraction s'appuie sur la construction progressive d'une famille de treillis de concepts et converge pour donner un treillis final, qui est en fait un point fixe du processus de création de la famille de treillis de concepts. Un des mécanismes sous-jacents est l'*échelonnage* ou *graduation conceptuels* (de l'anglais *conceptual scaling*) — pour la construction d'échelles conceptuelles — qui permet de traiter un attribut relationnel (non binaire) comme un attribut binaire (appartenance ou non à un objet). L'échelle pour un attribut est un treillis qui détermine le co-domaine de l'attribut relationnel dans le formalisme de l'ARC.

Les avantages du processus d'ARC sont tout d'abord d'étendre les possibilités déjà très riches de l'AFC, en reprenant l'ensemble de la systématique pratique et théorique développée pour l'AFC, mais aussi d'offrir des structures complexes pouvant servir de support à une ontologie codée dans un langage de représentation des connaissances comme une logique de descriptions [1, 7].

Ainsi, ces travaux non seulement s'inscrivent dans la problématique de la classification pour l'extraction de connaissances mais aussi dans la problématique de la classification pour la représentation des connaissances. Cet article présente plutôt brièvement les travaux sur l'ARC. L'ensemble du processus de l'ARC est décrit plus en détails dans [6] et dans la thèse de M.H. Rouane [5].

2 De l'AFC à l'ARC

L'AFC consiste à construire un treillis de concepts à partir d'un tableau binaire *Objets* $\times$ *Attributs* [4, 2]. Formellement, un *contexte* $\mathcal{K}$ est la donnée d'un triplet (O, A, I) où O est un ensemble d'objets, A est un ensemble d'attributs, et $I \subseteq O \times A$ une relation entre O et A. Le processus de l'AFC cherche à construire un treillis de concepts, où un concept regroupe un ensemble (maximal) d'objets partageant un ensemble (maximal) d'attributs, ce qui se formalise par la définition de la connexion de Galois suivante (dénotée classiquement par l'apostrophe $'$) : $' : \mathcal{P}(O) \rightarrow \mathcal{P}(A)$; $\quad X' = \{a \in A \mid \forall o \in X, oIa\}$.

Un théorème de base stipule que les ensembles maximaux d'objets (appelés *extents*) sont en correspondance bijective avec les ensembles maximaux d'attributs correspondants (attributs que tous les objets possèdent, appelés *intents*). Les couples $(X, Y) \in \mathcal{P}(O) \times \mathcal{P}(A)$ d'ensembles en correspondance — ils vérifient $X = Y'$ et $Y = X'$ — sont appelés des *concepts (formels)* et forment un *treillis* complet par rapport à l'ordre de l'inclusion des "extents" (X). Ce treillis est appelé treillis des concepts et sa construction est une tâche primordiale en AFC.

L'AFC est bien adaptée au traitement de données se présentant sous forme de tableaux binaires ou pouvant s'y rapporter. Ainsi, l'échelonnage conceptuel permet de traduire des données multi-valuées non binaires en données binaires par l'intermédiaire de prédicats qui segmentent les attributs non binaires en attributs binaires : le domaine de valeur $\{1, 2, 3\}$ pour l'attribut a va être transformé en trois attributs binaires, $a = 1$, $a = 2$ et $a = 3$. Toutefois, la machinerie de l'AFC est limitée lorsqu'il s'agit de traiter des tableaux de données où figurent des relations (binaires) entre les objets.

Le modèle de données de l'ARC est la *famille de contextes relationnels* (FCR) notée $\mathcal{R} = (\mathbf{K}, \mathbf{R})$, où $\mathbf{K}$ est un ensemble de contextes $\mathcal{K}_i = (O_i, A_i, I_i)$, $\mathbf{R}$ un ensemble de

relations $r_k \subseteq O_i \times O_j$, où O_i et O_j sont les ensembles d'objets des contextes formels $\mathcal{K}_i$ and $\mathcal{K}_j$.

Une relation r, par exemple "les chercheurs qui sont auteurs de papiers", met en correspondance des objets d'un contexte $\mathcal{K}_i$, le *domaine* de r, avec des objets d'un contexte $\mathcal{K}_j$, le *co-domaine* de r. L'idée ici est de reprendre le processus de l'AFC et de l'adapter au traitement de données relationnelles par l'intermédiaire d'un échelonnage particulier, où les prédicats vont être associés à des concepts dans un treillis : en fait, les prédicats associés aux échelles ne traduisent plus un ordre total comme c'est le plus souvent le cas (cf. exemple ci-dessus) mais un ordre partiel. La prise en compte d'une relation r entre les objets de O_i dans $\mathcal{K}_i$ et les objets de O_j dans $\mathcal{K}_j$ consiste à considérer les objets o_j donnés par les liens où intervient r pour un objet o_i dans O_i, dénoté par $r(o_i)$, et à comparer o_j à l'"extent" d'un concept c_j du contexte $\mathcal{K}_j$ — $r(o_i)$ est-il ou non dans l'"extent" de c_j — le treillis associé à $\mathcal{K}_j$ ayant été au préalablement construit selon les principes de l'AFC. Ainsi, si $r(o_i)$ appartient à l'"extent" du concept c_j, l'attribut noté $r : c_j$ peut être assigné à l'objet o_i, comme cela va être détaillé ci-après.

3 Un exemple

Cette FCR comprend un contexte unique et deux relations binaires : dans le contexte, les objets dénotent des papiers auxquels sont associés des attributs relatifs aux sujets traités par ces papiers, soit ingénierie du logiciel (*se*), théorie des treillis (*lt*), interface homme machine (*mmi*). La relation *cites* modélise la citation entre papiers tandis que la relation *develops* indique qu'un papier et une extension d'un autre papier. La figure 1 montre l'exemple de FCR et le treillis correspondant construit selon les principes de l'AFC.

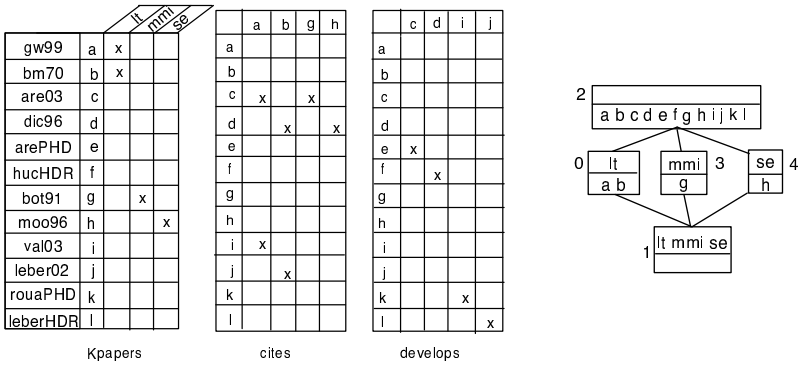


FIG. 1 – La FCR sur les papiers : le tableau (objets × attributs) à gauche, les tableaux des deux relations *cites* et *develops* au milieu, et le treillis de concepts en AFC correspondant (à droite).

L'exemple montre les attributs qui caractérisent les papiers, et les liens entre papiers :
- $I \subseteq O \times A$; $I = \{(a, lt), (b, lt), (g, mmi), (h, se)\}$,
- $cites \subseteq O \times O$; $cites = \{(c, a), (c, g), (d, b), (d, h), (i, a), (j, b)\}$,
- $develops \subseteq O \times O$; $develops = \{(e, c), (f, d), (k, i), (l, j)\}$.

Étant donné une relation r entre les objets des contextes $\mathcal{K}_i$ et $\mathcal{K}_j$, il existe plusieurs façons d'assigner un attribut du type $r : c_j$ à un objet o_i du contexte $\mathcal{K}_i$. Ici, une seule façon de faire est considérée, l'*échelonnage existentiel* (voir les détails et les autres façons

de faire dans [5] où cet échelonnage est appelé *wide scaling*) : l'attribut $r : c_j$ est assigné à l'objet o_i si la condition $r(o_i) \cap extent(c_j) \neq \emptyset$ est satisfaite. C'est sur cette base que repose le processus de RCA.

4 Le processus de RCA

Nous illustrons le processus de RCA avec échelonnage existentiel sur l'exemple introduit ci-avant.

- La première étape consiste à construire le treillis de concepts selon les principes de l'AFC en considérant le tableau binaire initial (voir figure 1).
- Lors de l'échelonnage de la relation *cites*, seules les descriptions des objets c, d, i, j vont changer dans un premier temps. Les objets i et j se voient assigner les attributs *cites :c0* et *cites :c2* car, par exemple, i est en relation par *cites* avec a et a est dans l'"extent" des concepts $c0$ et $c2$ dans le treillis de la figure 1, tandis que c a pour attributs *cites :c0*, *cites :c2* et *cites :c3*, et d a *cites :c0*, *cites :c2* et *cites :c4*.
- Le même processus appliqué à la relation *develops* provoque l'assignation de l'attribut *develops :c2* aux objets e, f, k, l.
  Un nouveau contexte et un nouveau treillis vont pouvoir être construits comme le montre la figure 2.
- Le processus est ré-appliqué pour la relation *cites* mais ne provoque aucune modification : par exemple, i et j ne sont en relation qu'avec eux-mêmes par l'intermédiaire de *cites :c0* et *cites :c2*.
- Le processus est ré-appliqué sur la relation *develops* qui provoque la création de nouveaux attributs : *develops :c2*, *develops :c5*, *develops :c6* et *develops :c7*. Par exemple, e est en relation par *develops* avec c qui est dans l'"extent" des concepts *c2*, *c5* et *c6*, pour f ce sont les concepts *c2*, *c5* et *c7*.
- Finalement, le nouveau contexte et le treillis associé sont construits et donnés à la figure 3. Une nouvelle application de l'échelonnage sur les relations *cites* et *develops* ne provoque plus aucune création de nouvel attribut : le point fixe du processus est atteint et le treillis final est obtenu.

Il est intéressant de voir comment peuvent être interprétés les concepts obtenus. Les nouveaux concepts *c9*et *c10* représentent des papiers qui développent d'autres papiers, qui eux-mêmes citent des papiers sur *lt* et *mmi* pour *c9* et sur *lt* et *se* pour *c10*. De plus, le concept *c8* (déjà existant à l'étape précédente) regroupe les papiers qui développent des papiers citant eux-mêmes des papiers sur *lt*.

		lt	mmi	se	cites 2	cites 0	cites 3	cites 4	develops 2
gw99	a	x							
bm70	b	x							
are03	c				x	x	x		
dic96	d				x	x		x	
arePHD	e								x
hucHDR	f								x
bot91	g		x						
moo96	h			x					
val03	i				x	x			
leber02	j				x	x			
rouaPHD	k								x
leberHDR	l								x

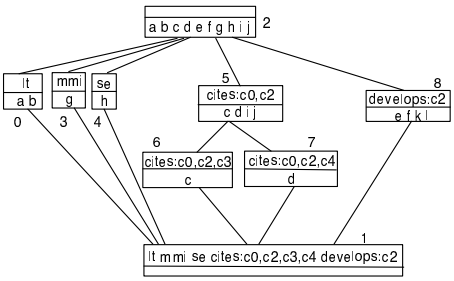


FIG. 2 – Le nouveau contexte à gauche et le treillis correspondant à droite.

		lt	mmi	se	cites 2	cites 0	cites 3	cites 4	develops 2	develops 5	develops 6	develops 7
gw99	a	x										
bm70	b	x										
are03	c				x	x	x					
dic96	d				x	x		x				
arePHD	e								x	x	x	
hucHDR	f								x	x		x
bot91	g		x									
moo96	h			x								
val03	i				x	x						
leber02	j				x	x						
rouaPHD	k								x	x		
leberHDR	l								x	x		

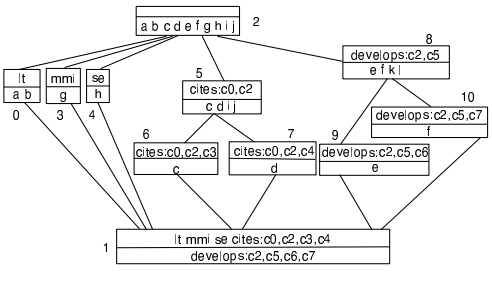


FIG. 3 – Le dernier contexte à gauche et le treillis correspondant à droite.

5 Conclusion

Le processus de l'ARC qui vient d'être brièvement illustré est une extension du processus bien connu de l'AFC, qui permet de traiter des données relationnelles (qui sont assez abondantes en ingénierie). C'est aussi un pas important vers une meilleure interopérabilité entre formalismes de classification et d'apprentissage comme l'AFC et l'ARC et formalismes de représentation des connaissances comme les logiques de descriptions. Ici, il est possible de qualifier l'AFC et l'ARC de processus d'extraction de connaissances à partir de données binaires et relationnelles. L'AFC permet de construire le treillis à partir du tableau binaire (le schéma de la base de connaissance ou de l'ontologie), treillis où ne figurent que des relations verticales de spécialisation entre concepts. L'ARC permet d'établir les relations — horizontales — entre les concepts dans le treillis, qui se réifient par des rôles en logiques de descriptions. Il reste encore beaucoup de choses à faire pour obtenir un système véritablement opérationnel, notamment étudier les problèmes posés par le passage à l'échelle, l'efficacité dans la construction des treillis, la mise au point et la combinaison de plusieurs modes d'échelonnage conceptuel (cardinalité sur les attributs).

Références

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, 2003.

[2] M. Barbut and B. Monjardet. *Ordre et classification – Algèbre et combinatoire (2 tomes)*. Hachette, Paris, 1970.

[3] D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors. *Spinning the Semantic Web*. The MIT Press, Cambridge, Massachusetts, 2003.

[4] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer, Berlin, 1999.

[5] M.H. Rouane. *Étude de l'analyse formelle dans les données relationelles — Application à la restructuration des modèles structuraux UML*. Phd thesis, DIRO, Université de Montréal, 2006.

[6] M.H. Rouane, M. Huchard, A. Napoli, and P. Valtchev. A proposal for combining formal concept analysis and description logics for mining relational data. In S.O. Kuznetsov and S. Schmidt, editors, *Proceedings of the 5th International Conference on Formal Concept Analysis (ICFCA 2007), Clermont-Ferrand*, LNAI 4390, pages 51–65. Springer, Berlin, 2007.

[7] S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer, Berlin, 2004.

Classification automatique pour la segmentation de signaux unidimensionnels

A. Samé[1], P. Aknin[1], G. Govaert[2]

1. Institut National de Recherche sur les Transports et leur Sécurité
2 avenue du général Malleret-Joinville 94114, Arcueil, France
2. Université de Technologie de Compiègne, HEUDIASYC UMR CNRS 6599
Centre de Recherches de Royallieu, BP 20529, F-60205 Compiègne, France
(same, aknin)@inrets.fr ; govaert@utc.fr

Mots clés : Diagnostic ferroviaire, classification automatique, régression typologique, optimisation, segmentation de signaux

1 Introduction

La Transmission Voie-Machine (TVM) sur les lignes à grande vitesse du réseau ferroviaire Français est utilisée pour transmettre aux conducteurs, par le biais de signaux, des consignes de sécurité telles que la vitesse maximale à ne pas dépasser. Son diagnostic [1] nécessite préalablement de segmenter, en arcs homogènes, des signaux unidimensionnels d'inspection ferroviaire (voir figure 1).

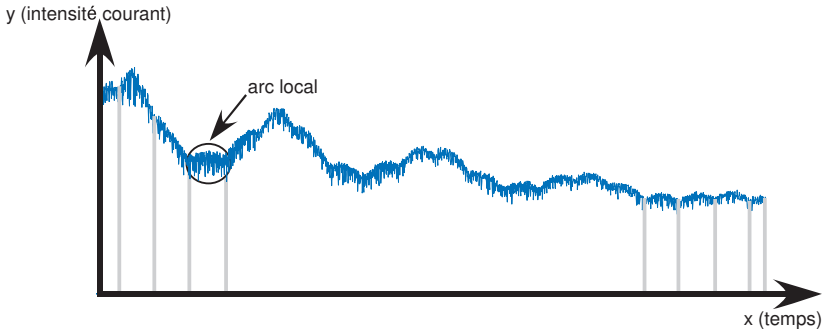


Figure 1: Exemple de signal réel du système de Transmission Voie-Machine.

Plusieurs approches ont été abordées dans le cadre de la régression typologique, appelée aussi régression linéaire par morceaux, pour résoudre implicitement ou explicitement le problème de segmentation de signaux unidimensionnels. Nous pouvons notamment faire référence aux approches de Charles [2], Ferrari-Trecate [3] et Fisher [4].

Dans le cadre du diagnostic de la TVM, nous proposons un nouvel algorithme de classification automatique dédié au partitionnement de signaux unidimensionnels en zones homogènes. L'algorithme proposé permet de minimiser itérativement un critère similaire au critère d'inertie intra-classes, sous la contrainte que les classes (arcs bruités du signal) soient ordonnées par rapport au temps. En outre, ll nécessite moins de temps de calcul que l'algorithme de Fisher [4].

Nous représentons un signal par n couples d'observations $(x_1, y_1), \ldots, (x_n, y_n)$ où $x_1, \ldots, x_n$ représentent des instants ordonnés, et y_i représente le signal à l'instant x_i.

2 Algorithme de classification proposé

L'algorithme proposé pour la segmentation des signaux consiste à partitioner le signal en K classes ordonnées suivant le temps. Une telle partition peut s'écrire $P_{n,K} = \big([x_{j_1}; x_{j_2}[, \ldots, [x_{j_{K-1}}; x_{j_K}[, [x_{j_K}; x_{j_{K+1}}]\big)$, où $\mathcal{J}_{n,K} = (j_1, \ldots, j_{K+1})$ désigne une suite croissante de $(1, \ldots, n)$ telle que $j_1 = 1$ et $j_{K+1} = n$. La partition $P_{n,K}$ étant entièrement déterminée par $\mathcal{J}_{n,K}$, nous l'assimilons dans la suite de l'article à $\mathcal{J}_{n,K}$.

2.1 Critère optimisé

Dans le cadre de notre application, le critère à minimiser est défini par l'écart entre le signal observé et un signal théorique représenté par le polynôme de régression du second degré par morceaux. Il s'écrit : $C_1(\mathcal{J}_{n,K}) = \min_{\boldsymbol{W}} C_2(\mathcal{J}_{n,K}, \boldsymbol{W})$, avec

$$C_2\big(\mathcal{J}_{n,K}, \boldsymbol{W}\big) = \sum_{k=1}^{K-1} \sum_{i=j_k}^{j_{k+1}-1} (y_i - \boldsymbol{z}_i^T \boldsymbol{w}_k)^2 + \sum_{i=j_K}^{j_{K+1}} (y_i - \boldsymbol{z}_i^T \boldsymbol{w}_K)^2, \tag{1}$$

où $\boldsymbol{w}_k = [w_{k0}\ w_{k1}\ w_{k2}]^T$ et $\boldsymbol{z}_i = [1\ x_i\ x_i^2]^T$ sont respectivement le vecteur des coefficients et le vecteur des monômes associés à chaque polynôme de degré 2, et $\boldsymbol{W}$ désigne la matrice des coefficients polynômiaux $w_{k\ell}$ ($1 \leq k \leq K$ et $0 \leq \ell \leq 2$).

2.2 Algorithme proposé

Rappelons d'abord que l'algorithme de Fisher [4] est un algorithme de programmation dynamique permettant de calculer la partition optimale qui minimise un critère additif $C(\mathcal{J}_{n,K}) = \sum_{k=1}^{K-1} D(j_k, j_{k+1} - 1) + D(j_K, j_{K+1})$, où $D(\alpha, \beta)$ est appelé *diamètre* associé à la classe $[x_\alpha, x_\beta]$. On peut remarquer que le critère C_1 que nous cherchons à minimiser sur l'ensemble des suites croissantes $\mathcal{J}_{n,K}$ peut s'écrire sous la forme additive du critère C, en considérant le diamètre $D(\alpha, \beta) = \min_{\boldsymbol{w}} \sum_{i=\alpha}^{\beta} (y_i - \boldsymbol{z}_i^T \boldsymbol{w})^2 = \sum_{i=\alpha}^{\beta} (y_i - \boldsymbol{z}_i^T \widehat{\boldsymbol{w}})^2$, avec $\widehat{\boldsymbol{w}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{y}$, où $\boldsymbol{\Phi} = [\boldsymbol{z}_\alpha, \ldots, \boldsymbol{z}_\beta]^T$ et $\mathbf{y} = [y_\alpha, \ldots, y_\beta]^T$. La complexité algorithmique de l'algorithme de Fisher est en grande partie liée au calcul initial de la matrice triangulaire supérieure formée des diamètres $D(\alpha, \beta)$ ($1 \leq \alpha \leq \beta \leq n$) [4]. Pour des diamètres plus simples à calculer comme le diamètre $D(\alpha, \beta) = \min_v \sum_{i=\alpha}^{\beta} (y_i - v)^2$, sa complexité est en $O(Kn^2)$ [4]. Cependant, pour le diamètre que nous avons utilisé, l'algorithme de Fisher a une complexité en $O(Kn^3)$, puisque chaque calcul de diamètre, qui nécessite de calculer le produit matriciel $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ puis de l'inverser, s'effectue en $O(n)$.

Le nouvel algorithme que nous proposons permet de minimiser itérativement le critère C_1, tout en nécessitant moins de calculs que l'algorithme de Fisher. Il est Basé sur le fait que la minimisation du critère C_1 est équivalente à la minimisation du critère C_2, puisque $\min_{\mathcal{J}_{n,K}} C_1(\mathcal{J}_{n,K}) = \min_{\mathcal{J}_{n,K}} (\min_{\boldsymbol{W}} C_2(\mathcal{J}_{n,K}, \boldsymbol{W})) = \min_{(\mathcal{J}_{n,K}, \boldsymbol{W})} C_2(\mathcal{J}_{n,K}, \boldsymbol{W})$. L'algorithme proposé part d'une partition initiale $\mathcal{J}_{n,K}^{(0)} = (j_1^{(0)}, \ldots, j_{K+1}^{(0)})$ puis alterne les deux étapes suivantes jusqu'à la convergence.

Etape 1 (itération q) Calcul des coefficients de régression minimisant $C_2\big(\mathcal{J}_{n,K}^{(q)}, \boldsymbol{W}\big)$. On peut montrer que cette minimisation s'obtient en effectuant K régressions :

$\boldsymbol{w}_k^{(q)} = \arg\min_{\boldsymbol{w}} \sum_{i=j_k^{(q)}}^{j_{k+1}^{(q)}-1} (y_i - \boldsymbol{z}_i^T \boldsymbol{w})^2 \quad \forall k = 1, \ldots, K-1,$

$\boldsymbol{w}_K^{(q)} = \arg\min_{\boldsymbol{w}} \sum_{i=j_K^{(q)}}^{j_{K+1}^{(q)}} (y_i - \boldsymbol{z}_i^T \boldsymbol{w})^2.$

Etape 2 (itération q) Calcul de la partition $\mathcal{J}_{n,K}^{(q+1)}$ minimisant $C_2\big(\mathcal{J}_{n,K}, (\boldsymbol{w}_k^{(q)})\big) = \sum_{k=1}^{K-1} D(j_k, j_{k+1}-1, \boldsymbol{w}_k^{(q)}) + D(j_K, j_{K+1}, \boldsymbol{w}_K^{(q)})$, où $D(\alpha, \beta, \boldsymbol{w}) = \sum_{i=\alpha}^{\beta}(y_i - \boldsymbol{z}_i^T\boldsymbol{w})^2$. Le critère à minimiser pouvant s'écrire de manière additive par rapport aux classes, nous effectuons cette étape par un programme dynamique similaire à celui de Fisher [4] mais dans lequel aucun calcul de produit matriciel $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ n'est nécessaire pour évaluer la matrice des diamètres ; ce qui rend l'algorithme plus rapide. Cette procédure garantit particulièrement l'obtention d'une partition formée de classes respectant la contrainte d'ordre par rapport au temps.

Convergence et complexité algorithmique Chaque itération de l'algorithme qui vient d'être décrit fait décroître le critère C_2, ce qui lui confère des propriétés de convergence locale. En pratique, l'algorithme est arrêté lorsque le taux de décroissance du critère devient plus petit qu'un seuil fixé. Sur les exemples traités, l'algorithme converge, en général, en une dizaine d'itérations. Comme l'algorithme des k-means, celui-ci est lancé plusieurs fois afin de sélectionner le meilleur optimum local. Sa complexité est en $O(T \times I \times K \times n^2)$, où T et I sont respectivement le nombre de lancers à partir de différentes initialisations et le nombre d'itérations moyen. L'algorithme proposé nécessite ainsi moins de calculs que l'algorithme de Fisher pour des tailles d'échantillon assez grandes.

3 Expérimentation sur des données simulées

Cette section évalue, sur des signaux simulés, les performances de l'algorithme proposé en termes de précision et de temps de calcul. Les résultats obtenus avec l'algorithme proposé sont comparés avec ceux donnés par l'algorithme de Fisher, qui fournit la solution optimale [4]. Chaque signal simulé est formé de $n = 1000$ paires (x_i, y_i) réparties en $K = 3$ classes. Pour simplifier, on considère un pas d'échantillonnage constant égal à 1 (c'est-à-dire que $x_i = i$), et une distribution de bruit identique pour chaque arc du signal.

Pour K vecteurs de coefficients polynômiaux $(\boldsymbol{w}_k)$ et des paramètres de la distribution du bruit fixés, la génération d'un signal s'effectue suivant le schéma suivant :

• Choix des limites j_1, j_2, j_3, j_4 des classes ; dans toutes les simulations, nous avons choisi $j_1 = 1$, $j_2 = 200$, $j_3 = 500$ et $j_4 = 1000$.

• Génération de chaque partie $k \in \{1, 2, 3\}$ du signal, suivant le modèle $y_i = \boldsymbol{w}_k^T\boldsymbol{z}_i + \varepsilon_i$ $(j_k \leq i < j_{k+1})$, où $\boldsymbol{z}_i = [1\ x_i\ x_i^2]^T$ et ε_i est un bruit généré suivant deux types de distributions : une distribution symétrique (distribution normale centrée d'écart-type σ) et une disribution non symétrique (distribution de Weibull centrée de paramètre de forme κ et de paramètre d'échelle λ).

Deux types de signaux (A et B), relatifs au choix des paramètres $\boldsymbol{w}_k$, ont été considérés. Pour chaque type de signal, deux distributions ont été testées avec trois différents écart-types ($\sigma = 1$, $\sigma = 4$, $\sigma = 7$) pour la distribution normale et les couples de paramètres $(\kappa = 1.5, \lambda = 1.6)$, $(\kappa = 1.5, \lambda = 6.5)$, $(\kappa = 1.5, \lambda = 11.5)$ pour la distribution de Weibull. La figure 2 montre des exemples de signaux de type A et B.

Le tableau 1 présente les pourcentages de bien classés entre les partitions fournies par l'algorithme proposé et par l'algorithme de Fisher, et la vraie partition simulée. Chaque pourcentage de bien classés représente une moyenne sur 25 signaux simulés différents. Le nombre d'initialisations différentes retenu dans l'algorithme proposé, qui a été jugé suffisant pour garantir l'obtention de minima locaux corrects, est de 15. Ces résultats montrent que l'algorithme proposé fournit de bons résultats, quasi-similaires à ceux de

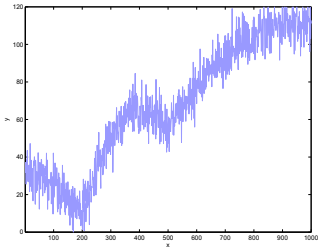
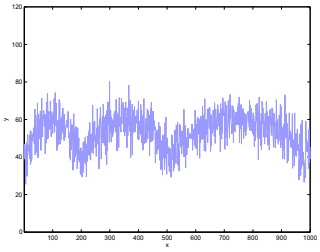



Figure 2: Exemples de signaux de type A (à gauche) et B (à droite) avec un bruit généré suivant une distribution normale d'écart-type $\sigma = 7$.

l'algorithme de Fisher. Le type de densité du bruit (symétrique ou non symétrique) n'a pas d'influence particulière sur la qualité des résultats. On peut également observer que les pourcentages de mal classés des deux méthodes croissent très légèrement en fonction de l'écart-type. Ce constat porte à croire que pour des valeurs d'écarts-type trop élevées du bruit, les deux algorithmes auraient du mal à distinguer les limites des classes.

Table 1: Pourcentages de mal classés obtenus avec l'algorithme proposé et l'algorithme de Fisher, pour les deux types de signaux.

						$\kappa = 1.5$	
		$\sigma = 1$	$\sigma = 4$	$\sigma = 7$	$\lambda = 1.6$	$\lambda = 6.5$	$\lambda = 11.5$
Type A	Algorithme de Fisher	0.45	1.21	1.84	0.38	1.20	1.88
	Algorithme proposé	0.45	1.17	1.85	0.38	1.24	1.92
Type B	Algorithme de Fisher	0.53	1.22	1.92	0.36	1.18	1.95
	Algorithme proposé	0.53	1.18	1.94	0.36	1.19	1.99

Pour ces mêmes simulations, nous avons obtenu des temps de calcul moyens, avec un ordinateur doté d'un Pentium Centrino de 1.5Ghz, d'environ 67 secondes pour l'algorithme de Fisher et 11 secondes pour le nouvel algorithme. Ces résultats viennent confirmer les complexités évoquées dans la section 2.

4 Conclusion

Cet article a proposé une nouvelle méthode de segmentation de signaux unidimensionnels en parties homogènes, par minimisation itérative d'un critère numérique, sous la contrainte que les classes soient ordonnées par rapport au temps. L'étude expérimentale sur des signaux synthétiques a mis en évidence de bonnes performances de l'algorithme proposé en termes de précision de partition et de temps de calcul, comparé à l'algorithme de Fisher. Une étude expérimentale complémentaire sur des signaux réels est la perspective directe de ce travail.

[1] P. Aknin, L. Oukhellou, F. Vilette, "Track circuit diagnosis by automatic analysis of inspection car measurement", *6th Word Congress on Railway Research*, Edinburgh, 2003.
[2] C. Charles, "Régression typologique et reconnaissance des formes", *Thèse de doctorat 3eme cycle* 1977.
[3] G. Ferrari-Trecate, M. Muselli, D. Liberati, M. Morari, "A clustering technique for the identification of piecewise affine and hybrid systems", *Automatica* 39, 2003, 205-217.
[4] W. D. Fisher, "On grouping for maximum homogeneity", *JASA* 53, 1958, 789-798.

Reconnaissance des dissimilarités de Robinson

M. Seston

Laboratoire d'Informatique Fondamentale
Faculté des Sciences de Luminy, Université de la Méditerranée,
13288 Marseille Cedex 9, France
morgan.seston@lif.univ-mrs.fr

Mots clés : Algorithme, Classification, Graphes

1 Introduction

En classification, il est connu qu'à certains types de dissimilarités sont associées des représentations graphiques. Parmi les plus connues, on peut citer les ultramétriques que l'on peut représenter sous forme d'arbres hiérarchiques [3]. Dans notre cas, nous nous intéresserons aux dissimilarités de Robinson introduites par Robinson [5], à l'origine pour des problèmes de chronologie en archéologie. Il a été par la suite montré que ces dissimilarités sont une généralisation des ultramétriques et peuvent être représentées par des pseudo-hiérarchies (ou pyramides) faiblement indicées. Mirkin et Rodin [4] donnent un algorithme en $\mathcal{O}(n^4)$ de reconnaissance des dissimilarités de Robinson, basé sur les algorithmes de reconnaissance des hypergraphes d'intervalles. Plus tard, Chepoi et Fichet [2] donneront un algorithme en $\mathcal{O}(n^3)$ pour le même problème en utilisant une stratégie "diviser pour régner". Ici, nous présenterons un nouvel algorithme pour le problème de reconnaissance avec une complexité en $\mathcal{O}(n^3)$ en temps et $\mathcal{O}(n^2)$ en espace comme dans [2]. L'intérêt de cet algorithme est qu'il n'utilise que des notions très connues de la théorie des graphes et qu'il est simple à implémenter.

2 Préliminaires

Soit d une dissimilarité définie sur l'ensemble X de cardinalité n. Un ordre total $\prec$ est dit *compatible* pour une dissimilarité d, si pour tout x, y, z tels que $x \prec y \prec z$, on a $d(x,z) \geq \max\{d(x,y), d(y,z)\}$ (si un ordre est compatible, son dual l'est aussi). On peut remarquer que si l'on construit, selon un ordre compatible, une matrice associée à la dissimilarité (notée D par la suite), alors les valeurs de celle-ci sont croissantes en lignes et en colonnes quand on s'éloigne de la diagonale principale. Une dissimilarité est dite de Robinson s'il existe un ordre compatible. Pour un ensemble $A \subseteq X$, on notera $\delta(A) = \max\{d(u,v) : u, v \in A\}$ le *diamètre* de A. Soit $\preceq$ un ordre partiel sur X. Pour deux ensembles A, B on notera $A \preceq B$ si et seulement si pour tout $a \in A, b \in B$, on a $a \preceq b$. Un ordre partiel $\preceq_2$ raffine un ordre partiel $\preceq_1$ si $x \preceq_1 y$ implique $x \preceq_2 y$. Remarquons qu'un ordre partiel $\preceq$ peut toujours être représenté sous la forme d'une partition ordonnée $(B_1, B_2, \ldots, B_m)$, telle que $B_i \preceq B_j$, pour tout $i < j$. En effet, la partition triviale en un seul bloc est toujours possible (pour cet algorithme, nos partitions seront composés de au moins deux blocs). Un sous-ensemble B_i de cette partition sera appelé *bloc*. Pour notre algorithme en plus de la matrice de dissimilarité D, on utilisera la matrice $D_{\leq}$ obtenue à partir de la matrice D en ordonnant ses lignes de manière non décroissante. La matrice $D_{\leq}$ peut être construite en $\mathcal{O}(n^2 log n)$.

3 L'Algorithme

Pour cet algorithme, nous allons utiliser une stratégie "diviser pour régner". Notre approche consiste à partitionner l'ensemble X, fixer un ordre entre les ensembles de la partition, puis raffiner cet ordre si nécessaire, et enfin appeler récursivement l'algorithme sur les sous-ensembles induits par le raffinement.

Soit une dissimilarité d définie sur l'ensemble X. Nous allons construire le graphe $\mathcal{G} = (X, E)$ où $E = \{xy : d(x, y) < \delta(X)\}$. Intéressons-nous d'abord au cas où $\mathcal{G}$ n'est pas connexe, soit $\mathcal{C}$ l'ensemble des composantes connexes de $\mathcal{G}$. Dans ce cas, pour tout x appartenant à une composante connexe C, et pour tout y, y' n'appartenant pas à C, on a $d(x, y) = d(x, y')$. Ceci implique que l'on peut ordonner de façon indépendante les différentes composantes connexes, et donc appeler récursivement notre algorithme sur chacune d'elles. Ainsi, on obtient un ordre total compatible pour chaque composante connexe (s'il existe), et il ne reste plus qu'à fixer un ordre arbitraire pour obtenir un ordre total compatible sur X. Si une composante connexe $C \in \mathcal{C}$ n'admet pas d'ordre compatible alors il n'existe pas d'ordre compatible pour d. En effet, C est un sous-ensemble de X. Maintenant, dans le cas où $\mathcal{G}$ est connexe. On choisit deux éléments distincts x et y dans X tels que $xy \notin E$. Par la suite, on notera respectivement $V_x = \{x' : xx' \in E\}$ et $V_y = \{y' : yy' \in E\}$ les voisinages de x et y dans $\mathcal{G}$. Dans ce graphe, on définit les ensembles (nécessairement non vides) suivants : Soit $C_{xy} = \{z \notin \{x, y\} : \exists P \in \mathcal{P}$ tel que $z \in P\}$, où $\mathcal{P}$ est l'ensemble des chemins entre x et y ne contenant qu'un seul élément de V_x et qu'un seul élément de V_y. Et soit C_x et C_y les composantes connexes (nécessairement distinctes) de $\mathcal{G} \setminus C_{xy}$ contenant respectivement x et y. On peut montrer que dans tout ordre compatible, C_{xy} est entre C_x et C_y, (i.e. $C_x \preceq C_{xy} \preceq C_y$ ou $C_y \preceq C_{xy} \preceq C_x$). Sans perte de généralité, on peut fixer $C_x \preceq C_{xy} \preceq C_y$. Maintenant, on peut utiliser la même procédure de raffinement que dans [4], [2]. Cette procédure est basée sur la propriété suivante : Soit un ordre partiel $(\preceq) = (B_1, B_2, \ldots, B_m)$, et $\prec$ un ordre total raffinant $\preceq$, compatible avec d, alors pour tout $x \in B_i$ et $y, z \in B_j$ tels que $d(x, y) < d(x, z)$, on a $y \preceq z$ si $j > i$ et $z \preceq y$ si $i > j$. Il a été montré dans [2] que pour tout bloc distinct B_i, B_j de l'ordre partiel obtenu par la procédure de raffinement et pour tout élément $x, y \in B_i$ et $z \in B_j$, on a $d(x, z) = d(y, z)$. Donc, comme dans le cas où $\mathcal{G}$ n'est pas connexe, on peut appeler récursivement notre algorithme sur chaque bloc de l'ordre $\preceq$, et ainsi obtenir un ordre total compatible avec d (s'il existe). Notons que cette procédure de construction d'un ordre partiel est basée sur des conditions nécessaires donc si à un instant l'ordre retourné n'est pas compatible alors d n'est pas une dissimilarité de Robinson.

Robinson

Input : Un ensemble fini X et une dissimilarité d définie sur X

Output : Un ordre total $\prec$ compatible avec X si il existe

if $|X| = 1$
  return $(\prec) := \{X\}$
else
  Construire $\mathcal{G} = (X, E)$
  if $\mathcal{G}$ n'est pas connexe
    Soit $C_1, C_2, \ldots, C_p$ les composantes connexes de $\mathcal{C}$
    return $(\prec) := (robinson(C_1), robinson(C_2), \ldots, robinson(C_p))$
  else
    Soit x, y deux éléments tels que $xy \notin E$
    Construire C_{xy}, C_x, C_y
    Construire $(\prec) := (C_x, C_{xy}, C_y)$
    $(\prec) :=$ refine$(X, (C_x, C_{xy}, C_y))$
    Soit $(\prec) = (B_1, B_2, \ldots, B_m)$
    $(\prec) := (robinson(B_1), robinson(B_2), \ldots, robinson(B_m))$
    if $\prec$ est compatible avec d
      return $\prec$
    else
      return d n'est pas Robinsonienne

Refine

Input : Un ordre partiel $\preceq = (B_1, B_2, \ldots, B_m)$ et une dissimilarité d définie sur X

Output : Un ordre partiel

$L := \emptyset$;
if $m = 1$
  return $\prec$
else
  $i := 1$
  until $i = m$
    for any $b \in B_i$
      $(\prec) := refine(X, \prec, b)$
      Soit $(\prec) = (B_{(1,1)}, \ldots, B_{(1,k_1)}, \ldots, B_{(i-1,k_{i-1})}, B_i, B_{(i+1,1)}, \ldots, B_{(m,k_m)})$
      $m := \sum_{j=1}^{i-1} k_j + \sum_{j=i+1}^{m} k_j + 1$
      $i := \sum_{j=1}^{i-1} k_j + 1$
  return $(\prec) := (refine((B_{(1,1)}, \ldots, B_{(1,k_1)}), R), \ldots, refine((B_{(m,1)}, \ldots, B_{(m,k_m)}), R))$

Maintenant, nous allons rapidement nous intéresser à la complexité de l'algorithme. Pour la complexité en espace, on doit stocker la matrice de dissimilarité, la matrice triée $D_{\leq}$ et le graphe $\mathcal{G}$, chacun de taille $\mathcal{O}(n^2)$. En ce qui concerne la complexité en temps, nous allons nous intéresser à la complexité d'un appel récursif. Construire $\mathcal{G}$ et ses composantes a une complexité en $\mathcal{O}(n^2)$. Si $\mathcal{G}$ est connexe, il nous faut construire C_{xy}. Pour cela, on construit le graphe $G_{xy} = \{X', E'\}$ où $X' = X \setminus \{x, y\}$, $E' = E \setminus \{uv : u, v \in V_x$ ou $u, v \in V_y\}$. On peut montrer que les composantes connexes de G_{xy} contenant au moins un élément de V_x et au moins un élément de V_y correspondent à C_{xy}. Donc la construction de C_{xy} peut se faire en $\mathcal{O}(n^2)$. De même, les ensembles C_x et C_y correspondent aux composantes connexes de $\mathcal{G} \setminus C_{xy}$, et peuvent être construits en $\mathcal{O}(n^2)$. Enfin, dans [2], il a été montré que la procédure de raffinement peut être faite en $\mathcal{O}(n^2)$ en utilisant $D_{\leq}$ et le tri par casiers (bucket sort) [1]. Comme on a au plus $\mathcal{O}(n)$ appels récursifs, la complexité générale de l'algorithme est donc en $\mathcal{O}(n^3)$.

Références

[1] A. V. Aho, J. E. Hopcroft et J. D. Ullman, "The Design and analysis of Computer Algorithms", *Reading, MA :Addison-Wesley*, 1974.

[2] V. Chepoi et B. Fichet, "Recognition of Robinsonian dissimilarities", *Journal of Classification* 14, 1997, 311–325.

[3] N. Jardine et R. Sibson, "Mathematical Taxonomy", *John Wiley & Sons*, 1971.

[4] B. Mirkin et S. Rodin, "Graphs and Genes", *Springer-Verlag, Berlin*, 1984.

[5] W. S. Robinson, "A method for chronologically ordering archaeological deposits", *American Antiquity* 16, 1951, 293–301.

Partitionnement par colonies de fourmis et essaims particulaires

J. Trejos[1], E. Piza[1], A. Murillo[2] et M. Villalobos[1]

1. CIMPA, Université du Costa Rica, San José, CostaRica.
2. Sede del Atlántico, Université du Costa Rica, Turrialba, Costa Rica.
(jtrejos, epiza, murillof, mvillalo)@cariari.ucr.ac.cr

Mots clés : Classification automatique, Optimisation en classification, Approches inspirées du vivant.

Résumé

On propose l'application des heuristiques connues sous le nom d'optimisation par colonies de fourmis et par essaims de particules, dans le problème de partitionnement en classification automatique. Le premier algorithme utilise le principe d'associer une partition à chaque fourmi, et cette partition est modifiée à chaque itération par la sélection aléatoire des éléments qui sont affectés aux classes. Cette sélection aléatoire depend du trait de phéromone —lié à la qualité mesurée par l'inertie inter-classes— et d'une heuristique locale définie à partir des distances originales. Le deuxième algorithme est basé par le mouvement dans l'espace multidimmensionelle des centres des classes selon les règles usualles de cette heuristique, et l'allocation des individus au centre le plus proche. On montrera quelques résultats comparatifs avec d'autres heuristiques d'optimisation que nous avons déjà utilisées, comme le recuit simulé, la recherche tabou, et les algorithmes génétiques, sur de données simulées.

1 Introduction

Le partitionnement est un problème de classification dont les méthodes trouvent souvent des optima locaux des critères à optimiser [3]. C'est pour cela que l'implémentation des heuristiques modernes d'optimisation combinatoire —telles que le recuit simulé, la recherche tabou et les algorithmes génétiques [1]— peut être intéressante. Nous l'avons fait avec succès en partitionnement numérique [5], binaire et croisé, ainsi qu'en analyse des proximités, régression non linéaire, rotations varimax obliques et ensembles *rough*, entre autres.

Dans ce travail on étudie deux nouvelles heuristiques qui ont montré des bonnes performances dans plusieurs problèmes d'optimisation, appellées *optimisation par colonies de fourmis* (*ant colony optimization* ou ACO, en Anglais) [2] et *optimisation par essaims particulaires* (*particle swarm optimization* ou PSO, en Anglais) [4].

Nous sommes en présence d'un ensemble d'objets $\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ dans $\mathbb{R}^p$, et on cherche une partition $P = (C_1, \ldots, C_K)$ de Ω, le nombre K de classes fixé à l'avance, qui minimise l'inertie intra-classes $W(P)$ ou, ce qui est équivalent, maximise l'inertie inter-classes $B(P)$.

Dans ce qui suit, on étudie l'application de l'Optimisation par Colonies de Fourmis (ACO, dans le paragraphe 2) de même que les Essaims de Particules (PSO, dans le paragraphe 3), puis on montre quelques résultats comparatifs dans le paragraphe 4.

2 Application de l'ACO en partitionnement

L'optimisation par colonies de fourmis est inspirée par la façon dont les fourmis cherchent leur nourriture. On peut trouver les détails de cette heuristique dans [2]. On propose un algorithme itératif tel qu'à chaque itération on examine toutes les fourmis. Au début, une fourmi m est associée à une partition P^m générée au hasard, on applique nuées dynamiques et on converge à un minimum local de W. Pendant les itérations, la fourmi modifiera P^m comme suit: un objet i est choisi au hasard, et un autre objet j est sélectionné au hasard selon une roulette aléatoire avec probabilité p_{ij}, où p_{ij} dépend du trait de phéromone et de l'heuristique locale. On peut dire que la fourmi decide si sera j affecté à la même classe que i.

Si on note t l'itération en cours, la partition associée à la fourmi m pendant l'itération t sera notée $P^m(t)$. La valeur du trait de phéromone est modifiée selon la règle $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho \sum_{m=1}^{M} \Delta^m\tau_{ij}(t+1)$, où τ_{ij} associe deux objecs i, j de Ω, et $\rho \in]0,1]$ est un *coefficient d'évaporation*. $\Delta^m\tau_{ij}(t+1)$ est la quantité de phéromone par agent dans l'association d'objets i, j dans la même classe, définie par $B(P^m(t))/I$ si i et j appartiennent à la même classe de $P^m(t)$, et 0 sinon, $B(P^m(t))$ étant l'inertie inter-classes de la partition $P^m(t)$. Deux objets classifiés dans la même classe laissent donc un trait de phéromone. L'heuristique locale ou visibilité à court terme est définie par $\eta_{ij} = 1/\|\mathbf{x}_i - \mathbf{x}_j\|$, de façon à ce que deux individus proches ont une influence en la probabilité de les affecter dans la même classe. Soit α et β deux paramètres réels positifs. Si une fourmi m est placée sur un objet i, l'objet j est choisi avec probabilité $p_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha[\eta_{ij}]^\beta}{\sum_{l=1}^{n}[\tau_{il}(t)]^\alpha[\eta_{il}]^\beta}$ et on introduit j dans la même classe que i. Ce choix aléatoire est semblable à ce qu'on appelle la roulette aléatoire dans les algorithmes génétiques: les lignes de la matrice $(p_{ij}(t))_{n\times n}$ somment 1; étant donné i, la valeur $p_{ij}(t)$ est la probabilité de choisir j, qui est modélée en utilisant la probabilité cumulée et en générant des nombres pseudo-aléatoires.

L'algorithme a les paramètres suivants: le nombre de fourmis M, la valeur initiale de phéromone τ_0, le nombre maximal d'itérations $t_{\max}$, les puissances α de la phéromone et β de l'heuristique locale, le coefficient d'évaporation ρ, et le nombre S de transferts pour chaque fourmi.

En considérant les éléments ci-dessus, l'algorithme est comme suit:

Algorithme AcoClus
Initialiser $\tau_{ij} = \tau_0$; calculer η
Initialiser les probabilités p
Initialiser au hasard les partitions P^1,...,P^M associées à chaque fourmi
Par nuées dynamiques sur chaque P^m converger vers un minimum local de W
pour $t = 1$ jusqu'à $t_{\max}$ faire:
  pour $m = 1$ jusqu'à M, faire S fois:
    choisir au hasard un objet i
    choisir un objet j avec probabilité $p_{ij}(t)$
    affecter j à la classe de i
  calculer $B(P^1), \ldots, B(P^M)$ et garder la meilleure valeure
  mettre à jour $\tau(t)$ et $p(t)$

3 Application de la PSO en partitionnement

On utilise le principe de la PSO sur des particules définies par un ensemble de K centres de classes. En effet, chaque particule est un point de $\mathbb{R}^{pK}$ et la partition associée est

définie par l'allocation des individus de Ω au centre de classe le plus proche. On utilise une partition de référence: la meilleure trouvée par l'algorithme et dont les centres sont notés $(\mathbf{g}_1^*, \ldots, \mathbf{g}_K^*)$, et chaque particule m a en mémoire sa meilleure position dans sa trajectoire, notée $(\mathbf{g}_1^{m*}, \ldots, \mathbf{g}_K^{m*})$.

L'algorithme possède les paramètres suivants: le coefficient de vitesse α, $r_1 + r_2 > 4$ (voir [4]), le nombre maximal d'itérations *maxiter*, la borne $V_{\max}$, et M, la taille de la population.

Algorithme PsoClus
Initialiser M, $V_{\max}$, α, *maxiter* et les partitions P^1,...,P^M:
Calculer les centres des classes $\mathbf{g}_1^1, \ldots, \mathbf{g}_K^1, \ldots, \mathbf{g}_1^M, \ldots, \mathbf{g}_K^M$
Calculer la meilleure valeur de chaque particule et le leader global,
Répéter pour $t = 1, 2, \ldots$ jusqu'à convergence ou *maxiter* fois:
  pour $m = 1$ jusqu'à M faire:
    soit $r_1 := random(0, 4)$, $r_2 := random(0, 4)$, vérifier que $r_1 + r_2 > 4$
    pour $k = 1$ jusqu'à K faire:
      pour $j = 1$ jusqu'à p faire:
        soit $v_{kj}^m(t) := \alpha v_{kj}^m(t-1) + r_1(g_{kj}^{m*} - g_{kj}^m(t-1)) + r_2(g_{kj}^* - g_{kj}^m(t-1))$
        si $v_{kj}^m(t) > V_{\max}$ alors $v_{kj}^m(t) := V_{\max}$
          sinon, si $v_{kj}^m(t) < -V_{\max}$ alors $v_{kj}^m(t) := -V_{\max}$
        soit $g_{kj}^m(t) := g_{kj}^m(t-1) + v_{kj}^m(t-1)$
  affecter tous les n objets de Ω au centre $\mathbf{g}_k^m$ le plus proche
  mettre à jour le vecteur $(\mathbf{g}_1^{m*}, \ldots, \mathbf{g}_K^{m*})$
mettre à jour $(\mathbf{g}_1^*, \ldots, \mathbf{g}_K^*)$

4 Resultats et perspectives

De même que pour les autres heuristiques, l'optimisation par colonies de fourmis a un problème pour fixer les valeurs des paramètres. Une première expérimentation nous a donné donc une meilleure idée sur le choix des paramètres. En particulier, on a noté une amélioration significative quand on utilise des solutions de nuées dynamiques avant de tourner le méthode des fourmis, au lieu de tourner celle-ci sur des partitions purement aléatoires. On peut donc considérer AcoClus comme une méthode d'amélioration des nuées dynamiques.

Afin de comparer les méthodes, nous avons généré aléatoirement des tableaux à 6 variables gaussiennes, en controllant quatre facteurs: la taille n du tableau de données (105 ou 525), le nombre K de classes (3 ou 7), le cardinal des classes (égales ou différentes) et la variance σ^2 des classes (égales ou différentes). La Tableau 1 montre les résultats d'appliquer les méthodes basées en heuristiques, AcoClus et PsoClus ainsi que le recuit simulé, la recherche tabou et l'algorithme génétique [5], sur les 16 tableaux de données générés. Seulement la meilleure valeur de W trouvée par n'importe quelle méthode est montrée, ainsi que le pourcentage de fois que cette valeur a été trouvée par la méthode indiquée en colonne. On montre aussi les résultats obtenus avec les nuées dynamiques pour la cas du centre de gravité, et nous indiquons si la classification hiérarchique suivant le critère d'agrégation de Ward (arbre coupé au niveau de classe respectif) trouve cette meilleure partition. Les résultats montrés ci-dessous ont été obtenus par l'application d'AcoClus avec une population de taille $M = 20$, un facteur de transferts $S = 2 \times n$, $\alpha = 1$, $\beta = 0.8$, $\rho = 0.3$, $\tau_0 = 0.001$, et un nombre maximum d'itérations de 20. Pour PsoClus, on a utilisé $M = 20$, $\alpha = 0.9$ et *maxiter* $= 100$. Toutes les heuristiques ont

été appliquées 100 fois, sauf la méthode hiérarchique de Ward [3] qui a été appliquée une seule fois car elle est déterministe.

Tableau 1: Meilleure valeur de l'inertie intra–classs W^* et fréquence de cette veleur, pour chaque méthode appliqué 100 fois: recuit simulé (RS), recherche tabou (RT), algorithme génétique (AG), colonies de fourmis (ACO), essaims de particules (PSO), nuées dynamiques (ND), et classification hiérarchique de Ward.

Facteurs				Critère	Méthodes						
n	K	σ^2	Card	W^*	RS	RT	AG	ACO	PSO	ND	Ward
105	3	=	=	5,42	100	99	100	100	100	91	oui
105	7	=	=	5,15	100	74	82	100	1	19	oui
525	3	=	=	5,99	100	100	100	100	94	98	oui
525	7	=	=	5,34	100	82	88	100	1	45	oui
105	3	$\neq$	=	13,15	100	99	100	100	1	13	non
105	7	$\neq$	=	9,90	100	51	69	75	0	1	non
525	3	$\neq$	=	15,81	100	51	82	99	1	2	non
525	7	$\neq$	=	8,26	100	100	94	100	0	53	non
105	3	=	$\neq$	5,01	100	100	100	100	99	91	oui
105	7	=	$\neq$	5,55	0	0	35	36	1	3	oui
525	3	=	$\neq$	5,67	8	100	100	100	84	95	oui
525	7	=	$\neq$	5,65	0	0	22	38	1	2	oui
105	3	$\neq$	$\neq$	11,73	100	100	100	100	12	95	non
105	7	$\neq$	$\neq$	7,63	0	0	37	85	0	6	non
525	3	$\neq$	$\neq$	13,82	3	100	100	100	1	59	non
525	7	$\neq$	$\neq$	7,46	0	0	21	54	0	0	non

Les résultats dans le tableau 1 montrent que AcoClus trouve de bons résults, supérieurs à ceux des autres heuristiques, et que PsoClus a des problèmes si les variances des classes sont différentes.

[1] I. Charon, A. Germa, O. Hudry, *Méthodes d'Optimisation Combinatoire*. Masson, Paris, 1996

[2] E. Bonabeau, M. Dorigo, G. Therauluz, *Swarm Intelligence. From Natural to Artificial Systems*. Oxford University Press, New York, 1999.

[3] E. Diday, J. Lemaire, J. Pouget, F. Testu, *Eléments d'Analyse des Données*. Dunod, Paris, 1982.

[4] J. Kennedy, R.C. Eberhart, *Intelligent Swarm Systems*. Academic Press, New York, 2000.

[5] J. Trejos, A. Murillo, E. Piza, "Classification tabou basée en transferts", "Partitionnement par recuit simulé", et "Un algorithme génétique de partitionnement", in: *IV Journées de la Soc. Franc. de Classif.*, S. Joly & G. Le Calvé (eds.), Vannes, 1996.

Une approche basée sur les treillis de Galois pour la construction des réseaux de neurones

N. Tsopzé[1,2], E. Mephu Nguifo[1] et G. Tindo[2]

1. CRIL - IUT de Lens, SP 16 Rue de l'Université, 62307 Lens Cedex
2. Département d'informatique - Université de Yaoundé I, BP 812 Yaoundé
tsopze.norbert@gmail.com,mephu@cril.univ-artois.fr,gtindo@uycdc.uninet.cm

Mots clés : Classification automatique, Réseau de neurones, Apprentissage.

L'utilisation des réseaux de neurones dans un système d'apprentissage suscite beaucoup de questions: *quel est le modèle approprié? Quel est le nombre de couches? Quel est le nombre de neurones par couche? Comment connecter les neurones entre eux? etc.* Les différentes réponses à ces questions constituent l'architecture (ou topologie) du réseau de neurones. Le problème majeur du réseau de neurones (et qui reste ouvert) est la définition de sa structure [2, 5]; en clair, il n'existe aucune démarche méthodique précise permettant de trouver l'architecture d'un réseau de neurones pour la résolution d'un problème donné. En pratique, la philosophie la plus courante pour construire l'architecture d'un réseau de neurones multicouches utilise la méthode ad-hoc qui consiste à définir une couche d'entrée avec un nombre de neurones égal au nombre d'attributs (variables), une couche de sortie ayant autant de neurones que le nombre de classes et une couche cachée avec un nombre de neurones égal à la moyenne du nombre de neurones en entrée et du nombre de neurones en sortie. Dans la littérature, deux approches existent pour construire automatiquement l'architecture du réseau:

1. Utiliser un ensemble de règles décrivant le domaine du problème à résoudre [11]; déduire de ces règles l'architecture du réseau pouvant résoudre le problème. L'avantage de cette approche est la possibilité d'interprétation des réseaux de neurones car chaque nœud représente une variable et une connexion entre deux nœuds représente une connaissance. Dans le cas des problèmes sans connaissances à priori, il est difficile de recourir à cette approche.

2. Construire à partir des exemples un réseau capable de bien classer ces données [8, 9]. Cette dernière méthode est utilisée sans aucune connaissance du domaine; une comparaison des systèmes dérivés est présentée dans [12]. Cette méthode a comme avantage la recherche d'une architecture optimale permettant de minimiser le nombre de neurones et de couches cachées; elle apporte une réponse constructive au cas des problèmes sans connaissances à priori. Elle est limitée par le problème d'interprétation de l'architecture du réseau; le réseau construit est toujours vu comme une boite noire.

Pour pallier à certaines limitations des approches citées ci-dessus, nous proposons une approche basée sur les treillis de Galois [4] pour la construction des architectures des réseaux de neurones. En effet le graphe de Hasse représentant graphiquement la relation de spécialisation/généralisation entre concepts est une structure de graphe pour laquelle les interprétations de ses sommets et de ses connexions ont été bien étudiées. Les treillis de Galois ont été largement utilisés en classification supervisée [7].

Nous utilisons dans ce travail le graphe (de Hasse) pour définir la structure du réseau de neurones qui permettrait de classer les données après l'apprentissage. Les intérêts de cette approche sont: (1) en absence de connaissances de domaine, la capacité de construire un réseau en utilisant implicitement (sans les extraire) les règles présentes dans le treillis; (2) la possibilité d'interpréter le réseau en utilisant les connaissances contenues dans le treillis; (3) la possibilité de justifier la présence des différentes unités et connexions du réseau.

Réseau de neurones

Un réseau de neurones est formellement représenté par un ensemble de poids et seuil permettant de calculer la propagation de l'information entre différentes unités (ou entre différentes couches). Il fonctionne généralement en deux phases:

1. *Phase d'apprentissage.* Elle consiste à construire à partir des données un modèle capable de produire des bonnes décisions sur ces données. Cette phase intègre la définition de l'architecture du réseau et la recherche des meilleurs poids de connexion entre différents neurones.

2. *Phase de classement.* Au cours de cette phase, le système construit à la phase d'apprentissage est utilisé pour produire des décisions sur des objets qui ne faisaient pas partie des données d'apprentissage. Généralement, ces décisions ne peuvent être justifiées que si l'architecture du réseau le permet.

Treillis de Galois

Le treillis de Galois se définit par rapport à un contexte et aux relations entre éléments du contexte. Un contexte C est un triplet (O, A, I) où A et O sont des ensembles finis (souvent appelés respectivement ensemble d'attributs et ensemble d'objets) et $I \subseteq O \times A$.

Une correspondance de Galois entre deux ensembles ordonnés E_1 et E_2 est un couple d'applications (f, g) telles que f (resp. g) soit une application monotone décroissante définie de E_1 vers E_2 (resp. E_2 vers E_1). Soient $h = g \circ f$ et $h' = f \circ g$ deux applications monotones croissantes, extensives et idempotentes, h et h' sont des opérateurs de fermeture sur les ensembles de parties $P(O)$ et $P(A)$.

Soient $O_1 \subseteq O$ et $A_1 \subseteq A$, la paire (O_1, A_1) est un concept formel si $f(O_1) = A_1 = h'(A_1)$ et $g(A_1) = O_1 = h(O_1)$. A_1 (resp. O_1) est l'intension (resp. l'extension) du concept.

Soient (O_1, A_1) et (O_2, A_2) deux concepts, et $\leq$ une relation sur l'ensemble des concepts définie par: $(O_1, A_1) \leq (O_2, A_2) \iff O_1 \subseteq O_2$ $(A_2 \subseteq A_1)$; on dit que le concept (O_1, A_1) est un successeur du concept (O_2, A_2) et (O_2, A_2) est un prédécesseur du concept (O_1, A_1). Le graphe de Hasse est une représentation graphique de la relation de couverture (sans intermédiaire) entre des concpets.

Le treillis de Galois est construit par une version modifiée de l'algorithme de Bordat [1]. L'approche utilisée est descendante (pour une spécialisation successive). La modification apportée consiste à selectionner au cours de la construction du treillis les concepts qui verifient l'heuristique (contrainte) appliquée. De nombreuses mesures (ou contraintes) permettant de selectionner les concepts sont présentées dans la littérature [6, 7]. Ces contraintes ont aussi pour effet de limiter le temps de génération du pseudo-treillis ainsi que sa taille. Parmi ces mesures, nous ponvons citer:

Fréquence.

Le support s d'un concept (O_1, A_1) est le rapport entre la cardinalité de son extension et la cardinalité de l'ensemble des objets ($s = \frac{100 \times |O_1|}{|O|}\%$). δ souvent appelé "*minsupp*" est une constante spécifiée par l'utilisateur. Formellement le concept (O_1, A_1) est fréquent si

$|O_1| \geq \delta$. La fréquence est une contrainte anti-monotone qui va permettre d'élaguer le treillis et de réduire la complexité de construction. Le treillis est construit uniquement à partir des exemples.

Validité du concept.
L'ensemble des données d'apprentissage O est divisé en deux sous ensembles (exemples (O^+) et contreexemples (O^-)). Les contraintes suivantes [6] y sont appliquées pour sélectionner les concepts à utiliser lors du classement:

- Validité. Un concept (O_1, A_1) est valide si A_1 est vérifie par au moins un certain nombre d'objets, cette notion est équivalente à la notion de fréquence avec le contexte restreint aux exemples. Formellement (O_1, A_1) est valide si $|O_1| \geq \alpha$ avec $0 < \alpha \leq |O^+|$. α peut être vu comme la fréquence (voir sous section précédente) de le motif dans le contexte réduit aux exemples positifs.

- Quasi-consistence. Un concept (O_1, A_1) est quasi-consistent s'il est valide et que sa description reconnaît peu de contreexemples ($\left|O_1^+\right| \geq \alpha$ et $\left|O_1^-\right| \leq \beta$).

La profondeur du treillis
Cette contrainte est facilement applicable lorsque les concepts sont générés par niveau. L'utilisateur peut l'utiliser pour limiter le nombre de couches du réseau. Pour un réseau de k couches, il suffit de contruire un pseudo-treillis de profondeur k.

L'approche

Le passage du treillis de Galois au réseau de neurones suit le processus suivant:

1. Le concept supremum (borne supérieure) du treillis devient la sortie du réseau de neurones.

2. Tous les autres concepts forment les neurones des couches cachées. La partie cachée du système peut être constituée d'une ou plusieurs couches définies à la construction du treillis.

3. Une nouvelle couche est créée; cette couche a autant de neurones que le nombre d'attributs caractérisant chaque exemple; cette nouvelle couche est la couche d'entrée du réseau de neurones.

4. Les connexions entre neurones n_1 et n_2 obtenus en réécrivant les concepts c_1 et c_2 sont celles directement du graphe de Hasse (la relation d'ordre) entre c_1 et c_2. La couche d'entrée est complétement connectée aux unités de la couche cachée n'ayant pas de successeurs.

Le réseau ainsi obtenu est appris par rétropropagation [10].

Expériences

Les expérimentations de cette technique ont été faites en utilisant les données SPECT (23 attributs, 80 exemples pour l'apprentissage et 187 exemples pour le test) et CHESS (36 attributs et 3196 exemples) de la base UCI. Les taux de généralisation obtenus de cette méthode varient de 80% à 93,5% pour les données SPECT et en moyenne 93% sur les données CHESS (résultat obtenu par validation croisée d'ordre 10). Un autre résultat très encourageant de cette méthode est la construction d'un réseau en l'absence de règles avec la possibilité de justifier les décisions que ce dernier peut produire; contrairement aux autres méthodes neuronales où le réseau est présenté comme une boite noire.

Conclusion

Une nouvelle approche (basée sur les treillis de concepts) de recherche des topologies de réseau de neurones a été présentée dans ce travail. En plus de la structure interprétable du réseau (obtenu en absence des connaissances du domaine), les résultats expérimentaux sur les données de test utilisées sont assez encourageants. Mais l'approche que nous avons ainsi présentée est limitée à une classification des données binaires à deux classes. Une extension de ce modèle à un système multiclasse et pouvant aussi traiter des données multivaluées sera étudiée dans le futur.

[1] J. Bordat, "Calcul pratique du treillis de Galois dune correspondance" *Mathmatiques, Informatiques et Sciences Humaines* 24: 31-47, 1986.

[2] A. Cornuéjols, L. Miclet, "Apprentissage Artificiel : Concepts et algorithmes", Eyrolles, 2002.

[3] G. Dreyfus, M. Samuelides, J.M. Martinez, M. Gordon, F. Badran, S. Thiria, L. Hérault, "Réseaux de Neurones : Méthodologie et applications" Eyrolles, 2002.

[4] B. Ganter, R. Wille, "Formal Concepts Analysis: Mathematical Foundations", Springer - Verlag,1999.

[5] J. Han, M. Hamber, "Datamining: Concepts and Techniques", Morgan Kauffman Publishers, 2001.

[6] E. Mephu Nguifo, "Une nouvelle approche basée sur les treillis de Galois pour l'apprentissage de concepts, *Mathématiques et sciences humaines*, 123: 19-38, 1993.

[7] E. Mephu Nguifo, P. Njiwoua, "Treillis de concepts et classification supervisée", *Revue Technique et Sciences Informatiques* 24: 449-488, 2005.

[8] R. Parekh, J. Yang, V. Honavar, "Constructive Neural Network learning Algorithm for Multi-Category Real Valued Pattern Classification", *Department of Computer Science Lowa State University,Tech Report TR 96-14*, 1997.

[9] R. Parekh, J. Yang, V. Honavar, "Constructive Neural-Network Learning Algorithms for Pattern Classification, *IEEE Transactions on Neural Networks* 11: 436-451, 2000.

[10] D.E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagating errors",*Nature* 323: 318-362, 1986.

[11] J.W. Shavlik,G. G. Towell, "Kbann: Knowledge based articial neural networks",*Artificial Intelligence* 70: 119-165, 1994.

[12] N. Tsopze, E. Mephu Nguifo, G. Tindo, "Etude des algorithmes de construction des réseaux de neurones", *EGC'2007*, 9-20, 2007.

INDEX PAR AUTEURS

Afonso Filipe 16, 54
Aknin Patrice 174
Aubry Marc 113
Azzag Hanane 117
Bahri Emna 20
Barthélémy Jean-Pierre 121, 149
Beninel Farid 24
Bertrand Frédéric 27
Bertrand Patrice 1, 74
Boc Alix 90, 161
Bock Hans 3
Boubou Mounzer 31
Bounekkar Ahmed 31
Brás Silva Helena 35
Briand Bénédicte 39
Brito Paula 35
Campedel Marine 43
Chavent Marie 47, 51
Clérot Fabrice 70
Cleuziou Guillaume 58
Conruyt Noël 62
Cozza Valentina 66
Csernel Baptiste 70
Csernel Marc 74
De Carvalho F. A. T. 78, 79
De Falguerolles Antoine 83
De Tayrac Marie 113
Denoeud Lucile 87
Dessertaine Alain 153
Diallo Alpha Boubacar 90
Diallo Abdoulaye Banire 161
Diatta Jean 94, 101
Diday Edwin 54, 153
Dossou-Gbété Simplice 97
Ducharme Gilles 39
Eliezer Estelle-Sarah 153
Feno Daniel 101
Ferrandiz Sylvain 105
Fertil Bernard 143
Fety Luc 165
Gama Joao 10
Govaert Gérard 174
Grosser David 62
Guarracino Mario Rosario 66
Guinot Christiane 117
Hardy André 109
Hébrail Georges 70
Hérault Jeanny 143
Huchard M. 169
Hurault-Plantet Martine 54
Husson François 113, 133
Jollois François-Xavier 147
Josse Julie 133
Kasoro Nathanaël 109
Kuentz Vanessa 47
Kuntz Pascale 11
Kyrgyzov Yvan 43
Lamure Michel 31
Lavergne Julien 117
Le Pouliquen Marc 121
Lê Sébastien 113, 133
Lebart Ludovic 12
Lechevallier Yves 79
Leclerc Bruno 137
Lerman Israël César 139
Lespinats Sylvain 143
Loménie Nicolas 147
Longree Dominique 149
Luong Xuan 149
Maddouri Mondher 20
Maître Henri 43
Makarenkov Vladimir 90, 161
Maumy Myriam 27
Mellet Sylvie 149
Mephu Nguifo Engelbert 186
Mercat-Rommens Catherine 39
Merroun Omar 153
Mosser Jean 113
Müller Nicolas S. 157
Murillo Alex 182
Napoli A. 169
Nguyen Dung 161
Nicoloyannis Nicolas 20
Nugier Sylvaine 54
Paul Nicolas 165
Peradotto Anne 54
Pinto da Costa Joaquim 35
Piza Eduardo 182
Puech Nicolas 87
Quatrain Yasmina 54
Rahal Mohamed Cherif 54
Ralambondrainy Henri 62, 94
Rasson Jean-Paul 78
Rigaux Philippe 153
Ritschard Gilbert 157
Rouanne M.H. 169
Samé Allou 174
Saracco Jérôme 47, 51
Seston Morgan 178
Studer Matthias 157
Terre Michel 165
Tindo Gilbert 186
Totohasina André 94, 101
Touati Myriam 54
Trejos Javier 182
Tsopze Norbert 186
Valtchev P. 169
Van Helden Jacques 14
Venturini Gilles 117
Verde Rosanna 66
Villalobos Mario 182

Société Francophone de Classification
http://sfc.enst-bretagne.fr/

Telecom Paris
http://www.telecom-paris.fr/

Coheris Spad

http://www.spad.eu/

École nationale supérieure des télécommunications
Groupe des Ecoles des Télécommunications - membre de ParisTech
46, rue Barrault - 75634 Paris Cedex 13
Tél. + 33 (0) 1 45 81 77 77 - Fax + 33 (0) 1 45 89 79 06 - www.enst.fr