K-moyennes contraintes par un classifieur Application à la personnalisation de scores de campagnes

Vincent Lemaire*, Nicolas Creff**, Fabrice Clérot*

* Orange Labs, 2 avenue Pierre Marzin, 22300 Lannion
**Epita 14-16 rue Voltaire 94276 Kremlin Bicêtre Cedex

Résumé. Lorsqu'on désire contacter un client pour lui proposer un produit on calcule au préalable la probabilité qu'il achètera 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 la plus forte probabilité d'acheter le produit. En parallèle, et avant le contact commercial, il peut être intéressant de réaliser une typologie des clients qui seront contactés. L'idée étant de proposer des campagnes différenciées par groupe de clients. Cet article montre comment il est possible de contraindre la typologie, réalisée à l'aide des k-moyennes, à respecter la proximité des clients vis-à-vis de leur score d'appétence.

1 Introduction

1.1 Problématique industrielle

Le data mining consiste en l'ensemble des méthodes et techniques qui permettent d'extraire des informations à partir d'une grande masse de données. Son utilisation permet d'établir des relations entre ces données et par exemple de définir des comportements type de clients dans le cadre de la gestion de la relation client.

Lorsqu'on désire contacter un client pour lui proposer un produit on calcule au préalable son appétence à ce produit. Il s'agit là de calculer la probabilité qu'il achètera ce produit. Cette probabilité (encore appelée score) est calculée à l'aide d'un modèle prédictif pour un ensemble de clients (le périmètre de la campagne). Le calcul du score exploite un grand nombre de variables explicatives issues du système d'information.

Les clients sont ensuite triés dans l'ordre décroissant de leur probabilité d'appétence. Le service marketing ne contacte ensuite que les plus appétents (nommés "tops scores"), i.e. ceux ayant la plus forte probabilité d'acheter le produit. En parallèle et avant le contact commercial il peut être intéressant de réaliser une typologie des clients qui seront contactés. L'idée étant de proposer des campagnes différenciées par segment. Un argumentaire commercial est construit pour chaque groupe de clients après analyse des caractéristiques du groupe : âge, CSP, offres actuellement détenues. Il est fréquent (pour des raisons pratiques de temps d'analyse) que l'analyse du groupe se résume en l'analyse du centre (ou représentant) du groupe.

Aujourd'hui, cette typologie est très souvent réalisée de manière non supervisée à l'aide d'une technique de partitionnement de type k-moyennes. La valeur de k est prédéfinie et la

métrique utilisée ne tient pas compte du prédicteur (le modèle délivrant la probabilité d'appétence) ce qui pose deux problèmes :

1. Les clients dans les clusters ne sont pas liés par leur probabilité d'appétence : un cluster peut contenir des clients très appétents et des clients peu appétents. L'analyse du centre du groupe rend l'argumentaire commercial erroné.

2. Les segments créés ne sont pas stables dans le temps lorsque le classifieur est déployé plusieurs mois consécutivement sur le même périmètre de campagne (voir les deux critères section 4.3).

On pourrait essayer alors de poser ce problème comme un problème supervisé en essayant de construire un second modèle de classification (ou de régression) avec pour variable cible les scores issus du premier classifieur. On obtiendrait alors, par exemple dans le cas d'un arbre de décision, les raisons qui impactent le plus les scores. Cette voie bien qu'intéressante ne correspond pas à notre souhait qui est bien plus de découvrir la "structure" des scores plus que de grouper les clients par pureté des scores. En effet deux mêmes scores peuvent avoir été "obtenus" par des voies différentes et ce sont ces voies, ou raisons, que nous cherchons à découvrir.

Aussi pour résoudre les problèmes cités ci-dessus cet article propose de réaliser une typologie à l'aide d'un algorithme de partitionnement. Cet algorithme sera contraint par la connaissance issue du classifieur qui calcule les scores d'appétence. Il s'agit de construire un clustering qui conserve la proximité des clients ayant les mêmes scores d'appétence.

La section 2 de cet article décrit le processus qui a conduit à choisir l'algorithme des k-moyennes comme algorithme de partitionnement. Muni du choix de l'algorithme la section 3 détaille comment il est possible d'utiliser une métrique, au cours du calcul du partitionnement, qui dépende du classifieur utilisé pour calculer les scores d'appétence. La section 4 présentera les résultats obtenus avant de conclure au cours de la dernière section.

2 Choix d'une technique parmi les différentes méthodes de clustering basées sur le partitionnement

Le clustering est un processus de partitionnement d'un ensemble de données en un ensemble significatif de groupes appelés clusters. Le but du regroupement est de trouver des groupes d'éléments similaires au sens d'une mesure de similarité donnée. Il y a donc deux éléments principaux à choisir : la méthode de création des groupes et la métrique utilisée lors de la création des groupes.

Les notations qui seront utilisées dans la suite de cet article sont les suivantes :

– une base d'apprentissage, $\mathcal{D}$, comportant N instances, M attributs et une variable à prédire comportant J modalités (les classes à prédire sont notées C_j) ;
– chaque instance, D, de données est un vecteur de valeurs (continues ou catégorielles) $D = (D_1, D_2, ..., D_M)$;
– k est utilisé pour désigner le nombre de classes souhaitées.

2.1 Introduction

Il existe 4 grandes techniques de partitionnement qui peuvent être utilisées pour regrouper les éléments d'un ensemble de donnée autour : d'un centre de gravité (la moyenne empirique) : les k-moyennes (MacQueen, 1967) ; d'une médiane géométrique : les k-médianes (Bradley et al., 1997) ; d'un centre contenant les modes les plus fréquents : les k-modes (Huang, 1998) ; d'un medoid (l'élément d'un ensemble qui minimise la somme des distances entre lui et chacun des autres éléments de cet ensemble) : les k-medoids (Kaufman et Rousseeuw, 1990).

Le choix d'un de ces algorithmes dépend de la nature des données sur lesquelles il devra être appliqué ; du résultat souhaité (moyenne, medoid, ...) ; du temps disponible et donc de la complexité de l'algorithme. De plus, chacun des ces algorithmes dépend des représentants initialement choisis, de la valeur de k, de l'indicateur de mesure qui évaluera la qualité de la partition (la cohésion des clusters obtenus), de la distance ou mesure de similarité utilisée, de la représentation des données présentées en entrée de l'algorithme. Ces différents points sont discutés ci-dessous et sont mis régulièrement en relation avec le contexte industriel de l'étude.

2.2 Influence de la nature des données initiales

On est ici dans un contexte industriel très précis. Les données proviennent du système d'information d'Orange. Les variables explicatives qui sont placées en entrée du classifieur servant à calculer les probabilités d'appétences sont numériques ou catégorielles (avec un grand nombre de modalités) et il existe des valeurs manquantes. Le lecteur pourra trouver une description de ces données dans (Guyon et al., 2009). Si on reste à une telle représentation des données le choix de la technique de partition devrait se tourner vers la technique des k-prototypes (Huang, 1997) qui est un mixage des k-moyennes et des k-modes.

Les données utilisées peuvent aussi contenir un certains nombre de clients atypiques (ou des données erronées) ce qui dans ce cas amènerait plus au choix des k-medoid qui sont par nature moins sensibles aux valeurs aberrantes.

2.3 Influence du type de résultat souhaité

Le résultat du partitionnement doit permettre de construire un argumentaire commercial par cluster. Un argumentaire de vente est un ensemble structuré d'arguments qui présente les caractéristiques d'un produit / service comme autant d'avantages pour le client. Il suppose une connaissance approfondie du produit (caractéristiques), mais aussi des besoins et des motivations du client. Il doit être adapté au client. Par conséquent on souhaiterait que le "centre" des clusters formés soit un "vrai" client et non un client moyen. Il est en effet difficile de savoir ce qu'est, par exemple, la moyenne de deux offres commerciales. Ce desideratum fait pencher le choix de la technique de partition vers les k-medoid.

2.4 Influence de la métrique

Un certain nombre d'éléments sont à prendre en compte lors du choix de la métrique. D'un côté la forme des clusters obtenus dépend de la norme utilisée. D'un autre coté chacun des algorithmes décrits ci-dessus (k-moyennes, ...) vise à minimiser une norme en particulier : les

k-moyennes la norme L2, les k-médianes la norme L1 (Jajuga, 1987)... Bien que les algorithmes de clustering basés sur le partitionnement fonctionnent à peu près avec n'importe quel type de fonction de distance (ou mesure de similarité) on n'obtient pas les mêmes garanties selon la métrique que l'on utilise. Par exemple le théorème de Huygens qui montre que la somme de l'inertie intraclusters et l'inertie interclusters est constante n'est valable que si l'on utilise la distance euclidienne. Dans notre cas on désire adapter la métrique à celle naturellement incluse dans le classifieur servant à calculer les probabilités d'appétences. Cette adaptation sera décrite au cours de la section 3 ci-dessous. On mentionne uniquement pour le moment, pour la compréhension de la suite de cette section, que l'on utilisera une norme L1 pondérée".

2.5 Influence de la complexité algorithmique

Les complexités algorithmiques des différentes techniques de partitionnement varient énormément selon la technique de partitionnement mais aussi selon l'implémentation qui en est faite. On trouve dans (Har-peled et Mazumdar, 2003) différentes implémentations des k-médianes, dans (Kaufman et Rousseeuw, 1990) différentes implémentations des k-medoids (PAM (Partitioning Around Medoids), CLARA (Clustering LARge Applications) et CLARANS (Clustering Large Applications based upon RANdomized Search))... Si on classe ces algorithmes de la complexité la plus faible à la plus élevée on trouve les k-moyennes, les k-modes, les k-medoids et finalement les k-médianes.

Les campagnes marketing concernées par cette étude utilisent des bases de données comportant des centaines de milliers de clients chacun décrit potentiellement par plusieurs (dizaines de) milliers de variables explicatives. Après la construction du classifieur (qui réalise une sélection de variables) et en ne retenant que les clients ayant les plus fortes probabilités d'appétences on obtient des bases de données de quelques dizaines de milliers de clients décrits par une à plusieurs centaines de variables explicatives. Ce sont ces bases de données qui sont utilisées pour construire le partitionnement. On s'aperçoit donc que certains de ces algorithmes seront difficilement utilisables (au-delà d'une certaine volumétrie).

2.6 Influence du prétraitement

Le classifieur utilisé par Orange dans le cadre de cette étude, pour calculer les probabilités d'appétences est Khiops^{TM}(au sein de la plateforme PAC (Féraud et al., 2010)). Khiops incorpore un classifieur naïf de Bayes (Langley et al., 1992) qui est construit après une étape de prétraitement des variables. Khiops va discrétiser les variables numériques et faire du groupage de modalités pour les variables catégorielles. A la fin du processus de prétraitement les variables numériques et catégorielles sont donc recodées : chaque attribut m est recodé en un attribut qualitatif contenant I_m valeurs de recodage. Chaque instance de données est alors recodée sous forme d'un vecteur de modalités discrètes $D = D_{1i_1}, D_{2i_2}, ..., D_{Mi_M}$. D_{mi_m} représente la valeur de recodage de D_m sur l'attribut m, avec la modalité discrète d'indice i_m. Ainsi les variables de départ sont alors toutes représentées sous une forme numérique, sur un vecteur de $M * J$ composantes : $P(D_{mi_m}|C_j)$.

Ce prétraitement rend inutile le choix d'un algorithme comme celui des k-modes puisque toutes les variables après l'étape de prétraitement sont de type numérique. Il atténue également l'intérêt des k-médianes / k-medoid vis-à-vis des 'outliers' car après ce type de prétraitement il n'y a plus de valeurs aberrantes dans les données.

2.7 Discussion

Les éléments présentés ci-dessus montrent que de nombreuses contraintes influent sur le choix d'un algorithme de partitionnement qui soit adapté à la problématique industrielle. A titre d'exemple, la complexité algorithmique et la nature des prétraitements effectués rend l'algorithme des k-moyennes très adapté à notre problématique industrielle mais rend cet algorithme moins adapté du fait de l'usage d'une norme L1 et du souhait d'avoir de vrais clients comme centres de cluster.

L'algorithme des k-médianes est plus adapté vis-à-vis de la norme utilisée et de la nature des données après prétraitement mais sa complexité algorithmique le rend inutilisable sur nos données.

L'algorithme des k-medoid est l'algorithme qui vient naturellement ensuite comme choix mais sa complexité algorithmique bien que plus faible que celle des k-médianes le rend difficilement exploitable car elle reste trop élevée (plusieurs heures de calcul pour de petites bases de données même avec des algorithmes de type CLARANS). D'autres algorithmes (Park et Jun, 2009) modifient légèrement l'algorithme des k-medoid pour le rendre proche de celui des k-moyennes en terme de complexité mais ils nécessitent de mettre en mémoire la matrice des distances entre les clients.

Décision a été prise alors d'utiliser l'algorithme des k-médianes en prenant une approximation de la médiane comme prototype sous l'hypothèse d'indépendance des variables et en lui ajoutant une étape finale après convergence. L'hypothèse d'indépendance des variables permet d'utiliser une version rapide du calcul de la médiane appelée "the component-wise median" (Kashima et al., 2008). L'étape réalisée après la convergence de l'algorithme consiste à remplacer chaque prototype par le "vrai" client (au sein de ce cluster) qui est le plus proche du prototype. La proximité entre le vrai client et le prototype du cluster est calculée à l'aide d'une distance en norme L1. Cette étape peut légèrement dégrader les résultats du partitionnement mais elle permet de répondre à l'ensemble des souhaits énoncés dans la section 1.1 ci-dessus.

3 K-moyennes basées sur la connaissance du classifieur

3.1 Introduction

On montre dans cette section qu'il est possible d'insérer dans la métrique qui servira pour la construction des k-moyennes une connaissance issue du classifieur : le naïve Bayes moyenné pour le logiciel Khiops. Il s'agit de construire une nouvelle représentation dite "supervisée" qui permet de construire une métrique pondérée telle que deux instances proches dans cette représentation supervisée ont des scores proches.

La section suivante décrit comment on définit la distance dépendante de la classe pour un classifieur naïf de Bayes. La section 3.3 présentera quant à elle comment les variables explicatives se voient attribuer des poids et comment ces poids pondèrent la distance.

3.2 Distance dépendante de la classe

Reprenant les notations introduites en 2.6, il est possible d'écrire une distance Bayesienne dépendant de la classe à prédire. Si on part du prédicteur Bayesien naïf et que l'on passe au

log, on a pour chaque classe cible :

$$log(p(C_j|D)) = \sum_{m=1}^{M} log\left(p(D_{mi_m}|C_j)\right) + log(p(C_j)) - log(p(D)) \quad (1)$$

$D = (D_m)_{m=1,...,M}$ une instance

On rappelle que la décision Bayesienne correspond à la classe cible C_j maximisant la formule précédente. On définit la distance entre deux instances, d^1_{NB} de la façon suivante :

$$d^1_{NB}(D, D') = \sum_{m=1}^{M}\sum_{j=1}^{J}\left|log\left(p(D_{mi_m}|C_j)\right) - log\left(\left(p(D^{'}_{mi_m}|C_j)\right)\right)\right| \quad (2)$$

On peut alors coder chaque instance sur un vecteur de $M * J$ composantes, comme illustré dans 3 pour $J = 2$:

$$(log(p(D_{i1_1}|C_1)), log(p(D_{i1_1}|C_2)), ..., ..., log(p(D_{Mi_M}|C_1)), log(p(D_{Mi_M}|C_2))) \quad (3)$$

La distance proposée correspond à la norme L1 pour ce codage. La distance entre deux instances est donc définie en fonction des recodages. Deux instances proches au sens de leur recodage supervisé seront proches au sens de leur comportement pour la classe à prédire. En effet si on définit la distance entre les distributions de classes prédites de la façon suivante :

$$\Delta^1(D, D') = \sum_{j=1}^{J}|log(p(C_j|D)) - log(p(C_j|D'))| \quad (4)$$

On a la majoration suivante :

$$\Delta^1(D, D') \leq \left[d^1_{NB}(D, D') + J\,|log(p(D)) - log(p(D'))|\right] \quad (5)$$

Donc deux instances de même probabilité globale proches au sens de d^1_{NB} seront proches au sens de la prédiction des probabilités par classe cible (deux instances qui ont des recodages proches dans l'espace supervisé auront des probabilités proches d'avoir été générées par le modèle de recodage). Cette majoration est vraie aussi dans le cadre de la régression linéaire.

3.3 Pondération de la distance

L'étape de construction des poids des variables utilisées par le classifieur naïf de Bayes est totalement décrit dans (Boullé, 2007). Elle comprend deux étapes clefs : une étape de sélection de variable décrite dans la section 3.5 de l'article et une étape de moyennage de modèle décrite dans la section 6.2 de cet article. L'étape de sélection de variable permet au classifieur d'éviter d'avoir des variables explicatives inutiles ou non liées au problème de classification. L'étape de moyennage de modèle permet de pondérer les variable de telle sorte que l'équation 1 devient :

$$log(p(C_j|D) = \sum_{m=1}^{M} W_m log\left((p(D_{mi_m}|C_j)\right) + log(p(C_j)) - log(p(D)) \quad (6)$$

où W_m est le poids de la variable m quelle que soit la classe cible

Chaque instance est alors recodée sur un vecteur de $M * J$ composantes mais où chaque composante est pondérée par un poids. La distance (équation 2) est donc pondérée au sens des poids des variables, la majoration présentée équation 5 restant vraie.

3.4 Discussion - Algorithme modifié des K-moyennes

Dans ce qui suit on appellera "représentation supervisée" la représentation issue du passage de la base de données d'apprentissage initiale vers une représentation où chaque instance est représentée sur un vecteur de $M * J$ composantes (comme illustré dans l'équation 3) pour le classifieur naïf de Bayes. Chaque variable est pondérée à l'aide d'un poids W_m.

Le résultat ci-dessus (équation 5) donne la garantie que si on utilise un algorithme de type k-moyennes sur la représentation supervisée et à l'aide de la norme L1 on obtiendra des clusters où deux individus proches au sens de la distance seront proches au sens de leur probabilité d'appartenance à la classe cible.

L'algorithme des k-moyennes est dans la suite cet article est appelé "modifié" car il utilise (i) une représentation supervisée des données, (ii) la norme L1, (iii) une approximation de la médiane, (iv) une étape de post traitement de désignation de vrais clients comme centres. Ces 4 modifications devraient permettre d'atteindre les objectifs initiaux de l'étude tels que présentés dans l'introduction de cet article.

4 Résultats expérimentaux

4.1 Préambule

Initialisation : L'ensemble des méthodes d'initialisation mentionnées dans (Meila et Heckerman, 1998) ont été testées. Nous n'avons pas dans notre cas (prétraitements supervisés et/ou norme L1) mesuré de différences significatives entre les résultats obtenus. Les résultats présentés dans cet article sont ceux obtenus à l'aide d'une initialisation aléatoire des prototypes.

Validation croisée : Dans chacune des phases expérimentales (et pour toutes valeurs de k) les bases de données ont été découpées en 10 sacs afin de réaliser une validation croisée. Dans ce qui suit ce sont les résultats moyens en test et pour l'AUC (Area Under ROC Curve) qui sont indiqués. Le score d'appartenance à la classe cible d'un exemple est défini comme la proportion d'éléments de la classe cible du cluster de cet exemple. Dans le cas ou le nombre de classes cibles est supérieur à 2 on donne l'espérance de l'AUC.

4.2 Première phase expérimentale

Une première phase expérimentale a été menée de manière à (i) mesurer l'impact de la représentation supervisée sur les k-moyennes et à (ii) mesurer l'écart entre les résultats obtenus entre l'algorithme des k-medoid PAM (qui travaille directement sur les "vrais clients") et l'étape de post-désignation incluse dans l'algorithme modifié des k-moyennes.

Le logiciel Khiops a été testé à l'aide (i) des données natives et (ii) des données mises dans leur représentation supervisée. L'ensemble des valeurs de k testées vont de 1 à $\sqrt{N}$ par pas de

1, de 1 à 10, puis en doublant ensuite la valeur de k : $k \in \mathcal{A} = \{1, 2, ..., 9, 10, 20, 40, ..., \sqrt{N}\}$. Pour pouvoir comparer les résultats avec ceux obtenus par PAM la volumétrie a été limitée à de "petites" bases de données provenant de l'UCI (Blake et Merz, 1998). La somme des erreurs aux carrés (SSE) n'a pas été utilisée pour évaluer les résultats obtenus car elle est ici inappropriée du fait qu'on travaille ici sur deux représentations différentes (native et supervisée). Le critère de l'AUC a alors été choisi car il donne une indication de pureté des clusters au sens d'une classe cible.

Le tableau 1 compare le gain entre les résultats obtenus avec la représentation supervisée par rapport à la représentation native pour PAM et l'algorithme modifié des k-moyennes sur les bases Iris et Phonème. Pour les bases Letter et Shuttle PAM n'ayant pas abouti dans un temps acceptable (pour les différentes valeurs de k testées et la 10-fold cross-validation) seuls les résultats pour les k-moyennes sont présentés. Ce tableau présente les résultats en test et en AUC. Il s'agit là de résultats moyens calculés à l'aide des valeurs individuelles obtenues en fonction de k et du 10-fold (f) cross validation ($AUC = \frac{1}{|\mathcal{A}|10} \sum_{k \in A} \sum_{f=1}^{10} AUC(k, f)$). On ne présente ici que quelques résultats représentatifs (croissants en taille (J, N, M)) des tests effectués mais le lecteur intéressé pourra trouver plus de détails dans (Creff, 2011). Nous observons dans ce tableau que l'utilisation d'une représentation supervisée ne dégrade pas les résultats voire permet une amélioration du critère de l'AUC.

Base	Pam	Pam "supervisé"	Kmeans	Kmeans "supervisés"	J	N	M
Iris	0.959	0.951	0.946	0.966	2	150	4
Phonème	0.926	0.935	0.910	0.919	5	2554	256
Shuttle	-	-	0.902	0.929	7	58000	9
Letter	-	-	0.711	0.787	26	20000	16

TAB. 1 – *Phase 1 - AUC : Résultats moyens en test entre une représentation native et une représentation supervisée*

Les figures 1 et 2 illustrent les résultats obtenus sur les bases Abalone ($N = 4177, J = 28$) et Titactoe ($N = 958, J = 2$) en utilisant (uniquement) la représentation supervisée. Dans ces figures la courbe Rouge correspond à PAM, la courbe Bleu correspond à l'algorithme modifié des k-moyennes à l'aide de la métrique issue d'un naïve Bayes calculé par Khiops, enfin la courbe Noire correspond au classifieur naïf de Bayes calculé par Khiops.

Ces résultats illustratifs, ainsi que ceux présentés dans (Creff, 2011), montrent que l'algorithme modifié des k-moyennes utilisant la représentation issue du naïve Bayes Khiops est très compétitif. On observe aussi que, pour des valeurs élevées de k, l'algorithme modifié des k-moyennes, utilisé en tant que classifieur, peut atteindre des performances supérieures au naïve Bayes.

4.3 Deuxième phase

Plusieurs bases de données ont été mises à notre disposition pour cette phase. Trois bases de 200 000 clients datant de mars, mai, et août 2009 sur un problème de churn à l'un des produits d'Orange ont été utilisées. Ces bases sont constituées d'environ 1000 variables. La base de données du mois de mars a été utilisée pour construire le classifieur. Ensuite les tops scores de mars ont été utilisés pour réaliser la partition en k groupes à l'aide de l'algorithme

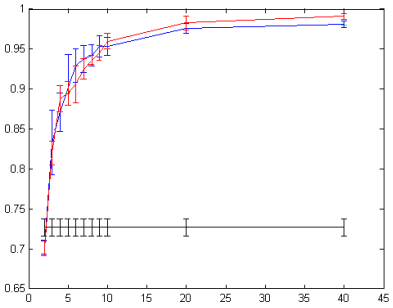

FIG. 1 – *AUC en Test pour Abalone*

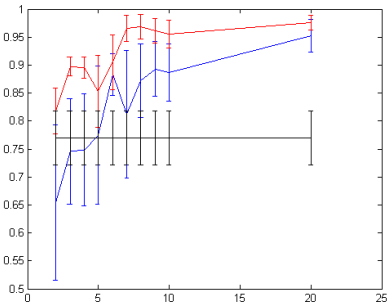

FIG. 2 – *AUC en Test pour Titactoe*

modifié des k-moyennes. Les bases de Mai et Août correspondront à des ensembles de tests. Les critères d'évaluation ont été calculés pour chacun des mois (mars, mai et août).

Dans notre contexte industriel, les utilisateurs de l'algorithme de clustering basé sur le partitionnement souhaitent pouvoir choisir eux-mêmes la valeur de k. Ils appliqueront ensuite à certains groupes une campagne téléphonique, à d'autres une campagne de mailing, à d'autres une campagne de courrier. Dans ce contexte, il est donc préférable de laisser le choix de k à l'utilisateur qui se basera sur son expertise, sur une connaissance à priori qu'il a sur les données, sur le fait qu'un nombre d'argumentaires commerciaux maximal peut être établi. Après consultation de l'entité concernée, 3 valeurs de k ont été testées : 4, 10, 20. Pour des considérations de place seuls les résultats avec k=4 sont présentés ci-dessous sachant que les conclusions énoncées restent valides pour $k = 10$ ou $k = 20$ (consultables dans (Creff, 2011)).

Au moment où les tests ont été effectués il existait une solution logicielle au sein de l'entreprise pour réaliser ce type de campagne. Mais cette solution n'était quasiment pas utilisée car les groupes obtenus deviennent trop différents de mois en mois. L'algorithme modifié des k-moyennes proposé dans cet article a été donc été évalué à l'aide d'un critère de stabilité dans le temps des clusters trouvés. Ce critère comprend deux axes.

Le premier axe est l'évolution du pourcentage d'appartenance à un cluster. Au mois T, on observe le pourcentage d'éléments de l'ensemble de données appartenant à un cluster. On recommence la même opération aux mois suivants avec d'autres ensembles de données. D'un mois à l'autre les proportions d'éléments appartenant à un cluster devraient rester les mêmes pour que l'on considère la solution comme stable par rapport à ce critère.

Le deuxième axe est l'évolution de la répartition des classes ou des valeurs cibles au sein des clusters. En fait chaque client est associé à une classe, ces classes ne sont pas utilisées pour réaliser le clustering. Au mois T, on observe dans les clusters la répartition des clients appartenant à une classe. On recommence l'opération les mois suivants et si la répartition des clients reste la même d'un mois à l'autre alors on pourra considérer la méthode de clustering comme stable au cours du temps.

Les résultats obtenus sur ces 2 axes de stabilité sont présentés figures 3 à 6. L'axe des abscisses représente les mois (T=1=mars, T=2=mai, T=3=août) et l'axe des ordonnées un pourcentage. Dans les figures 3 et 5 les pourcentages somment à 100% et correspondent aux tops scores. Par contre dans les figures 4 et 6 le pourcentage ne somme pas à 1 puisqu'il représente

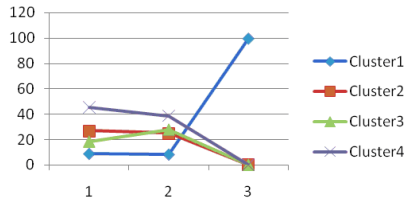

FIG. 3 – *Pourcentage d'éléments par cluster avec la solution actuelle.*

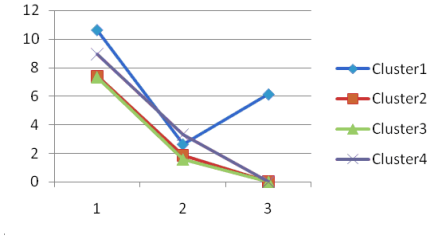

FIG. 4 – *Proportion d'éléments (churn=1) avec la solution actuelle.*

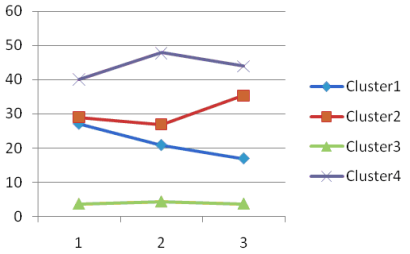

FIG. 5 – *Pourcentage d'éléments par cluster avec le k-moyennes "supervisé".*

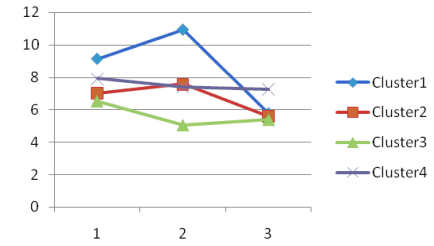

FIG. 6 – *Proportion d'éléments (churn=1) avec le k-moyennes "supervisé".*

la proportion d'éléments de chaque cluster ayant l'étiquette churn=1. On observe à l'aide de ces 4 figures que le but est atteint : les clusters trouvés à l'aide de la représentation supervisée, qui dépendent du classifieur construit au mois T, sont beaucoup plus stables dans le temps (Figures 5 et 6 par comparaison aux Figures 3 et 4). On sait par ailleurs que les clients présents dans un cluster ont des scores de churn proches.

4.4 Discussion - Un clustering contraint par la proximité entre scores

L'utilisation d'une représentation supervisée issue de discrétisation ou de groupage supervisé permet la majoration établie équation 5. Cette majoration donne la garantie que si on utilise l'algorithme des k-moyennes, à l'aide de la norme L1 on obtiendra des clusters où deux individus proches dans la représentation supervisée seront proches au sens de leur probabilité d'appartenance à la classe cible. Cependant cette majoration indique seulement que $\Delta^1(D, D') \leq d^1_{NB}(D, D')$. Si donc deux instances D et D' sont très éloignées dans l'espace supervisé on a uniquement la garantie que la distance entre leurs scores sera plus petite. La distance entre les scores de deux instances éloignées dans la représentation supervisée peut donc être grande.

Il serait donc intéressant, dans la représentation supervisée, de contraindre l'algorithme des k-moyennes à ne regrouper que des instances qui soient au plus éloignées d'une valeur seuil

(notée ϵ). Un algorithme de type Xmeans (Pelleg et Moore, 2000) pourrait être utilisé sous la contrainte de recouper tout cluster où la distance maximale entre deux instances est supérieure à ϵ. Cette contrainte garantirait le fait de n'avoir aucun cluster avec un diamètre supérieur à ϵ et donc d'instance qui a un score supérieur à ϵ au score d'une autre instance. Cette garantie permettrait d'améliorer l'algorithme modifié des k-moyennes proposé dans cet article.

On pourra aussi noter que la représentation supervisée construite avant l'étape de clustering pourrait être utilisée avec d'autres méthodes de clustering. Les cartes de Kohonen qui ont la propriété de conservation de la proximité des données dans l'espace initial pourraient être utilisées.

5 Conclusion

Cet article a montré comment il est possible de réaliser une typologie à l'aide d'une technique de type partitionnement mais qui soit contrainte par la connaissance issue d'un classifieur. On a montré qu'il est possible de construire une représentation supervisée à l'aide d'un classifieur de type naïve Bayes, d'une régression linéaire ou d'une régression logistique. Cette représentation supervisée permet de créer un partitionnement qui conserve la proximité des exemples ayant les mêmes probabilités d'appartenance aux classes cibles. Cette technique a été utilisée avec succès dans un cadre applicatif de scoring de clients. Les résultats expérimentaux montrent son bon comportement en termes de mesure d'AUC mais aussi vis-à-vis du critère imposé de stabilité dans le temps.

Références

Blake, C. L. et C. J. Merz (1998). UCI Repository of machine learning databases. `http://archive.ics.uci.edu/ml/` visité pour la dernière fois : 15/09/2010.

Boullé, M. (2007). Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research 8*, 1659–1685.

Bradley, P. S., O. L. Mangasarian, et W. N. Street (1997). Clustering via concave minimization. In *Advances in Neural Information Processing Systems -9*, pp. 368–374. MIT Press.

Creff, N. (2011). Clustering à l'aide d'une représentation supervisée. Master's thesis, Epita, 14-16 rue Voltaire 94276 Kremlin Bicêtre Cedex.

Féraud, R., M. Boullé, F. Clérot, F. Fessant, et V. Lemaire (2010). The orange customer analysis platform. In *Proceedings of the 10th Industrial Conference on Data Mining*, Berlin, Germany, pp. 584–594. Springer Verlag.

Guyon, I., V. Lemaire, M. Boullé, G. Dror, et D. Vogel (2009). Analysis of the KDD cup 2009 : Fast scoring on a large orange customer database. *JMLR : Workshop and Conference Proceedings 7*, 1–22. Data available on `http://www.kddcup-orange.com`.

Har-peled, S. et S. Mazumdar (2003). Coresets for k-means and k-median clustering and their applications. In *In Proc. 36th Annu. ACM Sympos. Theory Comput*, Chicago, Illinois, USA, pp. 291–300.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *Pacific Asia Knowledge Discovery and Data Mining Conference*, pp. 21–34. Singapore : World Scientific.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Discov. 2*, 283–304.

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 *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1 –4.

Kaufman, L. et P. J. Rousseeuw (1990). *Finding Groups in Data : An Introduction to Cluster Analysis*. John Wiley.

Langley, P., W. Iba, et K. Thompson (1992). An analysis of Bayesian classifiers. In *Proceedings of the tenth National Conference on Artificial Intelligence*, San Jose, California, USA, pp. 223–228. MIT Press.

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

Meila, M. et D. Heckerman (1998). An experimental comparison of several clustering and initialization methods. In *Machine Learning*, pp. 386–395.

Park, H.-S. et C.-H. Jun (2009). A simple and fast algorithm for k-medoids clustering. *Expert Syst. Appl. 36*(2), 3336–3341.

Pelleg, D. et A. W. Moore (2000). X-means : Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, San Francisco, California, USA, pp. 727–734. Morgan Kaufmann Publishers Inc.

Summary

When the marketing service has to contact customers to propose them a product, the probability that these customers will buy this product is calculated beforehand. This probability is calculated using a predictive model. The marketing service contacts then those having the highest probability of buying the product, the strongest appetency. In parallel and before the commercial contact it may be interesting to realize a typology of the customers who will be contacted. The idea is to propose differentiated campaigns by group of customers. This article shows how it is possible to force the typology, realized using a k-means type algorithm, to respect the nearness of the customers as refers to their appetency score.