Atelier CluCo :

Clustering et Co-clustering

Organisateurs :

Vincent Lemaire (Orange Labs),
Jean-Charles Lamirel (Loria),
Pascal Cuxac (CNRS-INIST)

Organisé conjointement à la conférence EGC
(Extraction et Gestion des Connaissances)
le 28 janvier 2014 à Rennes

Editeurs :

Vincent Lemaire - Orange Labs
2 avenue Pierre Marzin, 2300 Lannion
Email : vincent.lemaire@orange.com

Pascal Cuxac - INIST - CNRS
2 allée du Parc de Brabois, CS 10310, 54519 Vandoeuvre les Nancy Cedex
Email : pascal.cuxac@inist.fr

Jean-Charles Lamirel - LORIA - SYNALP Research Team
Campus Scientifique, BP. 239, 54506 Vandoeuvre les Nancy Cedex
Email : jean-charles.lamirel@loria.fr

———

Publisher:

Vincent Lemaire, Pascal Cuxac, Jean-Charles Lamirel
2 avenue Pierre Marzin
22300 Lannion

Lannion, France, 2014
ISBN : 978-2-7466-6842-3

PRÉFACE

La classification non supervisée ou clustering est de nos jours largement utilisée dans un nombre croissant d'applications dans des domaines aussi divers que l'analyse d'image, la reconnaissance de formes , la fouille de textes, la gestion de la relation client, la veille scientifique ou technologique, la bio-informatique, la recherche d'information, l'analyse de réseaux sociaux...

Bien que le clustering forme un domaine de recherche en soi, avec une longue histoire, et d'innombrables méthodes, de nombreuses questions se posent toujours, telles que par exemple:

- quels sont les bons paramètres : nombre de classes versus finesse d'analyse ?
- comment estimer la qualité d'un clustering, l'importance des variables explicatives ?
- les classes doivent-elles être strictes, floues, ou recouvrantes ?
- comment rendre un clustering robuste et résistant au bruit ?
- comment évaluer l'évolution temporelle du déploiement d'un clustering ?
- ...

L'objectif de cet atelier est de favoriser des présentations et des discussions plutôt que de se focaliser sur des articles écrits complets. La soumission de prises de position bien articulées, d'expériences industrielles et de travaux en cours sont les bienvenus et privilégiés.

Le but est le partage d'expérience et de savoir sur les problématiques liées au clustering (coclustering). Le but est aussi de vous (industriels et/ou universitaires) permettre de présenter des problèmes non résolus avec les méthodes de l'état de l'art et/ou les logiciels sur étagères.

V. Lemaire
Orange Labs

J.-Ch. Lamirel
Loria

P. Cuxac
CNRS-INIST

Membres du comité de lecture

Le Comité de Lecture est constitué de:

Gilles Bisson (LIG)
Alexis Bondu (EDF RD)
Marc Boullé (Orange Labs)
Laurent Candillier (Expertise lcandillier.free.fr)
Fabrice Clérot (Orange Labs)
Guillaume Cleuziou (LIFO)
Antoine Cornuéjols (AgroParisTech)
Pascal Cuxac (INIST-CNRS)
Patrick Gallinari (LIP6)
Nistor Grozavu (LIPN)
Romain Guigoures (Data Scientist, Zalando)
Pascale Kuntz-Cosperec (Polytech'Nantes)
Jean-Charles Lamirel (LORIA-SYNALP)
Mustapha Lebbah (LIPN)
Vincent Lemaire (Orange Labs)
Fabien Torre (Lille 3)
Christel Vrain (LIFO)

TABLE DES MATIÈRES

Conférencier invité

Articles sélectionnés

Apprentissage actif pour le clustering semi-supervisé

Nicolas Labroche*

*Université Pierre et Marie Curie, LIP6-UMR-CNRS 7606, Equipe LFI.
nicolas.labroche@lip6.fr,
http://lfi.lip6.fr/

Résumé. Cette présentation s'intéresse aux algorithmes de clustering semi supervisé qui sont capables de tirer profit de connaissances fournies par un expert sous la forme de données étiquetées ou de contraintes entre les données. Après avoir décrit la problématique générale du clustering semi-supervisé, la première partie de cet exposé illustre les principaux bénéfices de ces méthodes par rapport aux algorithmes classiques, en dresse un tour d'horizon rapide et pointe les limitations de ce types de travaux : (1) des connaissances expertes correctes mais non pertinentes peuvent dégrader les performances des algorithmes de clustering, et (2) peu d'attention a été portée jusqu'à présent à la minimisation de l'effort d'annotation de l'expert.

Dans ce contexte, la seconde partie de la présentation présente des résultats de travaux conduits au Laboratoire d'Informatique de Paris 6 dans l'équipe LFI (Learning, Fuzzy and Intelligent systems), concernant la réalisation d'algorithmes actifs pour la sélection de contraintes et de données étiquetées. Ces méthodes ont pour objectif de solliciter efficacement un expert du domaine en ne proposant que les contraintes les plus utiles et en les propageant de façon à réduire rapidement l'ensemble des contraintes candidates potentiellement soumises à l'expert. Notamment, nous décrivons une approche permettant de cibler les questions posées à l'expert sur les zones où les algorithmes font généralement le plus d'erreurs d'affectation.

Enfin, une dernière partie de l'exposé ouvre la discussion en décrivant de nouvelles problématiques de recherche pour le domaine du clustering semi-supervisé.

1 Présentation de Nicolas Labroche

Nicolas Labroche est maître de conférences à l'Université Paris 6 et chercheur au Laboratoire d'Informatique de Paris 6 (LIP6, UMR CNRS 7606) depuis 2004. Ses thèmes de recherche concernent le développement de méthodes de clustering (méthodes floues, passage à l'échelle, traitement de flux, clustering semi-supervisé et recherche de sous-espaces, interprétation/visualisation) pour l'analyse des interactions utilisateurs. Il est diplômé de l'Ecole d'Ingénieurs en Informatique pour l'Industrie de Tours (2000), et du DEA "Signaux et Images en Biologie et Médecine" de l'Université de Tours (2000). Sa thèse, portant sur le développement de méthodes de clustering biomimétiques pour l'analyse de traces d'usages sur Internet, a été réalisée au Laboratoire d'Informatique de Tours (EA CNRS 6300) sous la direction du Pr. G. Venturini et soutenue en 2003. Il est habilité à diriger les recherches depuis décembre 2012. Il a publié plus de 30 publications dans des revues et des conférences internationales.

Clustering Bayésien Parcimonieux Non-Paramétrique

Marius Bartcus*,** Faicel Chamroukhi*,** Hervé Glotin*,**,***

*Université de Toulon, CNRS, LSIS, UMR 7296, 83957 La Garde, France
**Aix Marseille Université, CNRS, ENSAM, LSIS, UMR 7296, 13397 Marseille, France
***Institut Université de France, 75015 Paris, France
{nom}@univ-tln.fr

Résumé. Cet article propose une nouvelle approche Bayésienne non paramétrique pour la classification automatique (clustering). Elle s'appuie sur un modèle de mélange Gaussien infini avec une décomposition en valeurs propres de la matrice de covariance de chaque classe. Le distribution a priori sur les partitions choisie est celle du processus du restaurant chinois (CRP). Cette distribution a priori permet de contrôler la complexité du modèle en reposant sur une formulation statistique solide, et d'estimer automatiquement les nombres de classes à partir des données. De plus, la décomposition en valeurs propres de la matrice de covariance permet d'avoir des modèles flexibles allant du modèle sphérique le plus simple au modèle général qui est plus complexe. L'apprentissage des différents modèles s'effectue par un échantillonnage MCMC de Gibbs. L'approche à été appliquée sur des données simulées et des jeux de données réelles standard afin de valider l'approche et l'évaluer. Les résultats obtenus mettent en évidence l'intérêt du modèle de mélange parcimonieux infini proposé.

1 Introduction

La classification automatique (ou en anglais clustering), est l'une des tâches essentielles en apprentissage automatique et en statistique. L'une des approches les plus populaires en clustering est celle basée sur les modèles de mélange paramétriques finis (McLachlan et Peel., 2000; Fraley et Raftery, 2002; Robert, 2006). Cependant, ces modèles paramétriques se trouvent inadaptés pour représenter des ensembles de données réelles et complexes. L'autre problème de l'approche de clustering à base du modèle du mélange paramétrique fini est celui du choix du nombre de classes, à savoir le problème de sélection de modèle.

Les méthodes Bayésiennes Non Paramétriques (BNP) (Hjort et al., 2010) pour le clustering, y compris le modèle de mélange Gaussien (GMM) infini (Rasmussen, 2000), les CRP et processus de Dirichlet (DP) dans leur version mélange pour le clustering, (Samuel et Blei, 2012; Sudderth, 2006; Pitman, 1995; Aldous, 1985; Ferguson, 1973), fournissent une alternative pertinente pour surmonter ces problèmes. Ils permettent d'éviter de supposer des formes paramétriques restreintes, et permettent ainsi d'inférer la complexité et la structure du modèle à partir des données. L'aspect non-paramétrique de ces approches concerne le fait de supposer que la complexité du modèle associé au nombre de paramètres du modèle croît avec le nombre

et la complexité des données. Ils représentent également une bonne alternative au problème difficile de sélection de modèle rencontrés dans les modèles paramétriques finis. Dans ce travail, nous nous appuyons sur cette formulation bayésienne non paramétrique du mélange Gaussien (GMM) et effectuons une décomposition en valeurs propres de la matrice de covariance de chaque densité Gaussienne comme dans Celeux et Govaert (1995) Banfield et Raftery (1993) pour les GMM finis. Cela conduit à un mélange Gaussien parcimonieux infini qui est plus flexible en termes de modélisation et de son utilisation en clustering, et fournit automatiquement le nombre de classes.

Ce papier est organisé comme suit. La Section 2 rappelle brièvement l'état de l'art sur les mélanges finis parcimonieux pour le clustering. Ensuite, la Section 3 présente la nouvelle approche de clustering parcimonieux non paramétrique proposée. Dans la section 4, on étudie expérimentalement la performance de l'approche proposée en l'appliquant sur des données simulées et réelles.

Notons par $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ un échantillon de n individus i.i.d dans $\mathbb{R}^d$, et soit $\mathbf{z} = (z_1, \ldots, z_n)$ les labels correspondants inconnus où $z_i \in \{1, \ldots, K\}$ représente le label du $i$ème individu $\mathbf{x}_i$, K étant le nombre de classes éventuellement inconnu.

2 Clustering paramétrique Gaussien parcimonieux

Le clustering paramétrique est en général basé sur le modèle de mélange Gaussien fini (GMM) (McLachlan et Peel., 2000). Dans l'approche GMM fini pour le clustering (McLachlan et Peel., 2000; Fraley et Raftery, 2002), les données suivent la densité mélange suivante :

$$p(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}_k(\mathbf{x}_i|\theta_k) \tag{1}$$

ou $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\theta}_k\}_{k=1}^{K}$ est le vecteur paramètre du GMM qui comprend les proportions du mélange π_k qui sont non-négatives et somment à 1 et $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ sont le vecteur moyenne et la matrice de covariance pour la $k$ième composante Gaussienne du mélange.

Le mélange Gaussien fini parcimonieux (Celeux et Govaert, 1995; Banfield et Raftery, 1993) exploite une décomposition en valeurs propres des matrices des covariances. Ceci fournit une variété de modèles très flexibles. En effet, la décomposition en valeurs propres de la matrice de covariance de chaque composante gaussienne, permet d'avoir des classes ayant différents volumes, formes et orientations (Celeux et Govaert, 1995; Banfield et Raftery, 1993). Cette paramétrisation de la matrice de covariance est de la forme suivante :

$$\boldsymbol{\Sigma}_k = \lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T \tag{2}$$

où λ_k est un scalaire qui représente le volume du cluster k, $\mathbf{D}_k$ est une matrice orthogonale qui représente son orientation et $\mathbf{A}_k$ est une matrice diagonale de déterminant un qui représente sa forme. Cette décomposition conduit à plusieurs modèles flexibles (Celeux et Govaert, 1995) allant de modèles sphériques simples aux modèles généraux plus complexes. Les paramètres $\boldsymbol{\theta}$ du modèle de mélange peuvent être estimés par maximum de vraisemblance (MV) en maximisant la vraisemblance des données observées $p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \, \mathcal{N}_k(\mathbf{x}_i|\theta_k)$ où dans un cadre de maximum a posteriori (MAP) (cadre Bayésien) en maximisant la loi a posteriori des paramètres suivante : $p(\boldsymbol{\theta}|\mathbf{X}) = p(\boldsymbol{\theta})p(\mathbf{X}|\boldsymbol{\theta})$, $p(\boldsymbol{\theta})$ étant la distribution a priori de $\boldsymbol{\theta}$.

L'estimation du maximum de vraisemblance s'effectue en général par l'algorithme Espérance-Maximisation (EM) (Dempster et al., 1977; McLachlan et Krishnan, 1997), où par l'une de ses extensions comme CEM, GEM, etc. L'estimation du maximum a posteriori peut également être effectuée par l'algorithme EM dans le cas de lois conjuguées comme dans Fraley et Raftery (2007). Les techniques d'échantillonnages Markov Chain Monte Carlo (MCMC) peuvent être utilisées aussi pour estimer les paramètres comme dans (Bensmail et al., 1997; Bensmail et Meulman, 2003; Ormoneit et Tresp, 1998). Pour le cas spécifique du GMM fini parcimonieux, plusieurs algorithmes d'apprentissage ont été proposés. Ils s'appuient en majorité sur une estimation par MV via l'algorithme EM ou sur l'une de ses extensions (Banfield et Raftery, 1993; Celeux et Govaert, 1995). On trouve également la version Bayésienne de l'estimation, par l'algorithme EM comme dans Fraley et Raftery (2007), ou par des techniques d'échantillonnages MCMC, notamment l'échantillonneur de Gibbs comme dans Bensmail et al. (1997); Bensmail et Meulman (2003). Cependant, dans l'approche basée sur le mélange GMM fini pour le clustering, le nombre de classes doit être fourni à l'algorithme. L'un des principaux problèmes de cette approche de clustering à base de modèle de mélange fini est donc celui du choix du nombre de composants du mélange (classes) qui correspond au mieux aux données. Le choix du nombre optimal de classes dans le cas du mélange paramétrique Bayésien ou non Bayésien peut être effectuée via des critères de sélection de modèles se basant sur une log-vraisemblance pénalisée, comme BIC (Schwarz, 1978), AIC (Akaike, 1974), etc.

3 Clustering Bayésien non-paramétrique parcimonieux

Les mélanges bayésiens non-paramétriques (BNP) pour le clustering offrent une alternative reposant sur un formalisme statistique solide pour résoudre ce problème de choix du nombre de classes ; le nombre de classes est inféré directement à partir des données (Hjort et al., 2010; Samuel et Blei, 2012; Sudderth, 2006; Rasmussen, 2000), plutôt qu'en s'appuyant à une approche en deux étapes comme pour les mélanges finis. Ce clustering non-paramétrique basé sur les mélanges infinis suppose que les données observées sont crées par un nombre infini des classes, mais seulement un nombre fini d'entre elles a réellement généré les données. Ceci est effectué en posant un processus général comme distribution a priori sur les partitions possibles, ce qui n'est pas restrictif comparé à l'inférence bayésienne classique, et ce de telle manière que seulement un nombre fini de classes sera réellement actif. Une telle distribution a priori peut être celle du processus du restaurant chinois (CRP) (Aldous, 1985; Pitman, 2002; Samuel et Blei, 2012) ou un processus de Dirichlet dans une version mélange pour le clustering (DPM) (Ferguson, 1973; Samuel et Blei, 2012). Plusieurs modèles bayésiens non-paramétriques ont considéré le cas du modèle de mélange Gaussien (GMM) dans sa version générale (non parcimonieuse). On distingue le mélange Gaussien infini (Rasmussen, 2000), la modélisation par mélange de densités et un processus du restaurant chinois (CRP) (Samuel et Blei, 2012), ou le mélange de processus de Dirichlet (DPM) (Antoniak, 1974; Samuel et Blei, 2012). Pour un état de l'art détaillé sur ces approches, le lecteur peut se referer par exemple à ces deux références Samuel et Blei (2012); Sudderth (2006).

Dans l'approche de clustering Bayésien non-paramétrique (BNP) parcimonieux proposée, nous exploitons la décomposition en valeurs propres de la matrice de covariance de chaque classe comme dans Celeux et Govaert (1995) et Banfield et Raftery (1993) pour les GMM finis, et l'intégrant dans un cadre de mélange Gaussien infini. Cela conduit à un mélange Gaussien

infini parcimonieux qui est très flexible en terme de modélisation, et qui permet d'estimer automatiquement le nombre de classes à partir des données. Nous utilisons le processus du restaurant chinois (CRP) comme distribution a priori sur les partitions.

3.1 Processus du restaurant chinois (CRP) et mélange parcimonieux pour le clustering

Le Processus du restaurant chinois (CRP) fournit une distribution sur les partitions infinies des données, qui représente la distribution sur les entiers positifs $1, \ldots, n$. Considérons la distribution jointe suivante sur les labels correspondants des données : $p(z_1, \ldots, z_n) = p(z_1)p(z_2|z_1)p(z_3|z_1, z_2) \ldots p(z_n|z_1, z_2, \ldots, z_{n-1})$. Chaque terme de cette distribution jointe peut être calculé à partir de l'a priori CRP comme suit. Supposons qu'il y a un restaurant avec un nombre infini de tables et dans lequel les clients viennent s'installer dans les tables. Les clients sont sociables, de telle façon que le $i$ème client s'installe à la $k$ième table avec une probabilité proportionnelle au nombre de clients qui y sont déjà installés (n_k), et peut choisir une nouvelle table avec une probabilité proportionnelle à un petit réel positif α représente le paramètre de concentration pour le CRP. Cela peut être formulé comme suit :

$$p(z_i = k|z_1, ..., z_{i-1}) = \text{CRP}(z_1, \ldots, z_{i-1}; \alpha) = \begin{cases} \frac{n_k}{i-1+\alpha} & \text{if} \quad k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & \text{if} \quad k > K_+ \end{cases} \tag{3}$$

où K_+ est le nombre de tables pour lesquelles le nombre de clients installés à la $k$ième table est $n_k > 0$, $k \leq K_+$ signifie que k est une table précédemment occupé et $k > K_+$ signifie une nouvelle table à été choisie pour être occupée. A partir de cette distribution, dans un contexte de clustering, on peut donc commencer avec une seule classe et ensuite affecter de nouvelles données à des classes éventuellement nouvelles. En effet, en clustering se basant sur le CRP, les clients correspondent à des données et les tables à des classes. Dans la version mélange CRP pour le clustering, l'a priori du CRP $\text{CRP}(z_1, \ldots, z_{i-1}; \alpha)$ est complété par une densité de paramètres $\boldsymbol{\theta}$ (par exemple dans le cas du GMM une densité Gaussienne multivariée), pour chaque table (classe), et une distribution a priori (G_0) pour les paramètres. Par exemple, dans le cas du mélange Gaussien on peut utiliser des lois conjuguées, à savoir une distribution normale pour chaque moyenne et une distribution inverse-Wishart pour chaque matrice de covariance.

En terme de modèle génératif, cela correspond au processus suivant. Le $i$ème client qui est installé à la table $z_i = k$ choisit un plat (le paramètre $\boldsymbol{\theta}_{z_i}$) de la distribution a priori des plats de la table (cluster). Cela peut se résumer selon le processus génératif suivant.

$$\boldsymbol{\theta}_i \sim G_0 \tag{4}$$

$$z_i \sim \text{CRP}(z_1, \ldots, z_{i-1}; \alpha) \tag{5}$$

$$\mathbf{x}_i \sim p(.|\boldsymbol{\theta}_{z_i}) \cdot \tag{6}$$

Selon ce modèle génératif, les paramètres générés $\boldsymbol{\theta}_i$ présentent une propriété de regroupement automatique, c'est-à-dire qu'ils partagent des valeurs répétées avec une probabilité positive et où les valeurs uniques de $\boldsymbol{\theta}_i$ partagées parmi les variables correspondent à des tirages indépendants de la distribution de base G_0 (Ferguson, 1973; Samuel et Blei, 2012). La structure de valeurs partagées définit une partition des entiers de 1 à n, et la distribution de cette partition est un CRP (Ferguson, 1973; Samuel et Blei, 2012). Dans notre mélange Gaussien parcimonieux

infini proposé, les paramètres $\boldsymbol{\theta}_i$ qui comprennent, pour chaque classes, le vecteur moyen et la matrice de covariance décomposée en valeurs propres, permettra ainsi de fournir des classes plus flexibles avec éventuellement différents volumes, formes et orientations. En terme d'interprétation du processus du restaurant chinois, cela peut être vu comme une variabilité et richesse des plats.

3.2 L'échantillonnage MCMC pour l'apprentissage du modèle

Nous avons utilisé un échantillonnage de Gibbs (Rasmussen, 2000; Neal, 1993; Wood et al., 2006; Samuel et Blei, 2012) pour apprendre le modèle parcimonieux non paramétrique Bayésienne proposée.Les distribution a priori utilisée sur les paramètres du modèle dépend du type du modèle parcimonieux considéré. Ainsi, l'échantillonnage des paramètres varie selon le modèle parcimonieux choisi.Nous avons étudié jusqu'à présent sept modèles parcimonieux, couvrant les trois familles du modèle de mélange : la famille générale, la famille diagonale et la famille sphérique. Le tableau 1 présente les modèles considérés et les distribution a priori correspondant à chaque modèle utilisé dans l'échantillonnage de Gibbs.

Decomposition	Type du Modèle	Prior	Appliqué à
$\lambda\mathbf{I}$	Sphérique	$\mathcal{IG}$	λ
$\lambda_k\mathbf{I}$	Sphérique	$\mathcal{IG}$	λ_k
$\lambda\mathbf{B}$	Diagonal	$\mathcal{IG}$	chaque élément de la diagonale de $\lambda\mathbf{B}$
$\lambda_k\mathbf{B}$	Diagonal	$\mathcal{IG}$	chaque élément de la diagonale de $\lambda_k\mathbf{B}$
$\lambda\mathbf{DAD}^T$	Général	$\mathcal{IW}$	$\mathbf{\Sigma} = \lambda\mathbf{DAD}^T$
$\lambda_k\mathbf{DAD}^T$	Général	$\mathcal{IG}$ et $\mathcal{IW}$	λ_k et $\mathbf{\Sigma} = \mathbf{DAD}^T$
$\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$	Général	$\mathcal{IW}$	$\mathbf{\Sigma}_k = \lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$

TAB. 1: Les GMMs parcimonieux considérés en paramétrant la matrice de covariance et la distribution a priori associée à chaque cas. $\mathcal{I}$ signifie une distribution inverse, $\mathcal{G}$ une distribution Gamma et $\mathcal{W}$ une distribution de Wishart.

Le pseudo-code 1 montre l'algorithme détaillé pour l'échantillonnage de Gibbs dans l'apprentissage des modèles de mélange infini parcimonieux gaussien. Une des étapes principales dans cet algorithme est l'échantillonnage des étiquettes avec la distribution a priori du processus du restaurant chinois (CRP) décrit dans la section 3.1.

Le coût de la méthode est principalement lié à la simulation des étiquettes z_i (et donc au nombres de classes et au nombre d'individus) et des paramètres $\boldsymbol{\theta}_i$ (donc nombre et dimension des paramètres). Plus précisément, la complexité de chaque itération de Gibbs de l'algorithme proposé est proportionnelle à la valeur actuelle du nombre de classes K (K étant estimé automatiquement), et varie donc aléatoirement d'une itération à l'autre, du fait de la distribution sur le nombre de classes. Asymptotiquement, K tend vers $\alpha\log(n)$ quand n tend vers l'infini (Antoniak, 1974). Chaque itération requiert donc $O(\alpha n\log(n))$ opérations pour simuler les étiquettes des classes z_i. La simulation des paramètres (moyennes et matrices de covariances), requiert quant à elle, dans le pire des cas (matrice de covariance pleine) approximativement $O\left(\alpha\log(n)\left(d+\frac{d(d+1)}{2}\right)\right)$ ce qui nous donne une complexité totale de $O\left(\alpha\log(n)\left(N+d+\frac{d(d+1)}{2}\right)\right)$.

Algorithm 1 L'échantillonnage de Gibbs pour l'IPGMM proposé

Entrées :données $\{\mathbf{x}_i\}$, hyper-paramètres et nombre d'échantillons

1: iteration de Gibbs $q \leftarrow 0$
2: hyper-paramètres $\boldsymbol{\alpha}^{(q)}$
3: Commencer avec une classe : $K_+ \leftarrow 1$
4: **for** $i = 1, \ldots, n$ **do**
5: Simuler le label $z_i^{(q)} \sim \text{CRP}(\{z_1, \ldots, z_n\}_{\setminus z_i}; \alpha^{(q)})$
6: Si $z_i^{(q)} = K_+ + 1$ nous avons une nouvelle classe, et on augmente donc K_+ : $K_+ = K_+ + 1$
7: Simuler les paramètres de classe $\boldsymbol{\theta}_i^{(q)}$ selon la distribution *a priori* comme dans le tableau 1.
8: **end for**
9: Simuler les hyper-paramètres $\boldsymbol{\alpha}^{(q)}$
10: $\mathbf{z}^{(q+1)} \leftarrow \mathbf{z}^{(q)}$
11: $\boldsymbol{\alpha}^{(q+1)} \leftarrow \boldsymbol{\alpha}^{(q)}$
12: $q \leftarrow q + 1$

Outputs : $\{\hat{\boldsymbol{\theta}}, \hat{\mathbf{z}}, \hat{K} = K_+\}$

4 Expérimentations

Nous avons effectué des expérimentations sur des données simulées et réelles afin d'évaluer la méthode non paramétrique proposée. A travers ces expérimentations, nous essayons d'abord de souligner la flexibilité de du mélange bayésien non paramétrique parcimonieux proposé, et ce en termes de modélisation, ainsi qu'en terme de son utilisation pour le clustering et la sélection du nombre de classes. Les résultats numériques sont reportés en termes de comparaisons des valeurs de la log-vraisemblance pour les données observés [1], la partition estimée des données, et l'évaluation du nombre de classes estimé. Quand le nombre de classes de la partition estimée est égal au nombre réel de classes, nous calculons également le taux d'erreur de classification. Nous avons comparé le mélange bayésien non paramétrique parcimonieux, avec notamment l'approche bayésienne paramétrique se basant sur le mélange Gaussien fini.

Pour les approches bayésiennes paramétriques (le cas fini), nous avons utilisé l'échantillonnage de Gibbs pour apprendre les paramètres du modèle. Pour chaque ensemble de données simulées, et pour chaque valeur de K, l'échantillonneur de Gibbs est exécuté 10 fois avec différentes initialisations, dans chaque exécution sont générés 2000 échantillons. La solution correspondant à la plus haute probabilité a posteriori est sélectionnée. La selection du nombre de classes est dans ce cas effectuée par le critère AWE (approximate weight of evidence) comme dans Banfield et Raftery (1993).

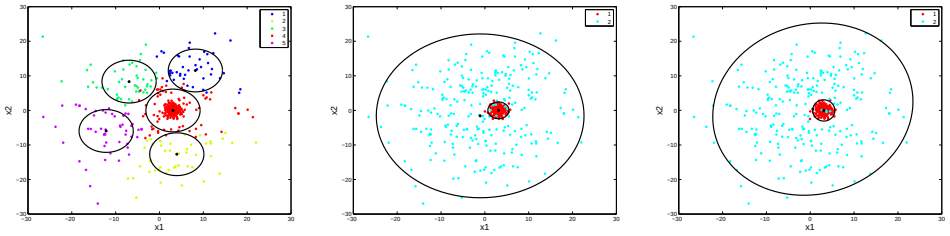
Pour l'approche bayésienne non paramétrique proposée (IPGMM), nous avons utilisé l'échantillonneur de Gibbs. De même, l'échantillonneur de Gibbs est exécuté dix fois sur chaque jeu de données et la meilleure solution au sens du maximum a posteriori est alors sélectionnée ; le

1. Pour les mélanges infinis, les proportions du mélange sont estimées à partir de la partition obtenue, comme dans Wood et al. (2006)

nombre de classes étant dans ce cas estimé automatiquement au cours de l'échantillonnage de Gibbs.

4.1 Expérimentations sur des données simulées

Nous avons considéré une situation à deux classes pour illustrer l'approche de clustering proposée. Cette situation est la même que dans Celeux et Govaert (1995) et consiste en un échantillon de $n = 500$ observations simulées selon un mélange gaussien à composantes en $\mathbb{R}^2$ et de paramètres $\pi_1 = \pi_2 = 0.5$, $\boldsymbol{\mu}_1 = (0, 0)^T$, $\boldsymbol{\mu}_2 = (3, 0)^T$, $\boldsymbol{\Sigma}_1 = 100\,\mathbf{I}_2$ et $\boldsymbol{\Sigma}_2 = \mathbf{I}_2$. La figure 1 montre les données simulées et les partitions obtenues par trois modèles de l'approche de clustering bayésien non paramétrique parcimonieux proposée.



$\lambda\mathbf{I}$ $\quad$ $\lambda_k\mathbf{I}$ $\quad$ $\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$

FIG. 1: Un jeu de données à classes et les partition obtenues par le mélange bayésien non-paramétrique pour trois modèles parcimonieux : modèle sphérique à volume égal (gauche), modèle sphérique à volume différent (milieu) et modèle général (droite).

On peut observer que le fait de supposer le même volume pour toutes les classes (modèle $\lambda\mathbf{I}$) ne permet pas de reconstruire la vraie structure cachée des données. Alors que, le modèle parcimonieux dans lequel on suppose seulement que seul le volume des classes peut varier (modèle $\lambda_k\mathbf{I}$), fournit une partition très satisfaisante et qui est très proche de la partition réelle. En effet, le nombre de classes estimé correspond au vrai nombre de classes et le taux d'erreur de classification est de 4.80%. Ce taux d'erreur est même légèrement moins élevé que celui du modèle général (modèle $\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$) alors que ce dernier est beaucoup plus complexe. En effet, le modèle le plus général $\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$ qui permet également d'avoir des volumes de classes différents, fournit un résultat très semblable pour un taux d'erreur est égal à $4,40\%$. Le meilleur modèle au sens de la valeur du log-vraisemblance correspond au modèle sphérique avec différents volumes ($\lambda_k\mathbf{I}$) et surpasse donc même le modèle général qui n'est parcimonieux. On peut conclure que, en ce clustering non-paramétrique basé sur le mélange Gaussien parcimonieux, il est important de prendre en compte des classes avec des volumes différents ; pour cet ensemble de données au moins, le modèle sphérique avec différents volumes ($\lambda_k\mathbf{I}$), est le meilleur modèle.

En termes de comparaison des différents modèles non-paramétriques parcimonieux proposés, et de ces modèles avec l'approche bayésienne paramétrique basée sur le mélange Gaussien fini (GMM), le tableau 2 reporte les valeurs de la log-vraisemblance et du nombre de classes estimé obtenus pour les données simulées par l'approche bayésienne paramétrique basée sur le

mélange Gaussien fini, et l'approche proposée (IPGMM). On peut notamment observer qu'on

TAB. 2: Valeurs de la log-vraisemblance et nombre de classes estimé obtenus pour les données simulées par l'approche bayésienne paramétrique basée sur le mélange Gaussien fini (GMM), et l'approche proposée (IPGMM).

	Données simulées			
	GMM		IPGMM	
Vrai valeur de K	2			
Modèle	$\hat{K}$	log-lik	$\hat{K}$	log-lik
$\lambda\mathbf{I}$	2	-5.5836	5	-5.7707
$\lambda_k\mathbf{I}$	5	-5.1577	2	$\mathbf{-5.1111}$
$\lambda\mathbf{B}$	4	-5.4745	9	-5.4289
$\lambda_k\mathbf{B}$	5	-5.1577	7	-5.2888
$\lambda\mathbf{D}\mathbf{A}\mathbf{D}^T$	2	-5.5608	8	-5.4125
$\lambda_k\mathbf{D}\mathbf{A}\mathbf{D}^T$	4	-5.4175	7	-5.3127
$\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$	5	$\mathbf{-5.0938}$	2	-5.1194

a deux modèles pour chaque approche pour lesquels le nombre de classes estimé correspond au nombre de classes réel. L'approche proposée peut donc être une bonne alternative moins coûteuse pour l'approche de clustering paramétrique standard.

4.2 Expérimentations sur des données réelles

Nous avons considéré deux jeux de données réels bien connus qui sont Iris et Geyser. Rappelons que Iris est un ensemble de 150 données de dimension 4 comportant trois classes. Le jeux de données Geyser contient 272 individus de dimension 2, le nombre de classes est cependant inconnu mais plusieurs études de clustering le situent entre deux et trois.

La figure 2 montre la partition et les densités estimées par l'approche proposée pour les deux jeux de données pour trois modèles parcimonieux différents dont le modèle général : modèle sphérique (gauche), modèle diagonal (milieu) et modèle général (droite). Les trois modèles permettent d'avoir des classes ayant des volumes différents (à travers λ_k).

Le tableau 3 reporte les résultats numériques pour les deux jeux de données.

Pour le jeu de données iris, on peut remarquer que les deux modèles parcimonieux (sphérique et diagonal, à volume différents) permettent de retrouver le bon nombre de classes et reconstruire la structure des données, alors que le modèle général, qui est plus complexe et est le plus souvent utilisé, quant à lui sous-estime le nombre de classes. Notons que le taux d'erreur pour le modèle diagonal $\lambda_k\mathbf{B}$ est de 5.33% et celui du modèle sphérique $\lambda_k\mathbf{I}$ est de 10.66%. Pour l'approche fini, les modèles pour lesquels on trouve le bon nombre de classes sont les modèles diagonaux $\lambda_k\mathbf{B}$ et $\lambda\mathbf{B}$. Les taux d'erreur correspondant sont respectivement 11.33% $\lambda_k\mathbf{B}$ et 9.33%. Ceci montre un avantage de cette alternative non-paramétrique.

Pour les données Geyser, on peut observer sur les résultats graphiques que les partitions obtenues par l'approche non-paramétrique en utilisant le modèle sphérique à volume différent ($\lambda_k\mathbf{I}$) et le modèle général ($\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$) sont similaires. On peut observer aussi que la partition fournie par le modèle diagonal dans cas infini peut être retenue. Ensuite, on peut noter

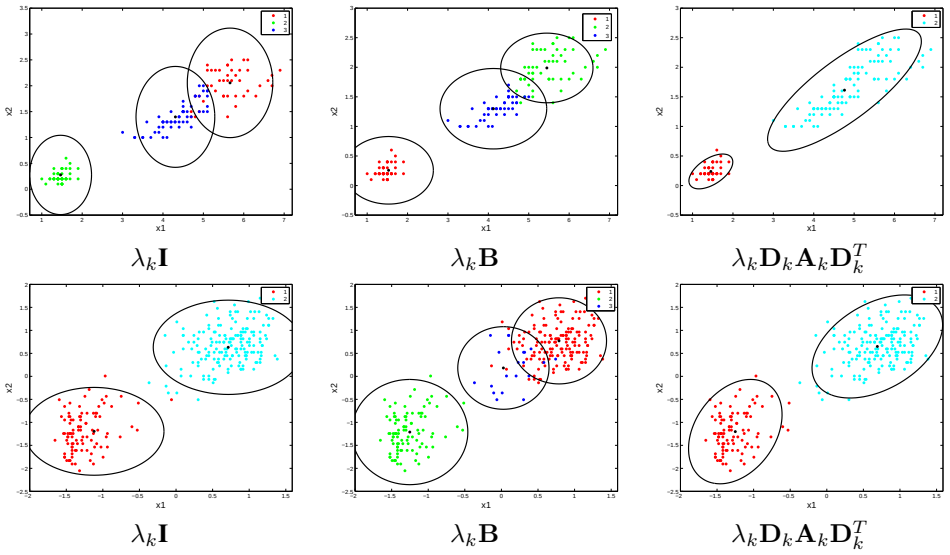


FIG. 2: Résultats obtenus pour les Iris (haut) et Geyser (bas) obtenues par les trois GMMs infinies parcimonieux : modèle sphérique (gauche), diagnoal (milieu) et général (droite).

TAB. 3: Valeurs de la log-vraisemblance et nombre de classes estimé obtenus pour les données simulées par l'approche bayésienne paramétrique basée sur le mélange Gaussien fini (GMM), et l'approche proposée (IPGMM), pour les données Iris et Geyser.

	Iris				Geyser			
	GMM		IPGMM		GMM		IPGMM	
Vrai valeur de K	3				inconnue			
Modèle	$\hat{K}$	log-lik	$\hat{K}$	log-lik	$\hat{K}$	log-lik	$\hat{K}$	log-lik
$\lambda\mathbf{I}$	5	-1643.5	5	-1712.6	3	-1597.4	10	-1659.8
$\lambda_k\mathbf{I}$	5	-1663.8	3	-1722.8	2	-1630.6	2	-1634.5
$\lambda\mathbf{B}$	3	-1700.4	4	-1647.5	2	-1622.2	3	-1605.1
$\lambda_k\mathbf{B}$	3	-1714.7	3	-1707.9	2	-1639.6	3	-1609.0
$\lambda\mathbf{D}\mathbf{A}\mathbf{D}^T$	2	-1641.6	4	-1566.4	2	-1605.6	3	$\mathbf{-1593.9}$
$\lambda_k\mathbf{D}\mathbf{A}\mathbf{D}^T$	2	-1629.3	4	-1562.8	2	-1638.0	2	-1601.9
$\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$	2	$\mathbf{-1583.1}$	2	$\mathbf{-1559.7}$	2	$\mathbf{-1594.7}$	2	-1595.7

également que, sauf pour le modèle sphérique à volume égal ($\lambda\mathbf{I}$), le nombre de classes estimé est conforme à celui de la littérature (entre deux et trois). L'approche paramétrique quant à elle l'estime à deux sauf également pour le cas sphérique à volume égal.

5 Conclusion

Dans cet article, nous avons présenté une nouvelle approche bayésienne non-paramétrique de clustering qui est basée sur un mélange Gaussien infini parcimonieux. Le mélange Gaussien infini parcimonieux se base sur une décomposition en valeurs propres de la matrice de covariance de chaque classe, et le processus du restaurant chinois (CRP) comme a priori. Cette approche permet de dériver plusieurs modèles flexibles et évite le problème difficile de sélection de modèle rencontré dans l'approche paramétrique des mélanges Gaussiens. Nous avons illustré cette méthode sur des données simulées et nous l'avons appliqué sur des données réelles. Les résultats obtenus mettent en évidence l'intérêt d'utiliser le clustering bayésien non-paramétrique parcimonieux comme une bonne alternative pour le clustering parcimonieux à base de GMM fini. Notre travail actuel portent sur plus d'expériences sur des données réelles et simulées. Un des points qui reste ouvert comme en toute approche de clustering est celui de l'évaluation de la partition obtenue. Les futurs travaux concerneront également d'autres techniques pour apprendre les modèles, notamment des méthodes variationnelles.

Références

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716–723.

Aldous, D. (1985). Exchangeability and related topics. In *École d'Été St Flour 1983*, pp. 1–198. Springer-Verlag. Lecture Notes in Math. 1117.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics 2*(6), 1152–1174.

Banfield, J. D. et A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*(3), 803–821.

Bensmail, H., G. Celeux, A. E. Raftery, et C. P. Robert (1997). Inference in model-based cluster analysis. *Statistics and Computing 7*(1), 1–10.

Bensmail, H. et J. J. Meulman (2003). Model-based clustering with noise : Bayesian inference and estimation. *J. Classification 20*(1), 049–076.

Celeux, G. et G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition 28*(5), 781–793.

Dempster, A. P., N. M. Laird, et D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society, B 39(1)*, 1–38.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The Annals of Statistics 1*(2), 209–230.

Fraley, C. et A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association 97*, 611–631.

Fraley, C. et A. E. Raftery (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification 24*(2), 155–181.

Hjort, N., H. C., P. Muller, et S. G. Waller (2010). *Bayesian Non Parametrics*. Cambrige University Press.

McLachlan, G. J. et T. Krishnan (1997). *The EM algorithm and extensions*. New York : Wiley.

McLachlan, G. J. et D. Peel. (2000). *Finite mixture models*. New York : Wiley.

Neal, R. M. (1993). Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.

Ormoneit, D. et V. Tresp (1998). Averaging, maximum penalized likelihood and bayesian estimation for improving gaussian mixture probability density estimates. *IEEE Transactions on Neural Networks 9*(4), 639–650.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probab. Theory Related Fields 102*(2), 145–158.

Pitman, J. (2002). Combinatorial stochastic processes. Technical Report 621, Dept. of Statistics. UC, Berkeley.

Rasmussen, C. (2000). The infinite gaussian mixture model. *Advances in neuronal Information Processing Systems 10*, 554 – 560.

Robert, C. (2006). *Le choix bayésien : principes et pratique*. Statistique et probabilités appliquées. Springer.

Samuel, J. G. et D. M. Blei (2012). A tutorial on bayesian non-parametric model. *Journal of Mathematical Psychology 56*, 1–12.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics 6*, 461–464.

Sudderth, E. B. (2006). *Graphical models for visual object recognition and tracking*. Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Wood, F., T. L. Griffiths, et Z. Ghahramani (2006). A non-parametric bayesian method for inferring hidden causes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Summary

This paper proposes a new Bayesian non-parametric approach for cluster analysis. It relies on an Infinite Gaussian mixture model with an eigenvalue decomposition of the covariance matrix of each cluster, and a Chinese Restaurant Process (CRP) prior. The CRP prior allows to control the model complexity in a principled way, and to automatically learn the number of clusters from the data, and the covariance decomposition allows to fit various flexible models going from simplest spherical ones to the more complex general one. We develop an MCMC Gibbs sampler to learn the various models and apply it to both simulated and real data. The obtained results highlight the interest of the proposed infinite parsimonious mixture model.

Analyse des trajets de Vélib par clustering

Yousra Chabchoub* Christine Fricker**

*ISEP, 21 rue d'assas 75006 Paris
yousra.chabchoub@isep.fr
**INRIA Paris Rocquencourt, Domaine de Voluceau 78153 Le Chesnay, France
christine.fricker@inria.fr

Résumé. Les Vélos en libre service deviennent un mode de transport urbain à Paris et dans de nombreuses autres grandes villes. Leur gestion est complexe et il est parfois difficile de trouver un vélo ou une borne libre dans certaines stations. Nous avons réalisé dans cet article une étude exploratoire des données de trajets de Vélib afin de classer les stations en plusieurs catégories en se basant sur la variation des trajets relatifs à ces stations au cours de la journée. Pour ce faire, nous avons appliqué l'algorithme de clustering largement utilisé "Kmeans". Le but de cette étude est de localiser les stations problématiques qui sont souvent vides ou souvent pleines afin de comprendre les besoins des usagers. De telles informations permettront de proposer par la suite des moyens d'amélioration de la disponibilité du système Vélib.

1 Introduction

Depuis son lancement en 2007, Vélib s'est imposé dans le paysage parisien et a servi de modèle à des systèmes analogues dans nombre de métropoles internationales. La géographie des lieux, zones basses et collines, et l'urbanisme, entre autres zones d'habitation et de bureaux, créent des différences fortes de fréquentation des stations. Cette fréquentation fluctue de plus au cours de la journée. Tout ceci déséquilibre le réseau.

Une bonne compréhension de l'utilisation actuelle du système Vélib axée sur les stations très déséquilibrées, est primordiale afin de proposer des méthodes qui améliorent les actions de régulation ou des méthodes incitatives qui encouragent les usagers à mieux répartir les vélos entre les stations. L'analyse des vrais besoins des usagers permet aussi d'envisager les actions d'évolution et d'extension du système Vélib, et apportera des éléments de réponse à plusieurs questions comme : Comment étendre le système Vélib à la banlieue parisienne ? Faut-il modifier la cartographie de Vélib en période de vacances ? Quel est l'impact des évolutions des autres moyens de transport sur le système Vélib ?

L'intérêt soulevé par ces systèmes a suscité récemment de nombreux travaux de recherche, concernant à la fois des travaux sur des modèles pour comprendre leur fonctionnement et des études relatives aux données.

Dans le premier cas, le problème de la régulation par les camions occupe une large place : il s'agit d'optimiser les actions des véhicules qui redistribuent les vélos à des stations bien

choisies. Ces opérations étant plus nombreuses la nuit, les études ont porté sur un réseau statique (voir l'étude réalisée par Chemla et al. (2012)) sans évolution extérieure au cours du temps comme des arrivées d'usagers ou des déplacements de vélos par des usagers. Les travaux récents tendent à intégrer l'aspect dynamique du réseau (cf par exemple Raviv et al. (2012), Schuijbroek et al. (2013) et les références incluses). L'étude de ces systèmes en tant que grands réseaux stochastiques remontent à George et Xia (2010). Puis la prise en compte de la capacité limitée des stations dans le modèle, en le rendant plus pertinent et encore analysable, a permis de discuter différentes politiques de repose et d'envisager une régulation dite naturelle, c'est-à-dire par les usagers eux-mêmes qui vont vers une station mieux choisie pour le fonctionnement du réseau. Fricker et Gast (2014) montrent que même dans un réseau parfaitement homogène, la proportion de stations vides ou pleines est grande et admet un minimum en fonction du nombre total de vélos, qui peut être quantifié. Le comportement d'un système très simple avec clusters de stations de même paramètre est aussi étudié dans Fricker et al. (2012). Pfrommer et al. (2013) ont proposé une méthode de régulation dynamique par les camions ainsi qu'une politique de tarification qui calcule en temps réel le tarif au moment du dépôt du vélo, en fonction du taux de remplissage de la station finale. Le but est d'encourager les usagers à déposer leurs vélos dans les stations les moins chargées. On peut émettre des réserves sur une politique de tarification, vu le faible coût de ce moyen de transport, mais elle peut être remplacée par une politique d'incitation. Le modèle est testé en utilisant des heuristiques extraites du système de vélos en libre service de Londres.

Dans le cas des données, des études concernent les réseaux parisien par Nair et al. (2013) où il s'agit de capter la matrice de routage du système, et lyonnais par Borgnat et al. (2011), où on détermine les groupes de stations qui échangent des vélos. Certaines autres études ont porté sur l'application d'algorithmes de clustering sur les stations afin de chercher des similarités entre les activités ou les états des stations au cours du temps. Parmi les méthodes de clustering non supervisées les plus utilisées, nous pouvons citer Kmeans et le clustering hiérarchique. Ces techniques sont simples à implémenter et ne nécessitent pas d'apprentissage. La classification hiérarchique consiste à partir d'un seul élément par classe et fusionner ensuite d'une façon récursive les classes en une classe mère selon un critère de similarité pour obtenir des clusters de plus en plus grands. Cette méthode appellée aussi 'Dendrogram Clustering" a été appliquée par Froehlich et al. (2009) sur les données des vélos en libre service de Barcelone. Dans Côme et Oukhellou (2012), les auteurs ont introduit un modèle statistique de clustering qu'ils ont testé sur les données du réseau Vélib. Dans cet article, vous avons choisi d'utiliser l'algorithme de classification Kmeans, proposé par James MacQueen (1967). Kmeans est un algorithme récursif, simple à implémenter. Il est parfaitement adapté aux données qui comportent la notion de centre, ce qui est le cas pour le système Vélib où certains lieux sont de véritables centres d'attraction. Kmeans a aussi l'avantage de converger très rapidement dans la pratique (faible nombre d'itérations) et d'avoir une complexité très réduite (faible temps d'exécution).

Le but de cet article est d'explorer des données relatives aux trajets Vélib afin de classer les stations en différentes catégories suivant leurs implications dans les trajets des usagers au cours de la journée. La classification est basée sur un algorithme de clustering très répandu appelé "Kmeans". Les résultats sont comparés à ceux obtenus dans Côme et Oukhellou (2012). L'analyse des données et le découpage en zones homogènes permettront de découvrir les paramètres réels du trafic, d'identifier les régions saturées de vélos ou vides, notamment leur localisation et leur état au cours de la journée. De telles données sont cruciales pour la compréhension et

l'amélioration de la disponibilité du système Vélib.

2 Description de l'algorithme Kmeans

L'algorithme Kmeans a été proposé par James MacQueen (1967). C'est l'un des plus simples algorithmes de clustering. Il vise à répartir les objets en classes en optimisant un critère particulier. Basé sur un apprentissage non supervisé, il ne suppose aucune connaissance a priori sur les données traitées. Kmeans est aujourd'hui largement utilisé dans des domaines divers comme l'imagerie médicale, le marketing ou les sciences humaines.
L'idée-clé de Kmeans est d'attribuer dynamiquement les objets à des centres de classes qui sont à leur tour recalculés à chaque itération. Chaque objet est représenté par un vecteur de $\mathbb{R}^p$ et est associé au centre de classe le plus proche au sens d'une métrique préalablement définie. La distance euclidienne est la métrique la plus utilisée. Dans ce ce cas les centres de classes sont définis comme des centres de gravité. L'affectation de tous les objets et la mise à jour des centres de classes seront répétées jusqu'à la convergence définie par la stabilité des centres de classes.
Le nombre de classes K doit être préalablement choisi. Il dépend de l'application et du niveau de détail visé par le clustering. Les centres des classes sont arbitrairement initialisés au démarrage de l'algorithme. Cette affectation aléatoire peut engendrer des clusters vides dans le cas où le centre du cluster est très éloigné des toutes les données à partitionner. Une meilleure façon d'initialiser les centres des classes est proposée par Pakhira (2009) afin d'éviter les clusters vides. Voici une description détaillée de l'algorithme ainsi obtenu :

L'algorithme Kmeans

---

1. Initialisation : Tirer au hasard K objets parmi l'ensemble des données à partitionner. Ces objets seront désignés comme centres des K classes.

2. Répéter l'étape (a) et (b) jusqu'à ce que les centres des classes soient constants (convergence de l'algorithme)
   (a) Attribuer chaque objet au centre de cluster le plus proche.
   (b) Calculer les nouveaux centres des classes.

---

3 Expérimentations

3.1 Description du jeu de données

Dans le cadre de la nouvelle politique "Open Data" lancée par la ville de Paris, des données de trajets Vélib ont été mises à la disposition des scientifiques afin de promouvoir la recherche et l'innovation au service des usagers. Un trajet est caractérisé par un timestamp de départ,

un timestamp d'arrivée, une station de départ et une station d'arrivée. L'analyse de quelques mois de trajets (avril, juin et juillet 2013) a montré que les trajets peuvent être divisés en deux grandes catégories : les jours travaillés et le week-end. Deux jours issus d'une même catégorie se ressemblent beaucoup. Les quelques différences notées sont dues aux conditions climatiques ou à des évènements rares comme les jours fériés ou les grèves. Les profils des trajets étant très différents entre les jours travaillés et le week-end, nous nous focalisons dans cet article sur les jours travaillés. Nous avons choisi d'analyser 24 heures de trajets : les trajets qui ont eu lieu à la date du vendredi 28 juin 2013. A cette date 121709 trajets ont été effectués, impliquant 1225 stations Vélib. Ces trajets comprennent $1,03\%$ de trajets maintenance et $1,48\%$ de trajets de régulation.

3.2 Classification en clusters

Le but est d'utiliser l'algorithme Kmeans présenté dans la section 2 afin de répartir les stations Vélib en plusieurs classes et mieux comprendre le besoin des usagers. Un prétraitement des données de trajets a été effectué afin de générer les séries temporelles relatives à chaque station. Seuls les vrais trajets des usagers, soit les trajets excluant maintenance et régulation, ont été pris en compte. Nous nous intéressons particulièrement dans cette étude aux stations problématiques qui sont souvent vides ou souvent saturées. Les données de trajets ne contiennent pas d'information sur l'état de remplissage de la station. Pour une station i, on note c_i sa capacité, définie comme son nombre total de bornes, et on introduit sa série temporelle des trajets donnée par la relation suivante :

$$T_{i,t} = \frac{a_{i,t} - d_{i,t}}{c_i}$$

où $a_{i,t}$ désigne le nombre d'arrivées cumulées à la station i (entre 0h et t) et $d_{i,t}$ est le nombre de départs cumulées de la station i. On normalise par c_i, pour compenser l'hétérogénéité des stations due à une différence de capacité ($c_i \in [8, 72]$). $T_{i,t}$ est générée à partir des trajets pour chaque station i, toutes les 10 minutes sur une durée totale de 24 heures.

En l'absence de régulation, on a $T_{i,t} \in [-1, 1]$, car la différence entre le nombre d'arrivées et de départs est toujours bornée en valeur absolue par la capacité de la station. Même si les trajets de maintenance et de régulation ne sont pas pris en compte pour le calcul de $T_{i,t}$, leur impact n'est pas complètement annulé. En effet c'est grâce aux vélos déplacés par les camions vers une station i par exemple que de nouveaux trajets d'usagers au départ de cette station i ont pu avoir lieu. Ces vrais trajets sont intégrés dans le calcul de $T_{i,t}$, ce qui explique le fait que $T_{i,t}$ a atteint $1,5$ et $-1,5$ pour quelques stations i.

L'algorithme Kmeans a été appliqué sur les 1225 séries temporelles ($T_{i,t}$), contenant chacune 144 valeurs. Le nombre de clusters K a été fixé à 6, et la distance utilisée est la distance euclidienne. Kmeans répartit les stations Vélib en 6 clusters décrits dans le tableau suivant :

Cluster	Mixte	Emploi	Périphérie	Habitation	Divertissement	Gares
Taille	500	177	177	169	115	87

TAB. 1 – *Taille des 6 clusters obtenus par Kmeans*

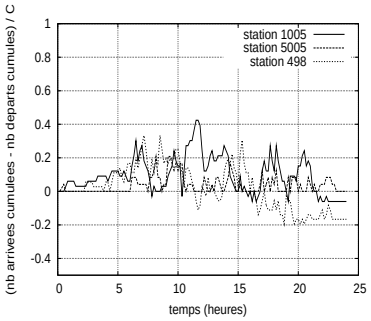


FIG. 1 – *Cluster "Mixte"*

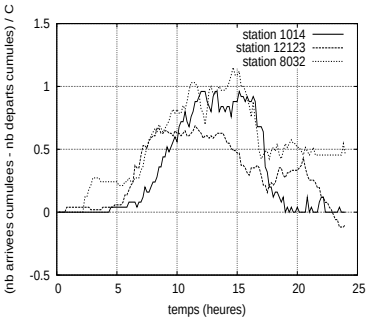


FIG. 2 – *Cluster "Emploi"*

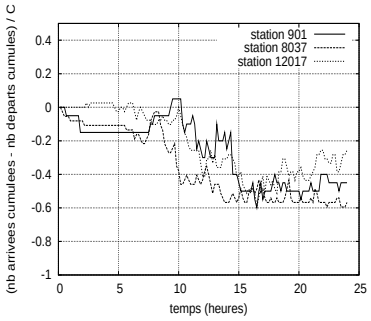


FIG. 3 – *Cluster "Périphérie"*

Au démarrage de l'algorithme, les centres des clusters contiennent 6 stations choisies au hasard parmi les stations Vélib. Nous avons exécuté l'algorithme 20 fois et nous avons vérifié que le contenu des 6 clusters varie très peu. Nous avons obtenu en moyenne $26,35$ itérations et un temps moyen d'exécution de seulement 585 ms. Le nombre de clusters a été fixé à posteriori : au delà de 6 clusters, le nombre de stations par cluster devient non significatif.

Les résultats obtenus sont très similaires à ceux présentés dans Côme et Oukhellou (2012). Ils seront détaillés dans ce qui suit :

Le plus grand cluster appelé "Mixte" contient presque 40% des stations. La distribution géographique des stations issues de ce cluster est uniforme. La Figure 1 montre les séries temporelles $T_{i,t}$ de 3 stations choisies au hasard dans ce cluster. On voit clairement que ces stations sont équilibrées du point de vue arrivées et départs. Ainsi la plupart des stations Vélib' ne nécessitent pas de régulation.

Le second cluster "Emploi" correspond aux zones d'emploi. Les stations de ce cluster sont surtout concentrées dans les 2ème et 8ème arrondissements de Paris, sur les quais de la Seine, à Vincennes et à Boulogne. Les trajets relatifs à ce cluster sont illustrés par la Figure 2, ils sont

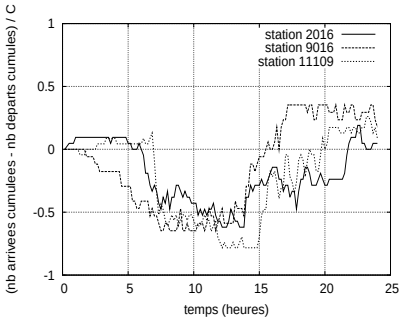


FIG. 4 – *Cluster "Habitation"*

cadencés par les horaires de travail. Globalement sur ces stations on a beaucoup d'arrivées entre 6h et 10h du matin et beaucoup de départs entre 15h et 18h. Des actions de régulation sont utiles surtout le matin pour libérer des places.

Les stations du cluster "Périphérie" sont majoritairement situées à la périphérie de Paris vers les terminus des lignes de métro. La répartition géographique de ce cluster est illustrée par la Figure 3. Les départs et les arrivées sont équivalents sauf entre 10h et 15h où on a beaucoup plus de départs que d'arrivées. Les déplacements aux départs de ces stations entre 10h et 15h se dirigent essentiellement vers le centre de Paris, plus précisément vers les zones de divertissement ou de travail. Contrairement à tous les autres clusters, ce cluster "Périphérie" n'apparait pas dans l'étude de Côme et Oukhellou (2012).

Le quatrième cluster correspond aux zones d'habitation (11ème, 15ème, 17ème, 18ème et 20ème arrondissement de Paris). Inversement au cluster "Emploi", dans ce cluster on a beaucoup de départs entre 5h et 10h du matin et beaucoup d'arrivées entre 15h et 18h. Le déséquilibre de ce cluster semble moins intense que celui du cluster "Emploi", en effet la différence entre les arrivées et les départs dépasse rarement la moitié de la capacité de la station (voir Figure 4). Ceci peut être expliqué par le fait que la zone d'habitation est géographiquement plus étendue que la zone d'emploi.

Le 5ème cluster "Divertissement" concerne essentiellement le 2ème, 3ème et 5ème arrondissement. La Figure 5 montre qu'il y a une affluence vers ces stations (des arrivées nettement supérieures aux départs) entre 15h et 19h.

Le dernier cluster "Gares" est constitué de stations à proximité des grandes gares (gare Montparnasse, gare de l'Est, gare du Nord, gare Saint-Lazare et Denfert Rochereau). D'après la Figure 6 ces stations sont souvent vides malgré les efforts de régulation.

3.3 Stations problématiques

Les stations problématiques sont les stations souvent vides ou souvent pleines. Comme on ne dispose pas d'information sur le taux de remplissage des stations, on va définir les stations saturées par les stations dont la série temporelle $T_{i,t}$ atteint $0,9$ à un instant t de la journée. Quand la différence entre arrivées et les départs atteint 90% de la capacité de la station, il suffit

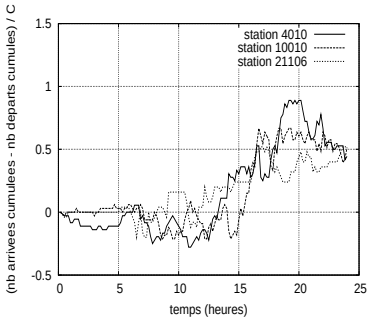


FIG. 5 – *Cluster "Divertissement"*

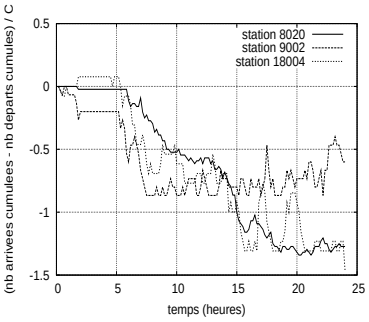


FIG. 6 – *Cluster "Gares"*

d'avoir un taux de remplissage initial égal à 10% pour que cette station soit complètement pleine. On ne tient pas compte ici des places libérées grâce aux actions de régulation. Les stations vides sont définies d'une manière analogue.

En respectant ces définitions, les stations problématiques sont représentées dans les Figures 7 et 8. On a en tout 73 stations vides. Elles appartiennent essentiellement aux clusters "Habitation" (30%) et "Gares" (65%). Les stations saturées sont situées au centre de Paris et sur les quais de la Seine. 50% de ces stations appartiennent au cluster "Divertissement", et 40% au cluster "Emploi" pour un nombre total de 66 stations.

FIG. 7 – *Stations vides*

FIG. 8 – *Stations pleines*

4 Conclusion

Cette étude donne une description qualitative et quantitative de l'utilisation actuelle du système Vélib. La répartition des stations en clusters permet d'avoir une idée sur les flux des trajets des usagers. Les clusters au nombre de 6 sont facilement interprétables. L'étude des stations problématiques est utile pour localiser les ressources critiques du système Vélib. La

connaissance de ces clusters est indispensable à la modélisation du système afin d'étudier ou de simuler si par exemple la modification de la cartographie de Vélib (ouverture de nouvelles stations, ajout de nouvelles ressources) est suffisante pour surmonter ou du moins atténuer ce phénomène de stations problématiques. Nos futurs travaux auront pour but de répondre aux questions suivantes : Quel est l'impact d'une augmentation de ressources (nombre de bornes, nombre de vélos) ? Quel est l'impact de la régulation ? Par quelles méthodes incitatives les utilisateurs peuvent-ils être amenés à mieux répartir les vélos entre les stations ?

Remerciements

Nous tenons à remercier JCDecaux et la mairie de Paris pour l'accès aux données de trajets de Vélib.

Références

Borgnat, P., P. Abry, P. Flandrin, C. Robardet, J.-B. Rouquier, et E. Fleury (2011). Shared bicycles in a city : A signal processing and data analysis perspective. *Advances in Complex Systems 14*(03), 415–438.

Chemla, D., F. Meunier, et R. Wolfler Calvo (2012). Bike sharing systems : Solving the static rebalancing problem. *Discrete Optimization*.

Côme, E. et L. Oukhellou (2012). Model-based count series clustering for bike-sharing system usage mining, a case study with the vélib' system of paris. *JACM-TIST Special Issue Urban computing*.

Fricker, C. et N. Gast (2014). Incentives and regulations in bike-sharing systems with stations of finite capacity, special issue: Shared mobility systems. *EURO Journal on Transportation and Logistics*.

Fricker, C., N. Gast, et H. Mohamed (2012). Mean field analysis for inhomogeneous bike sharing systems. *AofA 2012, International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms*.

Froehlich, J., J. Neumann, et N. Oliver (2009). Sensing and predicting the pulse of the city through shared bicycling. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, San Francisco, CA, USA, pp. 1420–1426. Morgan Kaufmann Publishers Inc.

George, D. K. et C. H. Xia (2010). Asymptotic analysis of closed queueing networks and its implications to achievable service levels. *SIGMETRICS Performance Evaluation Review 38*(2), 3–5.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *5-th Berkeley Symposium on Mathematical Statistics and Probability 1*, 281–297.

Nair, R., E. Miller-Hooks, R. C. Hampshire, et A. Bušić (2013). Large-scale vehicle sharing systems: Analysis of vélib'. *International Journal of Sustainable Transportation 7*(1), 85–106.

Pakhira, M. (2009). A modified k-means algorithm to avoid empty clusters. *International Journal of Recent Trends in Engineering 1*, 220–226.

Pfrommer, J., J. Warrington, G. Schildbach, et M. Morari (2013). Dynamic vehicle redistribution and online price incentives in shared mobility systems. *CoRR abs/1304.3949*.

Raviv, T., M. Tzur, et I. Forma (2012). Static repositioning in a bike-sharing system: Models and solution approaches. *Euro Journal of Transportation and Logistics*.

Schuijbroek, J., R. Hampshire, et W.-J. van Hoeve (2013). Inventory rebalancing and vehicle routing in bike sharing systems. Technical report, Schuijbroek.

Summary

The Bike Sharing System becomes recently a popular mode of transport in Paris and in many other big cities all over the world. One of the main challenges of this sharing system management is the non availability of bikes or docks in some attractive stations. In this paper, we analyze the Vélib trips information in order to classify stations into several categories based on their varying involvement in the trips during a whole day. For this purpose, we apply the well known clustering algorithm called "Kmeans". The goal of this study is to locate the problematic stations that are often empty or often saturated. Such information can be then used to propose new methods to improve the availability of the Vélib System.

Analyse visuelle de la co-évolution des termes dans les thématiques Twitter

Lambert Pépin*,**, Julien Blanchard**
Fabrice Guillet**, Pascale Kuntz**, Philippe Suignard*

*EDF R&D, Clamart, France
prenom.nom@edf.fr
**Equipe COD-LINA (UMR CNRS 6241), Université de Nantes, France
prenom.nom@univ-nantes.fr

Résumé. Twitter offre une vision directe des attentes de ses utilisateurs, et est devenu une source d'informations incontournable dans le cadre de la gestion de la relation client. Cependant, les méthodes d'analyse de texte produisent souvent de grandes tendances qui ne présentent pas de valeurs ajoutées pour des experts qui sont à la recherche d'informations plus fines. Dans cet article, nous proposons une méthode basée sur l'analyse de la co-évolution des termes utilisés au sein d'une thématique Twitter ciblée afin de faciliter leur analyse par un décideur. Nous réordonnons les termes afin de mettre en avant leur co-évolution, et nous utilisons une carte de chaleur afin de visualiser un grand nombre de distributions. Une étude a été menée avec un expert du groupe Électricité de France (EDF), et les premiers résultats permettent de détecter plusieurs comportements : les pics, les répétitions et les longues périodes d'activité.

1 Introduction

Les volumes croissants de données liés à l'usage des réseaux sociaux, et de Twitter en particulier, atteignent des ordres de grandeur toujours plus élevés et représentent pour toute entreprise une source d'informations stratégique dans un contexte d'amélioration de sa relation client. Les messages postés sur Twitter (*tweets*) ont la particularité d'être à la fois textuels, non-structurés et horodatés ce qui explique l'intérêt suscité par des domaines comme la modélisation de thématiques (*topic modelling* - Blei et Lafferty (2006)) et le suivi de thématiques (*news tracking* - Leskovec et al. (2009)). Cependant, modéliser l'information partagée sur Twitter demeure une question ouverte ; principalement à cause de fortes contraintes syntaxiques (limite de taille, idiomes) et de l'influence du facteur nouveauté. En effet, des études récentes montrent que la plupart des thématiques ont une durée de vie courte "73% des sujets ne sont abordés que pendant une seule période et dans 31% des cas cette période ne dure qu'une journée" (Kwak et al. (2010)), et qu'un grand nombre de mots nouveaux sont employés chaque jour alors qu'ils n'avaient jamais été utilisés avant. On observe sur la Figure 1, que même si la tendance générale est à la baisse, le vocabulaire continue de se renouveler au cours du temps. Cette volatilité est contradictoire avec le besoin d'analyse sur le long terme de la relation client

ciblée sur un groupe de thématiques liée à l'entreprise : évolution des sujets traités dans les tweets sur un mois, sur un an, répétitions, etc. L'application de méthodes de partitionnement à l'échelle des termes permet de faire émerger des concepts aux comportements variés (pics, ré-occurrences, réguliers) et d'un niveau de granularité plus fin que les catégories construites à l'échelle des documents.

Dans cet article, nous nous intéressons à la visualisation de la co-évolution des termes dans les messages publiés sur Twitter, sur une thématique ciblée, dans le but d'aider un spécialiste de la gestion de la relation client à comprendre l'évolution des centres d'intérêt des utilisateurs. Nous proposons une méthodologie en 3 étapes pour l'analyse de la co-évolution : premièrement nous définissons une mesure d'intérêt pour chaque terme (un score évoluant au cours du temps), puis nous regroupons les termes qui partagent les mêmes tendances durant la période étudiée, enfin nous visualisons l'évolution de leurs scores à l'aide d'une carte de chaleur. Notre approche a été testée sur un jeu de données réelles provenant de la société EDF et a permis à un expert de détecter plusieurs types de thématiques. La suite de l'article est organisée de la façon suivante : la section 2 positionne nos travaux par rapport aux méthodes d'analyse de thématiques et aux techniques de visualisation appliquées à Twitter. La section 3 expose notre méthodologie d'analyse de la co-évolution. La section 4 détaille notre protocole expérimental et les résultats obtenus en partenariat avec un expert de la gestion de la relation client. La section 5 conclut et présente les perspectives.

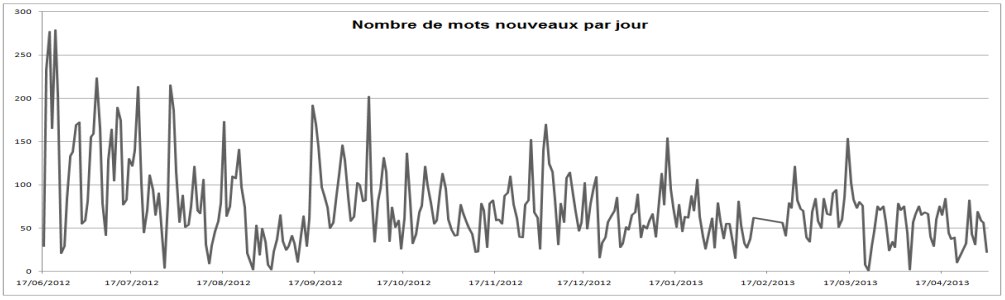


FIG. 1 – *Nombre de mots nouveaux, jamais employés les jours précédents dans notre corpus Twitter.*

2 Positionnement

La modélisation de thématiques cherche à structurer des données textuelles en analysant la sémantique sous-jacente des documents. De nombreux travaux (Kasiviswanathan et al. (2011), Caballero et al. (2012), Jo et al. (2011)) ont été publiés à ce sujet, dans des domaines allant de l'analyse statistique avec Tf-Idf (Jones (1972)) et Okapi-BM25 (Robertson et al. (1999)) aux modèles probabilistes avec l'analyse sémantique latente probabiliste (Hofmann (1999)) et l'allocation de Dirichlet latente (LDA)(Blei et al. (2003)) en passant par l'algèbre linéaire avec l'analyse sémantique latente (Deerwester et al. (1990)). Notre méthodologie propose une approche complémentaire qui permet à la fois de modéliser les thématiques à travers l'évolution,

dans le temps, des termes qui les composent mais également de tirer parti des méthodes précédentes en intégrant leurs résultats dans la chaîne de traitement afin d'en visualiser l'évolution.

Le domaine de la détection d'événements, et plus particulièrement de la détection de pics (Kleinberg (2002)), propose des méthodes qui prennent en compte la dimension temporelle dans l'analyse de données horodatées. Ces méthodes appliquent des techniques issues de l'analyse des séries temporelles pour définir les limites des sujets abordés (périodes, auteurs, zones géographiques, etc.) (Marcus et al. (2011)). En proposant une méthode d'analyse visuelle, nous permettons à notre expert de détecter une plus grande variété de motifs (termes rares et fréquents, ré-occurrence, périodicité, etc.).

Le domaine du suivi de thématiques s'intéresse à la dynamique des thématiques et cherche à fournir une vision globale de leur évolution au cours du temps. Certaines extensions de LDA sont dynamiques et proposent de suivre les transitions des *topics* détectés (Hoffman et al. (2010)). Les algorithmes de partitionnement de graphes sont également utilisés pour construire des groupes de termes et visualiser leur évolution (Leskovec et al. (2009)). Les algorithmes de partitionnement et de positionnement dynamiques peuvent également être utilisés pour construire et mettre à jour des visualisations qui préservent la carte mentale de l'utilisateur (Gansner et al. (2012)).

Stimulées par le nombre de données disponibles, de nombreuses méthodes de visualisations ont été développées pour tirer parti de la composante temporelle (voir Miksch et Schumann (2011) pour une synthèse récente). Si les premières méthodes se concentraient sur la visualisation de séries temporelles et ne prenaient pas en compte la sémantique, certaines plus récentes on été développées pour visualiser l'évolution des thématiques. Toutefois, ces dernières ne prennent pas en compte explicitement la co-évolution. Par exemple, Leskovec et al. (2009) utilisent des aires empilées pour visualiser plusieurs distributions sur un même graphique, mais si elles sont adaptées pour visualiser l'évolution d'une composante par rapport au comportement global, elles sont moins efficaces pour mettre en avant la co-évolution de différentes composantes. Mei et Zhai (2005) et Jo et al. (2011) utilisent des graphes d'évolution de thématiques, mais comme le comportement de ces dernières n'est pas toujours linéaire (certaines thématiques se séparent en deux, d'autre fusionnent), ces graphes ne sont pas adaptés à la visualisation de la co-évolution. Nous proposons dans cet article une représentation basée sur les cartes de chaleur qui met en avant la co-évolution des termes en s'appuyant sur une étape de partitionnement. Le partitionnement sur cartes de chaleur a montré son efficacité dans le domaine de la bio-informatique : "elles sont une des visualisations les plus populaires pour les données génétiques" (North et al. (2005)). Et pourtant, d'après nos connaissances, elles n'ont jamais été étudiées dans le cadre de l'analyse des réseaux sociaux.

3 Cadre expérimental

Notre méthodologie propose de regrouper les termes présentant des comportements similaires pour visualiser leur évolution, et ainsi mettre en avant des motifs de co-évolution. La chaîne de traitement est représentée sur la Figure 2 et peut être résumée ainsi : tout d'abord, nous divisons la période étudiée en sous-périodes de taille fixe et nous sélectionnons les termes à comparer. Puis, pour chaque terme, nous construisons un vecteur, appelé "vecteur d'évolution", de longueur égale au nombre de sous-périodes et contenant les valeurs de la fonction de score pour chaque période. Ensuite, nous calculons la matrice de co-évolution qui représente la

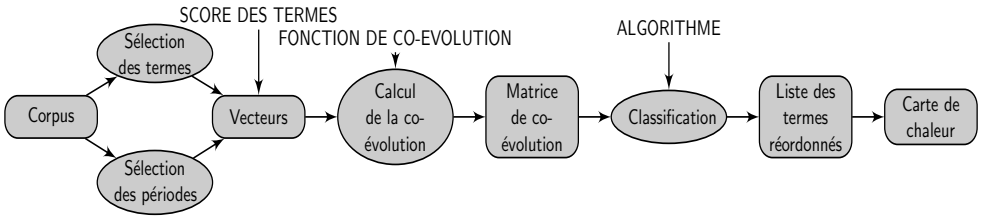


FIG. 2 – *Résumé de la méthodologie et des différents paramètres.*

similarité temporelle de chaque paire de termes. Un algorithme de classification hiérarchique est alors appliqué à la matrice de co-évolution pour regrouper les termes qui partagent un comportement commun. Finalement, nous retenons un ordre compatible avec le dendrogramme fourni par cet algorithme pour classer les vecteurs d'évolution des termes et les visualiser sur une carte de chaleur.

3.1 Sélection des périodes et des termes

Pour cette expérimentation, nous avons collecté tous les messages en français publiés sur Twitter et contenant le terme "edf", pour Électricité de France, entre le 17 Juin 2012 et le 02 Mai 2013. Sur ce réseau social où la taille des messages est limitée à 140 caractères, le terme "edf" est également utilisé pour faire référence à l'équipe de France (de Football, etc.). Une étape préliminaire est donc nécessaire pour filtrer les tweets faisant référence au sport et non à l'entreprise. Puisque le vocabulaire employé est en perpétuelle évolution, cette étape constitue une problématique à part entière et ne sera pas détaillée ici. Après cette étape de filtrage, nous disposons d'un corpus $\mathcal{T}$ correspondant à 73 023 tweets faisant référence à l'entreprise.

Chaque tweet est segmenté (*tokenized*) de telle sorte que chaque segment soit la plus longue chaîne possible de caractères de mot. Les mentions, les URL's, le symbole "RT" (pour retweet) et les accents sont retirés. L'outil TreeTagger [1] est utilisé pour obtenir la forme lemmatisée des termes puis ceux-ci sont passés en minuscule. Nous construisons le vocabulaire $\mathcal{W}$ en retenant les 500 paires de termes les plus fréquentes.

En parallèle, nous divisons la période totale en 307 sous-périodes d'une durée de 24h, $\{t_1, ..., t_{307}\}$. En effet, en calculant le nombre de tweets publiés chaque heure, nous avons observé que l'activité sur le réseau est très faible la nuit. Le corpus est donc découpé en période de 24h s'étendant de 4h du matin à 3h59. Nous construisons les 307 sous-ensembles de documents correspondant $\{\mathcal{T}_1, ..., \mathcal{T}_{307}\}$, tels que $\forall i$, $\mathcal{T}_i$ contient tous les tweets publiés pendant la période i.

3.2 Fonction de score

Nous définissons ensuite la mesure d'intérêt utilisée pour représenter le comportement des termes au cours du temps. Comme nous nous intéressons à l'évolution de la popularité des thématiques sur Twitter, nous définissons le score de chaque terme $w \in \mathcal{W}$ pour chaque période comme le nombre de documents contenant w publiés pendant cette période.

1. http ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

Nous définissons la fonction de score s telle que,

$$\begin{array}{rccc} s: & \mathcal{W} \times \{1,\ldots,n\} & \rightarrow \mathbb{R} \\ & w \times i & \mapsto N(w,i) \end{array} \quad (1)$$

où $N(w,i)$ est le nombre de documents contenant le terme w publiés pendant la période t_i. Le choix de la fonction de score dépend de l'application et de la nature des données ; elle peut être, par exemple, une valeur binaire correspondant au critère présence/absence ou une normalisation de type tf-idf.

3.3 Co-évolution des termes

Pour chaque terme $w \in \mathcal{W}$, nous définissons $\vec{ev}(w)$, le vecteur d'évolution de w tel que, $\forall i \in \{1,\ldots,n\}$ $e\vec{v}_i(w) = s(w,i)$. Pour chaque paire de termes (w_1, w_2), la co-évolution de w_1 et w_2 est définie comme la co-évolution de leurs vecteurs d'évolution $(\vec{ev}(w_1), \vec{ev}(w_2))$. Dans cette étude, nous calculons la corrélation de chaque paire de termes (w_1, w_2) à l'aide du coefficient des rangs de *Kendall*. Contrairement au coefficient de *Pearson* et à celui de *Spearman*, le coefficient de *Kendall* présente l'avantage de, non seulement, détecter les relations linéaires entre termes, mais également, les relations non linéaires. En fonction des buts de l'analyste, la fonction de co-évolution peut également être une mesure de similarité, comme la similarité cosinus.

Definition 1 *Paires de coordonnées concordantes*

Soit $(i,j) \in \{1,\ldots,n\}^2$ une paire de coordonnées, on dit que (i,j) est une paires de coordonnées concordantes entre les vecteurs $\vec{ev}(w_1)$ et $\vec{ev}(w_2)$, si et seulement si, $e\vec{v}_i(w_1) < e\vec{v}_i(w_2)$ et $e\vec{v}_j(w_1) < e\vec{v}_j(w_2)$, ou $e\vec{v}_i(w_1) > e\vec{v}_i(w_2)$ et $e\vec{v}_j(w_1) > e\vec{v}_j(w_2)$.

Definition 2 *Coefficient de Kendall*

Le coefficient de *Kendall* τ, des vecteurs $\vec{ev}(w_1)$ et $\vec{ev}(w_2)$ est défini par :

$$\tau(\vec{ev}(w_1), \vec{ev}(w_2)) = \frac{N_c - N_d}{\frac{1}{2}n(n-1)} \quad (2)$$

où N_c est le nombre de paires de coordonnées concordantes entre $\vec{ev}(w_1)$ et $\vec{ev}(w_2)$ et N_d est le nombre de coordonnées discordantes. Le dénominateur correspond au nombre total de combinaisons.

Le résultat du calcul de la co-évolution est une matrice carrée $\mathcal{M}_S$, de taille $|\mathcal{W}|$, appelée matrice de co-évolution. Les lignes et les colonnes de $\mathcal{M}_S$ correspondent aux termes ordonnés de la même manière et $\forall i,j \in \{1,\ldots,|\mathcal{W}|\}$,

$$\mathcal{M}_S(i,j) = \tau(\vec{ev}(w_i), \vec{ev}(w_j)). \quad (3)$$

A cette étape, l'ordre dans lequel les termes sont présentés dans $\mathcal{M}_S$ ne correspond à aucun comportement particulier. Selon la méthode utilisée pour construire $\mathcal{W}$, il peut être par exemple l'ordre alphabétique ou l'ordre dans lequel les termes apparaissent dans les documents. L'objectif de cette étape est de réorganiser les termes de $\mathcal{W}$ en fonction de leurs co-évolutions.

3.4 Classification

Nous appliquons un algorithme de classification ascendante hiérarchique (CAH) sur $\mathcal{M}_S$ afin de construire des groupes de termes similaires. A chaque étape de la classification, les deux classes les plus proches sont regroupées selon le critère du saut maximum qui produit des classes compactes et est moins couteux en temps de calcul que le saut moyen. Pour chaque classe C_1, C_2,

$$sim(C_1, C_2) = \min_{x \in C_1, y \in C_2} (sim(x, y)).$$

Nous utiliserons cette classification pour visualiser les termes similaires proches les uns des autres.

3.5 Carte de chaleur

Déterminer l'ordre optimal à utiliser pour afficher le résultat d'une classification ascendante hiérarchique, c'est à dire celui minimisant la somme des distances entre instances adjacentes, est très couteux en temps de calcul (Bar-Joseph et al. (2001)). Or, la tache proposée dans cet article consiste uniquement à détecter des groupes de termes évoluant conjointement. Un ordre compatible avec le résultat de la classification suffit à mettre en avant les motifs recherchés. Cet ordre est utilisé pour construire une carte de chaleur représentant l'évolution des scores des termes au cours du temps. Les colonnes de cette carte de chaleur correspondent aux n périodes affichées dans l'ordre chronologique, les lignes correspondent aux $|\mathcal{W}|$ termes réordonnés par la CAH et le gradient de couleur est défini pour correspondre aux scores fournis par la fonction de score pour chaque terme et chaque période.

4 Résultats

Cette expérimentation a été conduite en partenariat avec un expert du traitement du langage naturel et de la gestion de la relation client du groupe EDF. Nous lui avons demandé de détecter des groupes de termes à l'aide de la carte de chaleur construite par notre méthodologie.

Dans un souci de place, nous présentons un extrait de la carte de chaleur visualisée par l'expert, construit pour un nombre restreint de termes et centré sur leur période d'activité (voir Figure 3). Cet extrait permet de visualiser trois types de comportements distincts détectés par notre expert.

Un premier groupe de termes correspond aux thématiques courtes décrites par Kwak et al. (2010), ces termes saillants n'apparaissent que quelques jours (autour du 17 Août 2012) et ne sont plus utilisés ensuite. Ils correspondent à des messages rapportant une rumeur de remplacement à la tête de la direction d'EDF du président Henri Proglio par Guillaume Pepy, actuellement directeur de la SNCF. Cette rumeur a été relayée sur les médias traditionnels et n'est pas une nouveauté pour notre expert, toutefois, savoir qu'elle a été discutée sur Twitter et pouvoir estimer dans quelles proportions, reste à ces yeux une information utile.

Le second groupe de termes présente une activité qui s'étend sur une période plus longue et n'aurait pas nécessairement été détecté par une méthode de détection de pics. Ces termes sont portés par un volume de messages plus important dans lesquels les utilisateurs de Twitter

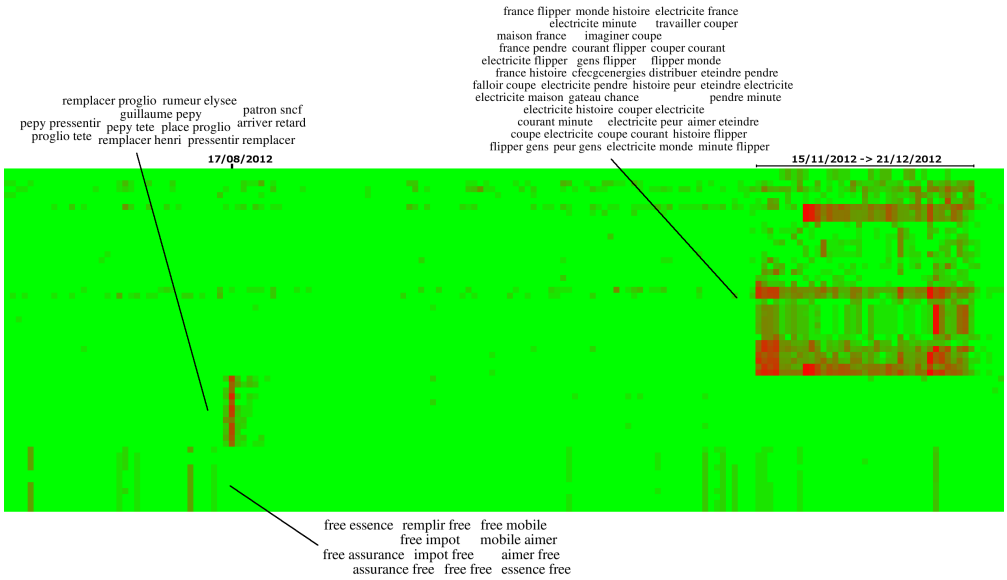


FIG. 3 – *Extrait de la carte de chaleur présentée à l'expert, pour un nombre réduit de termes, centrée sur leur période d'activité.*

s'amusent des prédictions de fin du monde qui ont animé cette fin d'année 2012 en proposant que l'électricien coupe le courant pendant 10 minutes pour faire "flipper les gens".

Enfin, le troisième groupe correspond à des termes qui sont employés ensemble à plusieurs reprises durant la période considérée, créant un motif de répétition. Ces termes correspondent à un ensemble de tweets dans lesquels les utilisateurs demandent, sur le ton de l'humour, à ce que les tarifs appliqués par l'entreprise Free dans la téléphonie le soient également dans d'autres domaines, en particulier dans celui de l'électricité. Notre expert porte un intérêt à ces messages qui font parti des signaux faibles liés aux tarifs de l'énergie. Le fait que ce sujet soit répété à plusieurs reprises, contrairement à la blague sur la fin du monde, montre qu'il n'est pas anecdotique. Même si le ton est léger, ce sujet intéresse les utilisateurs du réseau.

5 Conclusion

Dans cet article, nous avons présenté une méthodologie visant à mettre en avant des motifs de co-évolution dans les termes employés sur Twitter à l'aide d'un support visuel. Une expérimentation basée sur des données réelles nous a permis d'identifier plusieurs classes de termes associées à des comportements caractéristiques. Cette première étude vise à démontrer la faisabilité de notre méthodologie. Nous avons présenté dans cet article, une analyse conduite avec un seul expert, et même si les résultats sont encourageants, nous sommes confrontés à un problème d'évaluation de la qualité de notre visualisation. Cette question est récurrente en visualisation analytique (Jankun-Kelly et al. (2007)) mais la classification sur carte de chaleur est une technique de visualisation qui a fait ses preuves dans d'autres champs de recherche.

Par ailleurs, nos perspectives de travail viseront également à comparer des fonctions de co-évolution et des algorithmes de classification, ainsi que des méthodes permettant de calculer la co-évolution des termes sur une période plus courte.

Références

Bar-Joseph, Z., D. K. Gifford, et T. S. Jaakkola (2001). Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics 17*(suppl 1), S22–S29.

Blei, D. M. et J. D. Lafferty (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pp. 113–120. ACM.

Blei, D. M., A. Y. Ng, et M. I. Jordan (2003). Latent dirichlet allocation. *J. Mach. Learn. Res. 3*, 993–1022.

Caballero, K. L., J. Barajas, et R. Akella (2012). The generalized dirichlet distribution in enhanced topic detection. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, New York, NY, USA, pp. 773–782. ACM.

Deerwester, S. C., S. T. Dumais, T. K. Landauer, G. W. Furnas, et R. A. Harshman (1990). Indexing by latent semantic analysis. *JASIS 41*(6), 391–407.

Gansner, E., Y. Hu, et S. North (2012). Visualizing streaming text data with dynamic maps. *ArXiv e-prints*.

Hoffman, M., F. R. Bach, et D. M. Blei (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pp. 856–864.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, UAI'99, San Francisco, CA, USA, pp. 289–296. Morgan Kaufmann Publishers Inc.

Jankun-Kelly, T. J., K.-L. Ma, et M. Gertz (2007). A model and framework for visualization exploration. *Visualization and Computer Graphics, IEEE Transactions on 13*(2), 357–369.

Jo, Y., J. E. Hopcroft, et C. Lagoze (2011). The web of topics : discovering the topology of topic evolution in a corpus. In *Proceedings of the 20th international conference on World wide web*, pp. 257–266. ACM.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation 28*(1), 11–21.

Kasiviswanathan, S. P., P. Melville, A. Banerjee, et V. Sindhwani (2011). Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 745–754. ACM.

Kleinberg, J. (2002). Bursty and hierarchical structure in streams. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, New York, NY, USA, pp. 91–101. ACM.

Kwak, H., C. Lee, H. Park, et S. Moon (2010). What is twitter, a social network or a news media ? In *Proceedings of the 19th international conference on World wide web*, pp. 591–600. ACM.

Leskovec, J., L. Backstrom, et J. Kleinberg (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 497–506. ACM.

Marcus, A., M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, et R. C. Miller (2011). Twitinfo : aggregating and visualizing microblogs for event exploration. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, New York, NY, USA, pp. 227–236. ACM.

Mei, Q. et C. Zhai (2005). Discovering evolutionary theme patterns from text : an exploration of temporal text mining. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, New York, NY, USA, pp. 198–207. ACM.

Miksch, S. et H. Schumann (2011). *Visualization of time-oriented data*. Springerverlag London Limited.

North, C., T.-M. Rhyne, et K. Duca (2005). Bioinformatics visualization : introduction to the special issue. *Information Visualization 4*(3), 147–148.

Robertson, S. E., S. Walker, M. Beaulieu, et P. Willett (1999). Okapi at trec-7 : automatic ad hoc, filtering, vlc and interactive track. *Nist Special Publication SP*, 253–264.

Summary

The analysis of Twitter short messages has become a key issue in companies to understand consumer behaviour and expectations. However, automatic algorithms for text clustering often extract general tendencies at a high granularity level and do not provide added value to experts who are looking for more subtle information. In this paper, we focus on the visualization of the co-evolution of terms in tweets in order to facilitate the analysis of the topic evolution by a decision-maker. We take advantage of the perceptual quality of heatmaps to display our 3D data (term $\times$ time $\times$ score) in a 2D space. And, by computing an appropriate order to display the main terms on the heatmap, our methodology ensures an intuitive visualization of their co-evolution. An experimentation has been conducted on real-life data sets in collaboration with an expert of customer relationship management working at the French energy company EDF. The first results show three different kinds of co-evolution of terms: bursty features, reoccurring terms and periods of activity.

Une méthode basée sur des effectifs pour calculer la contribution des variables à un clustering

Oumaima Alaoui Ismaili*,***, Julien Salotti**, Vincent Lemaire***

*AgroParisTech 16, rue Claude Bernard 75005 Paris
**INSA Lyon 20, avenue Albert Einstein - 69621 Villeurbanne
***Orange Labs 2 avenue Pierre Marzin 22300 Lannion

Résumé. Cet article présente une étude préliminaire effectuée dans un contexte industriel. On dispose d'une typologie de clients que le service marketing souhaite contacter. Cette typologie est une segmentation des clients en groupes de clients dont les profils seront utilisés pour proposer des campagnes marketing différenciées. La constitution des groupes est réalisée à l'aide d'une technique de clustering qui ne permet pas actuellement de connaître l'importance des variables explicatives (qui décrivent les clients). Cet article propose de résoudre ce problème à l'aide d'une méthodologie qui donne dans notre contexte industriel, l'importance des variables explicatives. Cette méthode sera comparée à certaines méthodes de l'état de l'art.

1 Introduction

Lorsqu'on désire contacter un client pour lui proposer un produit on calcule au préalable la probabilité qu'il achète ce produit. Cette probabilité est calculée à l'aide d'un modèle prédictif pour un ensemble de clients. Le service marketing contacte ensuite ceux ayant les plus fortes probabilités d'acheter le produit. En parallèle, et avant le contact commercial, on réalise une typologie des clients auxquels on propose des campagnes différenciées par groupes. Plus formellement, le problème est celui du clustering supervisé, où un clustering est appliqué sur des données étiquetées. Ce problème peut être défini comme étant un processus de regroupement des individus en clusters, tels que les données de chaque cluster soient les plus similaires possibles et appartiennent à la même classe à prédire. Le lecteur pourra trouver une description détaillée de ce problème dans l'article (Lemaire et al., 2012).

Actuellement, la technique de clustering utilisée pour la constitution des groupes ne permet pas d'identifier les variables les plus importantes. Autrement dit, cette technique ne permet pas de connaître les variables qui contribuent le plus lors de la construction des clusters. Par conséquent, le service marketing éprouve des difficultés à adapter sa campagne aux différents profils identifiés. L'objectif de cette étude est donc de proposer une méthode qui permet de mesurer l'importance des variables à la fin de la convergence d'un clustering. Cette dernière doit prendre en compte trois points principaux :

1. Conserver toutes les variables utilisées lors du clustering.
2. Ne pas réapprendre le modèle.

3. Garder l'espace de représentation des données utilisé lors de la phase de prétraitement (discrétisation pour les variables continues et groupage des valeurs pour les variables catégorielles) qui précède la phase de clustering.

Au vu du contexte d'étude, cet article propose de poser le problème de mesure de contribution des variables comme un problème de classification supervisée. C'est à dire apprendre à prédire l'appartenance aux clusters à partir d'une variable explicative donnée, puis d'ordonner les variables selon leur pouvoir prédictif.

La section 2 de cet article décrit la méthode de clustering utilisée qui contraint le problème de calcul d'importance. La section 3 décrit la solution proposée pour trier les variables en fonction de leur importance dans ce contexte. La section 4 présente des résultats préliminaires avant de conclure au cours de la dernière section.

2 L'existant : la méthode de clustering utilisée

L'ensemble des notations qui seront utilisées par la suite, sont les suivantes :

- Une base d'apprentissage, E, comportant N éléments (individus), M variables explicatives et une variable Y à prédire comportant J modalités (les classes à prédire sont C_j).
- Chaque élément D des données est un vecteur de valeurs (continues ou catégorielles) $D = (D_1, D_2, \ldots, D_M)$.
- K est utilisé pour désigner le nombre de classes souhaitées.

2.1 L'algorithme de clustering

L'algorithme de clustering utilisé est décrit dans (Lemaire et al., 2012). Cet article a montré que si on utilise un algorithme de type k-moyennes à l'aide d'une présentation supervisée et de la norme L1, on obtient des clusters où deux individus proches au sens de la distance seront proches au sens de leur probabilité d'appartenance à la classe cible (voir équation 5 dans (Lemaire et al., 2012)). Cet algorithme peut être présenté de la manière suivante :

- Prétraitements des données (voir section 2.2)
- Pour replicate=1 à R [1]
  - initialisation des centres (voir section 2.3)
  - Algorithme usuel des k-moyennes avec comme centre une approximation de la médiane (Kashima et al., 2008) et la norme L1 (Jajuga, 1987)
- choix de la meilleure "replicate" parmi les R solutions obtenues (voir section 2.4)
- présentation des résultats (voir section 2.5)

Cet algorithme est en partie supervisé puisque les prétraitements et le choix du meilleur "replicate" sont basés sur des critères supervisés qui sont décrits ci-dessous.

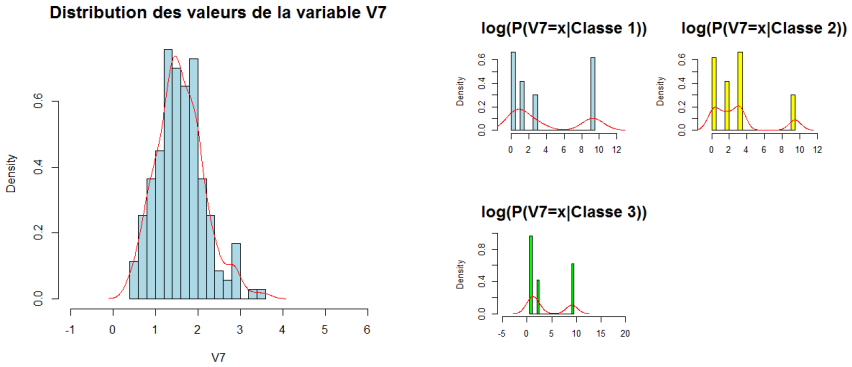
1. Dans cette étude, on fixe le nombre de replicates à R= 50

2.2 Représentation supervisée des données

Une représentation supervisée des données est utilisée. Elle recode les données brutes grâce à une technique de groupage supervisée ou de discrétisation supervisée qui utilise la variable *cible* contenant la liste des classes à prédire.

Les variables continues sont discrétisées (Boullé, 2004), c'est à dire découpées en intervalles, tandis qu'une méthode de groupage est appliquée sur les variables catégorielles (Boullé, 2005). Le prétraitement des données est réalisé à l'aide de l'approche MODL. Cette approche consiste à trouver la partition des valeurs de la variable continue (respectivement catégorielle) qui donne le maximum d'information sur la répartition des classes à prédire connaissant l'intervalle de discrétisation (respectivement le groupe de modalités).

A la fin du processus de prétraitement, les variables numériques et catégorielles sont donc recodées : chaque variable m est recodée en une variable qualitative contenant I_m valeurs de recodage. Chaque objet de données est alors recodé sous forme d'un vecteur de modalités discrètes $D = D_{1i_1}, D_{2i_2}, \ldots, D_{Mi_M}$. D_{mi_m} représente la valeur de recodage de D_m sur la variable m, avec la modalité discrète d'indice i_m. Ainsi les variables de départ sont alors toutes représentées sous une forme numérique. Le vecteur initial contenant M composantes de variables numériques et catégorielles devient un vecteur de $M * J$ composantes numériques : $log(P(D_{mi_m}|C_j))$.



(a) avant le prétraitement de la variable (b) après le prétraitement de la variable

FIG. 1 – *la distribution des valeurs de la variable V7 avant et après le prétraitement*

A titre illustratif, la figure 1 présente la discrétisation d'une variable numérique de la base UCI (Blake et Merz, 1998) "Wine" qui contient 13 variables explicatives et une variable cible à 3 classes ($Y \in \{1, 2, 3\}$). Après le prétraitement, on remarque que la distribution des variables prétraitées est multimodale et non gaussienne.

2.3 Initialisation des centres

L'initialisation des algorithmes de clustering basés sur le partitionnement influence la qualité de la solution trouvée et le temps d'exécution. C'est pourquoi le choix de la méthode d'initialisation de l'algorithme est un choix important lors de l'implémentation d'un algorithme de

clustering. Cependant, il n'y a pas une méthode d'initialisation meilleure que toutes les autres dans la littérature (Meila et Heckerman, 1998) mais plusieurs bonnes méthodes. Parmi ces dernières la méthode nommée K means++ a été utilisée (Arthur et Vassilvitskii, 2007). Cet algorithme est défini comme suit :

1. Choisir un centre uniformément au hasard parmi l'ensemble des points de données E.
2. Pour chaque point D, calculer $S(D)$: la distance entre D et le centre le plus proche qui a déjà été choisi.
3. Choisir le centre prochain $c_i = D' \in E$ suivant la probabilité $\frac{S(D')^2}{\sum_{D \in E} S(D)^2}$.
4. Répéter les étapes 2 et 3 jusqu'à ce que l'on ait placé tous les centres.

2.4 Choix de la meilleure replicate

Afin de prémunir contre le problème lié à l'initialisation et au fait que l'algorithme ne garantit pas d'avoir un minimum global, on exécute l'algorithme de clustering plusieurs fois. On obtient donc un certain nombre de partitionnements différents, dont on souhaite garder uniquement le meilleur. Pour se faire, et puisqu'on est dans le cadre de clustering supervisé, on utilise une mesure de qualité nommée *EVA* qui mesure la qualité d'un clustering supervisé en prenant en considération la variable 'cible'.

EVA mesure le gain qu'une partition établissant un compromis entre le nombre de groupe et la répartition des étiquettes peut apporter par rapport à la partition ayant un seul groupe. Plus formellement, EVA est une description scalaire comprise entre 0 et 1, décrite par la formule suivante : $EVA = 1 - \left(\frac{c(K)}{c(1)}\right)$, où

$$c(K) = log(N) + log\left(\begin{array}{c} N+K-1 \\ K \end{array}\right) + \sum_{k=1}^{K} log\left(\begin{array}{c} N_k+J-1 \\ J-1 \end{array}\right) + \sum_{k=1}^{K} log\left(\frac{N_k!}{N_{k1}! \ldots N_{kJ}!}\right) \tag{1}$$

et où K est le nombre de cluster, N_{kj} est le nombre d'individus du cluster k et de classe j et N_k le nombre d'individus dans le cluster k.

$c(K)$ mesure d'une manière supervisée l'intérêt d'une partition de Voronoi relative à un échantillon. Il quantifie le compromis entre le nombre de groupes de la partition et la distribution de la variable cible, ce qui correspond à un compromis entre complexité du modèle et ajustement du modèle aux données de l'échantillon. D'une manière générale, on cherche à maximiser cette mesure. Cette mesure est détaillée dans(Ferrandiz et Boullé, 2010).

2.5 Présentation des résultats du clustering

A la fin de la convergence de la méthode de clustering, on présente les résultats à l'aide des groupes de modalités et des intervalles créés lors de l'étape de prétraitement, en calculant les effectifs des individus dans chaque groupe de modalités ou intervalle pour chaque cluster. A titre d'exemple, le tableau 1 présente les effectifs des individus dans l'ensemble des intervalles de la variable V7 de la base Wine pour les trois clusters.

		id-cluster			
	Intervalle / Groupe de modalités	Cluster 1	Cluster 2	cluster 3	Total
V1	...	...	...	...	...
	...	...	...	...	...
	...	...	...	...	...
...					
V7	$]-\infty$; 0.975]	0	1	38	39
	]0.975 ; 1.575]	0	13	10	23
	]1.575 ; 2.31]	1	38	0	39
	]2.31 ; $+\infty$]	58	19	0	77
...					
V13	...	...	...	...	...
	...	...	...	...	...
	...	...	...	...	...
	...	...	...	...	...

TAB. 1 – *Discrétisation de la variable V7*

3 Choix d'une méthode de tri adaptée au contexte

3.1 Contribution d'une variable

Dans la littérature, plusieurs indices de qualité de clustering ont été développés afin de mesurer la contribution d'une variable au résultat d'un clustering. Cette problématique de mesure de contribution, de mesure d'importance, dans un clustering peut être divisée en deux sous-problèmes que l'on peut respectivement caractériser de *global* ou *local*. L'importance *globale* a pour but de mesurer l'impact que la variable a eu sur la structure entière du partitionnement et non pas l'impact qu'elle a eu sur un cluster en particulier. Par contre, l'importance *locale* a pour objectif de savoir quelle variable a été déterminante dans la formation d'un cluster en particulier. Nous nous intéressons dans cet article uniquement à l'importance globale.

Parmi les méthodes de l'état de l'art permettant de mesurer cette importance on trouvera de nombreux indices tels que : (i) l'indice de Dunn (Dunn, 1974) ; (ii) l'indice de Davies-Bouldin (DB) (Davies et Bouldin, 1979) ; (iii) l'indice Silhouette (Rousseeuw, 1987) ; l'indice SD (Halkidi et al., 2000) ; l'indice S_Dbw(Halkidi et Vazirgiannis, 2001) ...

La plupart de ces méthodes utilisent le théorème de Huygens et la décomposition de l'inertie totale en la somme de l'inertie intra cluster et de l'inertie inter cluster. La contribution d'une variable est alors, par exemple, calculée en mesurant la valeur de l'inertie inter calculée uniquement avec cette variable vis-à-vis de la somme des inerties inter calculée sur toutes les variables ((Benzécri, 1983), (Celeux et al., 1989) section 2.10 p154-164).

3.2 Notre proposition

Notre but est l'ordonnancement du tableau 1 selon la contribution des variables à l'affection des clusters. Nous pensons que dans le cadre de notre contexte et de nos prétraitements les critères classiques tel que ceux présentés ci-dessus ne sont pas totalement adaptés. La figure 1 montre par exemple que pour la base de données Wine la distribution de départ de la variable V7 (partie gauche de la figure) devient après prétraitements « multimodale » (partie droite de la figure).

Nous décidons alors de poser le problème comme un problème de classification supervisée. Le but sera d'essayer d'apprendre à prédire le cluster d'appartenance d'un individu (l'id-cluster du tableau 1) en utilisant une seule variable (classification univariée). Puis de trier les variables selon leur pouvoir prédictif vis-à-vis de l'id-cluster.

Comme on désire trier les variables selon le résultat de clustering initialement obtenu on s'interdira les classifieurs qui créent une nouvelle représentation des données. En effet on ne souhaite pas mesurer l'importance des variables dans un nouvel espace mais l'importance des variables avec la représentation supervisée obtenue juste avant la création des clusters. Le but est l'aide à l'interprétation du clustering de manière à permettre à l'analyste de concentrer son attention sur les variables les plus importantes vis-à-vis du clustering obtenu.

Parmi les méthodes capables d'utiliser la représentation issue de nos prétraitements supervisés et le tableau d'effectifs qui sert à présenter les résultats on choisit d'utiliser la méthode MODL qui mesure le pouvoir prédictif (appelé "level") d'une variable numérique dans (Boullé, 2004) et le pouvoir prédictif d'une variable catégorielle dans (Boullé, 2005).

Dans le cas d'une variable numérique [respectivement catégorielle] si les intervalles de discrétisation [les groupes de modalités] sont fixés, alors le critère se calcul à l'aide des effectifs observés dans les intervalles [groupes de modalités]. Nos prétraitements supervisés nous donnent les intervalles [les groupes de modalités] et la projection des individus sur les clusters (tableau 1) nous permettent d'avoir en notre possession les effectifs. L'ensemble des éléments nécessaire au calcul du level par variable est donc disponible pour toutes les variables explicatives.

4 Expérimentations

4.1 Jeu de données utilisé

Pour évaluer le comportement de notre nouvelle approche en termes de tri des variables selon leur importance, des tests préliminaires ont été effectués sur les bases de données suivantes (Blake et Merz, 1998) :

- Wine : Cette base contient les résultats d'une analyse chimique des vins produits dans la même région en Italie, mais provenant de trois cultivateurs différents (trois classes à prédire). Elle est constituée de 178 données caractérisées par 13 attributs continus.
- Letters : Cette base est constituée de 20000 données caractérisées par 16 attributs et 26 classe à prédire.
- Iris : Cette base est constituée de 150 données caractérisées par 4 attributs continus et trois classe à prédire.

4.2 Algorithme utilisé pour comparer les mesures d'importance

Une bonne mesure d'importance doit permettre de trier les variables en fonction de leur importance. Les moins bonnes de ces variables ne contiennent pas, ou peu d'information utile à la formation des clusters. Le résultat d'un clustering sur le jeu de données privé de cette variable, et donc sa qualité, devrait rester sensiblement identique, ou même être légèrement meilleur (moins de bruit). Inversement, le retrait d'une variable importante, priverait l'algo-

rithme d'une information importante pour former les clusters produisant alors un clustering de moins bonne qualité.

On définit alors un algorithme simple qui nous permet de recueillir les informations pour comparer les différentes mesures d'importance :

1. exécuter l'algorithme de clustering afin d'obtenir un premier partitionnement.
2. trier les variables selon leur importance, à l'aide de la méthode de tri que l'on souhaite tester.
3. exécuter l'algorithme de clustering afin d'obtenir un nouveau partitionnement.
4. estimer la qualité de ce partitionnement à l'aide des critères EVA et AUC.
5. retirer du jeu de donnée la variable la moins importante, d'après le tri effectué en 2.
6. réitérer à partir de l'étape 3, jusqu'à un critère d'arrêt (par exemple, toutes les variables ont été retirées).

On peut alors tracer la courbe des valeurs d'EVA (respectivement AUC (Fawcett, 2004)) en fonction du nombre de variables. L'examen des résultats peut alors être fait visuellement en observant l'évolution de la courbe des valeurs d'EVA (respectivement AUC) et/ou en calculant l'aire sous la courbe des valeurs d'EVA (respectivement AUC) (ALC = Area Under Learning Curve (Salperwyck et Lemaire, 2011)). Plus l'ALC est élevée plus la méthode de tri est de bonne qualité.

4.3 Les méthodes de tri implémentées

Nous avons listé dans la section 3.1 plusieurs indices permettant de trier les variables en fonction de leur importance dans un clustering. Pour des raisons de temps et de coût d'implémentation, à ce jour deux indices ont été implémentés à savoir Davies-Bouldin (BD) (Davies et Bouldin, 1979) et SD (Halkidi et al., 2000). Dans cette section ces deux indices seront comparés à notre approche présentée dans la section 3.2.

4.4 Résultats

Level	V3	V8	V5	V4	V2	V9	V11	V1	V6	V13	V10	V12	V7
Indice-DB	V4	V2	V13	V11	V3	V1	V10	V8	V9	V5	V12	V6	V7
Indice-SD	V4	V5	V3	V2	V13	V8	V11	V9	V1	V6	V10	V7	V12

TAB. 2 – *Tri des variables (de la moins importante à la plus importante) à l'aide des trois méthodes*

Le tableau 2 présente à titre illustratif l'ordonnancement des variables en fonction de leur importance dans le clustering obtenu, pour la base Wine, à l'aide des trois méthodes de tri implémentées. A partir de nos prétraitement [2], les deux dernières méthodes (Davies-Bouldin et SD) calculent pour chacune de ces variables trois valeurs de contribution conditionnellement à la classe à prédire. Cela veut dire qu'une seule variable peut avoir une forte contribution à la

2. Dans le cas du jeu de données wine (cas de 3 classe à prédire C_1, C_2 et C_3), chaque variable est prétraitée de la manière suivante : $log(P(X_i = x|C_1), log(P(X_i = x|C_2), log(P(X_i = x|C_3)$ avec $i \in [\![1, 13]\!]$.

construction des clusters conditionnellement à une classe à prédire et en même temps une faible contribution conditionnellement à une autre classe. Dans ce cas, on définit une contribution d'une variable comme étant la somme des trois valeurs [3]. La méthode proposée (level), est une méthode capable d'utiliser la représentation issue de prétraitement et le tableau des effectifs pour fournir une valeur de contribution par variable.

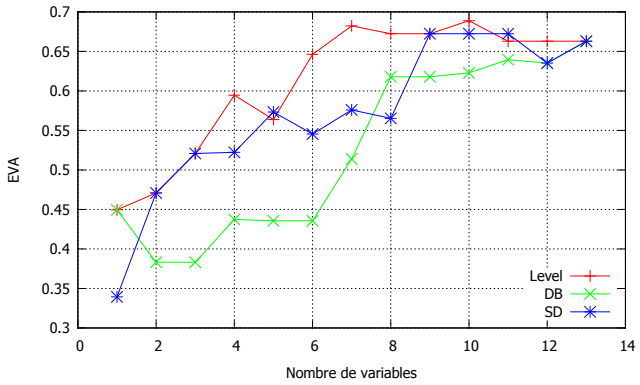


FIG. 2 – *Evolution du critère EVA pour les trois méthodes (K=3)*

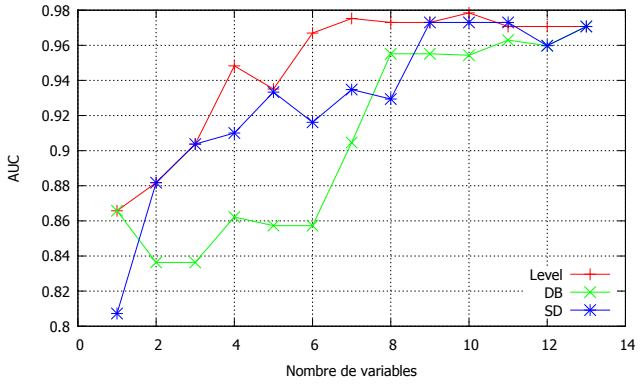


FIG. 3 – *Evolution du critère AUC pour les trois méthodes (K=3)*

La figure 2 (respectivement la figure 3) présente les trois courbes d'EVA (respectivement AUC [4]) à titre illustratif sur la base Wine en fonction de la méthode utilisée.

Le tableau 3 présente quand à lui les valeurs d'ALC pour EVA, l'AUC et l'ACC (le taux de bonne classification) selon le critère utilisé.

3. Une variable considérée comme moins contributrice pour un clustering, doit être retirée entièrement du jeu de données

4. Le critère AUC est donné par la formule suivante : $\sum_{j\in[\![1,J]\!]} P(C_j)AUC(C_j)$ avec J est le nombre de classe. Il permet de mesurer la qualité de la classification en traitant chaque cluster individuellement. Notons que la classe prédite d'un cluster est définit comme étant la classe majoritaire de celui-ci.

L'évolution du critère EVA (respectivement AUC) à l'aide de la méthode proposée est meilleure vis-à-vis de l'évolution des deux autres critères à mesure que l'on retire les variables jugées les moins contributrices pour le clustering obtenu.

		DB	SD	level
Wine	ALC(EVA)	0,5257	0,5714	0,6116
	ALC(AUC)	0,9060	0,9281	0,9472
	ALC(ACC)	0,8574	0,8863	0,9123
Letters	ALC(EVA)	0,3628	0,2871	0,3749
	ALC(AUC)	0,8930	0,8558	0,8952
	ALC(ACC)	0,3475	0,2813	0,3555
Iris	ALC(EVA)	0,6304	0,4571	0,6304
	ALC(AUC)	0,9675	0,9078	0,9675
	ALC(ACC)	0,9350	0,8267	0,9350

TAB. 3 – *Les valeurs d'ALC pour les trois méthodes selon le critère utilisé.*

On remarque également qu'il possible de trouver un nombre restreint de variables produisant la même valeur d'EVA que l'ensemble complet des variables de départ. Par exemple sur la base de données Wine, et à l'aide de la méthode proposée, on aurait pu déterminer un jeu de 7 variables qui auraient produit un clustering supervisé presque de même qualité que celui obtenu à l'aide de 13 variables.

5 Conclusion

Cette contribution a présenté une nouvelle méthode de tri des variables en cours d'élaboration dans notre contexte industriel particulier. Cette méthode trie les variables en fonction de leur importance à la fin de la convergence de notre clustering qui est "supervisé" en partie. Les résultats préliminaires qui ont été obtenus sont encourageants et semblent montrer l'intérêt de la méthode. Néanmoins ces résultats devront être confirmés sur d'avantage de base de données et comparés à un jeu de critère de qualité de la littérature de plus grande taille.

Références

Arthur, D. et S. Vassilvitskii (2007). K-means++ : The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035.

Benzécri, J. P. (1983). Analyse de l'inertie intraclasse par l'analyse d'un tableau de correspondance. pp. 351 – 358.

Blake, C. L. et C. J. Merz (1998). Uci repository of machine learning databases. last visited : 01/12/2013, http ://archive.ics.uci.edu/ml/.

Boullé, M. (2004). A Bayesian approach for supervised discretization. In Zanasi, Ebecken, et Brebbia (Eds.), *Data Mining V*, pp. 199–208. WIT Press.

Boullé, M. (2005). A grouping method for categorical attributes having very large number of values. In P. Perner et A. Imiya (Eds.), *Proceedings of the Fourth International Conference*

on Machine Learning and Data Mining in Pattern Recognition, Volume 3587 of *LNAI*, pp. 228–242. Springer verlag.

Celeux, G., E. Diday, G. Govaert, Y. Lechevallier, et H. Ralambondrainy (1989). *Classification automatique des données*. Dunod.

Davies, D. L. et D. W. Bouldin (1979). A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-1*(2), 224–227.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics 4*(1), 95–104.

Fawcett, T. (2004). Roc graphs : Notes and practical considerations for researchers. *Machine learning 31*(7), 1–38.

Ferrandiz, S. et M. Boullé (2010). Bayesian instance selection for the nearest neighbor rule. *Machine Learning 81*(3), 229–256.

Halkidi, M. et M. Vazirgiannis (2001). Clustering validity assessment : finding the optimal partitioning of a data set. In *Proceedings IEEE International Conference on ICDM 2001*, pp. 187–194.

Halkidi, M., M. Vazirgiannis, et Y. Batistakis (2000). Quality scheme assessment in the clustering process. In D. A. Zighed, J. Komorowski, et J. ?ytkow (Eds.), *Principles of Data Mining and Knowledge Discovery*, Volume 1910 of *Lecture Notes in Computer Science*, pp. 265–276. Springer Berlin Heidelberg.

Jajuga, K. (1987). A clustering method based on the l1-norm. *Computational Statistics & Data Analysis 5*(4), 357–371.

Kashima, H., J. Hu, B. Ray, et M. Singh (2008). K-means clustering of proportional data using l1 distance. In *19th International Conference on ICPR*.

Lemaire, V., F. Clérot, et N. Creff (2012). K-means clustering on a classifier-induced representation space : application to customer contact personalization. In *Annals of Information Systems, Springer, Special Issue on Real-World Data Mining Applications*.

Meila, M. et D. Heckerman (1998). An experimental comparison of several clustering and initialization methods. *Machine Learning*.

Rousseeuw, P. J. (1987). Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*(0), 53 – 65.

Salperwyck, C. et V. Lemaire (2011). Learning with few examples : An empirical study on leading classifiers. In *IJCNN*, pp. 1010–1019.

Summary

This article presents a preliminary study made in an industrial context. We have a typology of customers that the marketing service want to contact. This typology is a segmentation of customers into groups, whose profils will be used to propose differentied marketing campaigns. The constitution of groups is realised by using a clustering technique which does not currently allow the importance of the variables. This article proposes to solve this problem by using a methodology which gives in our industrial context the importance of variables. This method will be compared with some others methods from the literature.

Index