Nicolas Creff
Promotion 2011
Majeure SCIA

Du 1er février au 31 juillet 2011

Rapport de Stage

Titre : Clustering à l'aide d'une représentation supervisée

Sujet : Personnalisation de scores à l'aide de la technique des k-moyennes

Adviser : Vincent Lemaire

Sommaire

Introduction

1) Le sujet de stage initial

Le sujet du stage est la personnalisation des scores de campagnes à l'aide d'une technique de type k-moyennes.

Lorsqu'on désire contacter un client pour lui proposer par exemple 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 est en fait calculée à l'aide d'un modèle prédictif (dans notre cas un classifieur naïf de Bayes) pour un ensemble de clients (le périmètre de la campagne). Les clients sont ensuite triés dans l'ordre décroissant de leur probabilité d'appétence. Le service marketing ne contacte ensuite que les plus appétant, 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 différentiées par segment.

Ces segments sont actuellement réalisés tous les mois durant une période donnée. D'un mois à l'autre ces segments sont susceptibles d'être très différents. Ce qui pose un problème de stabilité des segments au cours du temps. On pense que le fait de travailler sur une représentation supervisée des données pourrait résoudre ce problème.

Au cours du stage, quelques changements ont été apportés au sujet initial à la suite de la présentation du stage à l'équipe de recherche. L'équipe a exprimé ses doutes vis-à-vis de la solution envisagée, et plus particulièrement sur une des hypothèses nécessaires à sa mise en place, à savoir la stationnarité des données clients. C'est-à-dire le fait que les données soit relativement similaire au cours du temps. L'équipe a proposé d'autres méthodes qui pourraient être envisagées pour répondre à la problématique industrielle.

Le sujet à donc été adapté pour prendre en compte ces remarques, notamment en séparant la partie étude de l'apport de l'utilisation d'une représentation supervisée lors d'un clustering, du problème industriel.

La première partie du stage consistera à décrire les aspects organisationnels du stage. Nous aborderons le découpage du stage, le respect de délais, la nature des points de contrôle et la gestion des crises.

La deuxième partie du stage consistera à étudier l'influence d'une représentation supervisée lors de l'utilisation d'une technique de clustering basée sur le partitionnement (comme les k-moyennes et les k-médoïdes). Les différentes techniques de clustering basées sur le partitionnement seront abordées mais aussi le paramétrage de ces techniques, les types de représentation qui seront étudiés, les expériences qui seront mises en place, et leurs résultats.

La troisième partie consistera à appliquer les résultats de l'étude de la première partie dans le cadre d'une problématique industrielle. Dans un premier temps, nous expliquerons le problème industriel, son contexte, la solution actuelle, et le problème existant avec cette solution. Dans un deuxième temps nous envisagerons différentes alternatives, nous mettrons en place une démarche d'évaluation, et nous analyserons les résultats obtenus.

La quatrième partie consistera à réaliser un bilan du stage.

2) Présentation de l'entreprise

France Télécom est la principale entreprise française de télécommunications. Elle emploie près de 167 000 personnes et compte près de 200 millions de clients.

Elle développe trois grandes familles de services commercialisés sous la marque Orange :
- des services de téléphonie fixe qui ont conservé le nom de la marque historique dans certains pays dont la France, la Pologne et le Sénégal
- des services de téléphonie mobile
- des services de communication d'entreprise, via Orange Business Services.

Bien que France Télécom soit née officiellement le 1er janvier 1991, son histoire remonte à plus de deux siècles. C'est en 1792 que le premier réseau de communication français, le réseau de télégraphie optique de Chappe, a vu le jour. Un ministère des Postes et Télégraphes est finalement créé en 1878. En y annexant les services du téléphone, le ministère des P&T devient celui des PTT en 1923.

En 1941, une Direction Générale des Télécommunications ou DGT est créée au sein de ce ministère. En 1988, la DGT se sépare des PTT et prend le nom de "France Télécom". Il faut toutefois attendre le 1er janvier 1991 pour que France Télécom devienne un exploitant autonome de droit public. Pour préparer l'ouverture des Télécommunications à la concurrence au 1er janvier 1998, France Télécom passe du statut d'exploitant public à celui de société anonyme dont l'État français est le seul actionnaire, en juillet 1996. En 1997, l'entreprise ouvre son capital et est cotée sur les marchés boursiers de Paris et New York.

En 2000, elle fait l'acquisition de l'opérateur mobile britannique Orange, au prix de 40 milliards d'euros, pour en faire une filiale nommée Orange SA. Elle constitue alors le deuxième réseau mobile européen. De nombreuses autres acquisitions de sociétés (GlobalOne, Equant, Internet Telecom, Freeserve, EresMas) lui permettent de devenir le quatrième plus grand opérateur mondial. En septembre 2004, l'Etat français cède une partie de ses actions et France Télécom devient une entreprise privée. En 2006, la plupart des activités du groupe passent sous la marque et le logo Orange. C'est le cas des services Internet, de Télévision et de téléphonie mobile ainsi que des services numériques.

Aujourd'hui, France Télécom continue d'élargir ses activités pour offrir une gamme complète de prestations dans l'audiovisuel et le multimédia, ainsi que de nouveaux produits et services : vente de contenus (musique, cinéma, téléchargement), E commerce et publicité en ligne, domotique et Téléassistance aux malades.

France Télécom possède plusieurs sites de recherche et développement et le site de Lannion est l'un des deux plus gros avec 1000 personnes. Chacun des sites travaille dans de nombreux domaines de recherche autour des problématiques de l'entreprise, ou des connaissances liées à ses problématiques.

L'équipe dans laquelle s'effectue le stage est l'équipe PROF (PROFiling et datamining). Elle est spécialisée dans la gestion d'informations, de statistiques et dans le développement d'algorithmes performants de datamining. Ces algorithmes sont utilisés dans de nombreuses applications comme le profiling, le scoring et la recommandation. Cette équipe transmet son expertise aux équipes opérationnelles. Elle les aide en leurs proposant des solutions techniques, et des choix stratégiques (méthodologie, architecture, technologies internes et externes…) Cette équipe est très mature dans les techniques d'apprentissage automatique, et dans les techniques de partitionnement.

Parmi ces techniques on retrouve des algorithmes de classification comme le classifieur Bayesien Naïf Sélectif (SNB) parmi les plus performants par rapport à l'état de l'art. On retrouve aussi dans l'équipe, des personnes qui travaillent sur les réseaux de neurones

(perceptron, cartes de Kohonen), sur les arbres de décisions, les règles d'associations, le clustering de courbes.

Toutes ces techniques sont généralement utilisées dans le cadre de problématiques similaires à la notre.

L'équipe de recherche travaille en relation avec le secteur CRM de l'entreprise (Customer Relationship Management) et plus particulièrement avec la cellule Score, la partie de l'entreprise chargée de noter et sélectionner les meilleurs clients avant de donner les résultats de leurs analyses au service marketing qui se chargera de les contacter.

3) Connaissances antérieures

Vincent Lemaire, mon maitre de stage avait de nombreuses connaissances en rapport avec le sujet du stage. Il effectue des cours sur les processus de datamining et sur les cartes de Kohonen.
De plus il a publié plusieurs articles sur le clustering comme *Data Mining Exploration, Sélection, Compréhension* [22], en 2008 ou *Une nouvelle stratégie d'Apprentissage Bayésienne* [23]*,* en 2010.
Et des articles sur l'utilisation d'une méthode de représentation supervisée des connaissances pour améliorer la classification comme *Elaboration d'une représentation basée sur un classifieur et son utilisation dans un déploiement basé sur un k-ppv* [24], en 2010.

Il connait donc bien les méthodes qui vont être utilisées durant mon stage. En particulier les méthodes de représentation de connaissances supervisées et les méthodes de classification. Il connait un peu moins les différentes méthodes possibles de clustering basées sur le partitionnement, les différentes méthodes d'initialisation possibles pour ce genre de méthodes. C'est pourquoi dans ce stage le travail consistera en partie à approfondir les connaissances sur ces sujets.

Lors de mon cursus à l'EPITA au sein de la majeure SCIA, j'ai eu l'occasion d'avoir des cours de Machine Learning, dans lequel j'ai étudié la technique de clustering des k-moyennes. Cette technique fait partie des techniques de clustering basées sur le partitionnement et constitue une base pour réaliser un stage sur ce type d'algorithme.

De plus j'ai pu étudier plusieurs algorithmes de classification dont le Naïf Bayes. Dans ce stage nous allons utiliser une variante de cet algorithme lors de la création d'une représentation supervisée des données, il était donc important de connaître le fonctionnement de cet algorithme pour pouvoir comprendre la représentation que nous allons utiliser.

Je maitrise tous les langages de programmation qui me seront utiles pour ce stage comme Matlab, python, java.

Ce stage est une occasion de découvrir l'utilisation des méthodes de clustering au sein d'une grande société telle que France Télécom.

De plus c'est la possibilité d'apprendre et de comprendre certaines problématiques liées au secteur des CRM (Customer Relationship Management).

Et ce stage pourra m'apporter des éléments de réflexion pour le choix de mon premier emploi, et mon avenir professionnel comme le type d'emploi, la taille de société ou encore le domaine dans lequel je pourrais envisager de travailler.

Enfin cela me permettra d'approfondir mes connaissances en techniques de clustering, et de valider mon stage de fin d'études.

4) Les moyens mis à disposition

Orange est une grande société qui met des moyens conséquents pour ses employés.
Pour réaliser mes calculs et stocker toutes les données de ces calculs, les moyens matériels suivants ont été mis à ma disposition :

- Un PC dual-cores 4 gHz sur Windows xp 32 bits
- L'accès à un serveur de stockage de données de plusieurs Téra octets de données.
- L'accès à un PC Octo-cores sur Windows 7, 64 bits.

Pour effectuer la partie de recherche du stage, il m'a été nécessaire d'avoir accès à différent types de documentation. L'entreprise m'a permis d'avoir accès à :

- la bibliothèque dédiée aux livres techniques.
- des bibliothèques numériques comme l'ACM.

Les bibliothèques numériques comme l'ACM permettent d'avoir accès à divers articles, qui sont parfois uniquement disponibles dans ce type de bibliothèque. Et la bibliothèque technique du site de Lannion m'a permis d'avoir accès à des ouvrages de références dans le domaine du clustering.

L'équipe est en relation avec une secrétaire pour aider à effectuer certaines taches administratives internes, permettre d'avoir accès à des fournitures de bureau, ou encore aider préparer certaines missions notamment au niveau des transports.

L'équipe a toujours été disponible, et grâce à elle j'ai toujours pu obtenir les informations que je ne pouvais trouver ailleurs.

Le maitre du stage a été là lors que j'ai eu besoin de plus d'informations sur le stage. Et il a su me rediriger vers les personnes les plus à même de répondre lorsqu'il ne le pouvait pas.

5) Les retombées potentielles du stage

Le sujet du stage vise à étudier les problèmes d'une méthode permettant de réaliser des campagnes marketing personnalisées pour la cellule Score de l'entreprise. Pour le moment, la méthode actuelle a été mise de côté car jugée pas assez stable pour être utilisée.

Ce stage vise donc à étudier des alternatives possibles à la méthode actuelle, en la prenant comme référence.

Si ce stage permet de montrer qu'il existe une méthode pour obtenir de meilleurs résultats que la méthode actuelle en particulier sur la stabilité alors le stage pourra aboutir sur un projet de création d'un nouveau logiciel pour réaliser des campagnes marketing personnalisées.

Une fois ce logiciel crée, l'utilisation de ce type de méthodes pourra être généralisée au sein de l'équipe marketing.

Partie I) Aspects organisationnels

1) Découpage du stage

Le stage a été découpé en deux parties :
- L'étude de l'influence d'une représentation supervisée lors d'un clustering
- L'application de cette étude à la problématique industrielle du stage.

La première partie de ce stage a été découpée en sous-partie :

- Recherche des différentes méthodes de clustering basées sur le partitionnement
- Recherche sur le paramétrage des méthodes de clustering
- Réunion de présentation du stage
- Implémentation des algorithmes de clustering
- Tests sur les bases de l'UCI
- Commentaires et critiques

La deuxième partie du stage a été découpée dans les sous-parties suivantes :

- Présentation du contexte et de sa problématique
- Présentation de la solution proposée
- Obtention des bases de clients
- Implémentation des indicateurs de qualité
- Tests sur les bases clients
- Commentaires et critiques

L'organisation de ces sous-parties du stage au cours du temps est décrite ci-dessous dans le tableau 3.

Mois	Février				Mars				Avril				Mai				Juin				Juillet			
Semaines	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Partie 1																								
Recherche des Méthodes																								
Recherche du paramétrage																								
Implémentation des algorithmes																								
Présentation du stage à l'équipe																								
Test sur les bases de l'UCI																								
Conclusion et commentaire																								
Partie 2																								
Présentation de la problématique																								
Présentation de la solution proposée																								
Récupération des bases																								
Implémentation des critères de qualités																								
Tests sur les bases clients																								
Conclusion et commentaire																								

Tableau 3 : Diagramme de Grant du découpage du stage.

Description des sous-parties :

Partie 1:

- **Recherche sur les différentes méthodes de clustering basées sur le partitionnement**

Cette partie consiste à étudier et comprendre les différentes méthodes de clustering basées sur le partitionnement. Cette étude vise à nous permettre d'avoir les éléments nécessaires pour faire le choix des techniques de clustering qui seront testées.

Pour réaliser cette partie de nombreux articles sur le sujet sont mis à ma disposition. D'autres articles sont également disponibles dans les bibliothèques du groupe.

Le livrable associé à cette recherche est la rédaction d'une page ou deux sur chacun des algorithmes étudiés. Les pages écrites doivent expliquer les algorithmes, en détailler la complexité, puis décrire explicitement les algorithmes.

Le temps prévu pour cette partie est de 2 semaines.

- **Recherche sur le paramétrage des méthodes de clustering**

Cette partie consiste à étudier et comprendre les différentes techniques pour paramétrer les méthodes de clustering basées sur le partitionnement. Le but de cette étude est de nous permettre de réaliser le choix du paramétrage de la méthode de clustering qui sera utilisée dans un contexte industriel.

Le livrable attendu est une présentation des différentes techniques d'initialisations, les différentes fonctions de distances à disposition, le réglage de la valeur de k, une explication des différentes représentations de données à notre disposition.

Le temps prévu pour cette partie est de 2 à 3 semaines.

- **Réunion de présentation du stage**

Cette réunion se positionne après avoir réalisé des recherches sur le sujet de stage, et après avoir effectué les choix déterminants du stage.
Le but de cette réunion est de présenter à l'équipe le sujet du stage, ses objectifs, et les méthodes retenues.

Cette réunion permet également d'avoir des remarques et des critiques de la part de l'équipe sur les choix effectués et les méthodes retenues.

- **Implémentation des algorithmes de clustering**

Les éléments suivants vont être implémentés dans cette partie :

- L'algorithme des fast k médoïdes comme décrit dans l'article « A simple and fast algorithm for k-medoids clustering » [13]
- une méthode des k-moyennes où l'on recherche les vrais éléments les plus proches des centres. (à l'aide de la méthode des k-moyennes de Matlab).
- les différentes techniques d'initialisation retenues pour les tests.
- des scripts khiops, kawab, et kxen, pour réaliser les tests de manières automatiques.
- Implémentation des critères de qualité de la première partie.

De plus pour tester la méthode PAM (Partition Around Medoids), nous rechercherons une implémentation de cette méthode.

Les livrables attendus sont les codes sources des algorithmes.

Le temps prévu pour cette partie est de 1 mois.

- **Tests sur les bases de l'UCI**

Cette partie est l'application des algorithmes choisis et implémentés dans les parties précédentes sur les bases de l'UCI. Une description des différentes bases est disponible en Annexe 3 et 46.

Il est attendu de faire une présentation des tests à mettre en place, puis de les réaliser et de présenter les résultats.

Le temps prévu pour cette partie est de 2 mois.

- **Commentaires et critiques**

Nous commenterons dans cette partie les différents résultats obtenus. Nous essayerons de conclure l'étude réalisée dans cette première partie du stage. Nous verrons quels enseignements nous retenons de notre étude, quels choix nous avons fait, et quelles méthodes nous choisissons pour la partie suivante.

Le temps prévu pour cette partie est de 3 semaines

Partie 2:

- **Présentation du contexte et de sa problématique**

Il est attendu dans cette partie de rédiger quelques pages pour présenter le contexte du stage, la méthode actuellement utilisée, et enfin le problème avec cette méthode.

- **Présentation de la solution proposée**

Nous proposerons ensuite une solution au problème industriel à partir de l'étude effectuée lors de la première partie du stage.

- **Obtention des bases de clients**

L'obtention des bases clients est un point essentiel pour pouvoir ensuite réaliser des tests dessus. Nous les obtiendrons par Nicolas Voisine. Nicolas Voisine est l'un des membres de l'équipe PROF. Il est en charge du projet BUSI en rapport avec le secteur CRM de l'entreprise. Il participe aussi à la partie d'anticipation de besoins faite pour la cellule Score.

- **Implémentation des indicateurs de qualité**

Nous implémenterons les différents critères de stabilité tel que définis dans la partie III section 1.3

Les livrables attendu à la fin de cette partie sont les codes sources des différents critères.

- **Tests sur les bases clients**

Nous décrirons la démarche de tests qui sera retenu pour répondre à la problématique industrielle, puis nous la réaliserons et nous présenterons les résultats.

- **Commentaires et critiques**

Une fois avoir obtenu les résultats, nous les analyserons. Nous espérons pouvoir montrer que la méthode proposée permet d'obtenir de meilleurs résultats que la méthode actuelle.

2) Respect des délais et critiques du découpage

Ce stage inclut une partie de recherche sur les différents algorithmes à notre disposition. Pour réaliser un planning strict à respecter, il faut être capable d'estimer correctement chacune des taches. Le temps que va prendre la recherche sur une question donnée est difficilement estimable. Ceci explique qu'aucun planning n'a été effectué pour ce stage.

Le découpage réalisé a été pertinent, il correspond bien aux différentes taches à réaliser. Ce découpage représente bien les dépendances entre les taches.

Les délais ont globalement été respectés. L'implémentation des algorithmes a pris un peu moins de temps que prévu et les tests ont pris plus de temps que prévu.

La partie de tests était très ambitieuse, on avait prévu de faire des tests sur 42 bases de l'UCI, en 10-foldcross validation avec des valeurs de k différentes allant de 2 à $\sqrt{n}$.

Le fait d'avoir choisi une partie de tests si ambitieuse a fait que le temps nécessaire pour réaliser une série de tests a été très long. En effet pour réaliser une série de tests d'un k-moyennes avec une méthode d'initialisation, il faut entre 1 et 2 semaines.

Pour réduire le temps, on a décidé de lancer la plupart des tests en parallèles ce qui a permis de réduire le temps global des tests.

Au début de la phase de test, une erreur a été faite dans l'algorithme de réalisation des tests. Cela a entrainé la création de résultats inutilisables. Nous avons donc corrigé le problème et recommencé les tests. Cette erreur a fait perdre 1 à 2 semaines sur la période de tests.

Malgré une partie de tests un peu longue les délais sur l'ensemble du stage devraient être respectés.

3) Nature et fréquence des points de contrôle

Durant mon stage, j'ai participé à de nombreuses discussions avec mon maître de stage. Le plus souvent je préparais des questions à poser, je faisais un petit bilan de ce que j'avais fais depuis la dernière discussion et sur ce qu'il restait à faire. Mon maître de stage était la pour contrôler de la qualité du travail effectué, et l'état d'avancement du stage.

Environ deux mois et demi après le début du stage, j'ai présenté les objectifs de mon stage à l'équipe. Elle a su faire des remarques constructives sur le stage.

Une réunion a été organisée pour répondre à certaines de mes questions sur la solution actuelle, et sur la manière de réaliser les campagnes markéting.

4) Gestion des crises

Après la réunion avec l'équipe, nous nous sommes remis en cause sur le sujet du stage et sur son organisation. Au cours d'une discussion juste après la réunion, nous avons fait le point sur les propositions de l'équipe. Nous avons décidé de continuer dans la direction choisie, et d'étudier les propositions d'alternative de l'équipe en fin de stage si le temps nous le permet.

Lors de la phase de tests, nous avons dû faire face à deux problèmes. Le premier est dû à une petite erreur réalisée dans l'algorithme de réalisation des tests. Nous avions utilisée une fonction qui ne faisait pas ce qu'elle était supposée faire. Une fois avoir détecté le problème, nous avons décidé de refaire la fonction en question. Cette erreur a été assez vite corrigée, mais elle nous a quand même fait perdre une à deux semaines de calculs.

Le deuxième problème que nous avons rencontré lors de la phase de tests est le temps de réalisation d'une série de tests. En effet une série de tests dure environs 2 semaines et la génération des indicateurs de qualité sur les résultats de ces tests dure environ 1 semaine.

Après avoir observé la vitesse de réalisation des tests, nous avons décidé de réduire le nombre de bases de données sur lesquels ils seront réalisés, ainsi que de réduire le nombre de valeurs différentes de k testées.

Pour résoudre ces problèmes, j'ai discuté avec mon maître de stage, et pour chacun d'entre eux nous avons trouvé un plan d'action adapté.

Au final aucun de ces petits problèmes ne devrait avoir d'influence sur le stage dans sa globalité.

Partie II) Clustering basé sur une représentation supervisée

1) Notations

X est l'ensemble de données.

C est l'ensemble des centres.

c_i est un centre avec $i \in [\![1, k]\!]$

x_j est un élément qui n'est pas un centre.

$D(x_j)$ est la distance entre l'élément x_j et son plus proche centre.

$d(x_j, x_h)$ est la distance entre l'élément x_j et x_h.

$E(x_j)$ est la distance entre l'élément x_j et son second plus proche centre.

k est utilisé lors des algorithmes de clustering basés sur le partitionnement pour désigner le nombre de partitions souhaitées.

2) Les différentes méthodes de clustering basées sur le partitionnement

Dans cette partie, nous allons étudier les différentes techniques de clustering basées sur le partitionnement. Nous verrons dans cette étude la méthodes des k-moyennes, des k-médianes, des k-médoïdes, des k-modes, et des k-prototypes.

2.1) K-moyennes

L'algorithme des k-moyennes [1] est utilisé pour regrouper les éléments d'un ensemble de données en k clusters autour d'un centre de gravité (centroïde). En général on ne connait pas le nombre de classes que contient l'ensemble de données.

La méthode des k-moyennes se déroule ainsi :

1) On tire au hasard k centres de gravité. Ces centres peuvent être tirés parmi les exemples de la base d'apprentissage.

2) On associe chaque exemple de l'ensemble de données au centre de gravité le plus proche. Après cette étape tous les exemples ont été affectés à un centre.

3) Chaque centre est mis à jour à l'aide de la moyenne des exemples qui lui sont associés.

4) Puis on recommence les étapes 2 et 3 jusqu'à ce que les exemples affectés à un centre ne changent plus.

Le pseudo-code de cet algorithme se trouve en annexe 5.

La complexité de cet algorithme est en $O(kni)$
Avec n est le nombre de données dans l'ensemble de données.
Et i le nombre d'itérations.

Il existe une variante de cet algorithme, qui s'appelle le kmean++ [2]. Cette variante est une amélioration de l'algorithme et consiste à changer la manière d'initialiser les centres de gravité.

Elle est détaillée dans la partie sur les différentes techniques d'initialisation (voir partie II section 2.1).

2.2) K-médianes

L'algorithme des k-médianes est similaire à l'algorithme des k-moyennes en utilisant la médiane au lieu de la moyenne.

Cet algorithme est souvent confondu avec celui des k-médoïdes (aussi appelé k-médianes discrètes).

On définit la médiane (géométrique) d'un ensemble de points comme le point artificiel qui minimise :

$$\sum_{x \in X} \|c - x\|_2$$

Avec x qui décrit les points de l'ensemble.

Il n'y a aucune formule analytique pour calculer la médiane géométrique exacte à partir d'un ensemble de points. Par conséquent en pratique on cherche à approximer cette médiane.

La recherche de la médiane géométrique d'un ensemble de points est aussi connue comme le problème Fermat-Weber [3] en utilisant des poids de 1 ou encore le problème d'emplacement d'installations (facility location problem) en dimension 2 avec des emplacements non-déterminés et pas de coût d'ouverture.

On recherche souvent la solution approchée avec l'algorithme de Weiszfeld [4].

Certaines personnes proposent des variantes, comme le mélange de cet algorithme avec de la programmation linéaire [5], ou encore Sanjeev Arora qui propose une méthode basée sur les arbres quaternaires [6].

C'est cette dernière méthode qui semble être la plus optimisée tout en conservant intacte la définition du problème. Et la complexité de cet algorithme est de l'ordre de $n^{0(\log(n/\varepsilon)^{d-1})}$ itérations pour calculer une médiane, avec une ε fixé, d la dimension des données, et n le nombre de données.

L'algorithme des k-médianes semble être le plus performant pour trouver une excellente solution au problème de partitionnement en k clusters, en revanche sa complexité est vraiment très élevée et il ne peut pas être appliqué sur de grande bases de données.

On considère dans ce document donc que la méthode des k-médianes se définit ainsi :

1) On initialise k médianes (par exemple aléatoirement)

2) On associe chaque élément de l'ensemble de données à la médiane la plus proche de lui.

3) Puis pour chaque ensemble associé à une médiane, on recalcule une approximation de sa médiane.

4) On recommence les étapes 2 et 3 à nouveau jusqu'à ce que les points associés à une médiane ne changent plus.

Le pseudo-code de l'algorithme des k-médianes se trouve en annexe 6.

Pour illustrer la différence entre une médiane et une moyenne, la figure1 présente une itération de la méthode des k-moyennes et une de celle des k-médianes.

L'image de gauche correspond à l'initialisation, et l'image de droite correspond au calcul d'une itération des k-moyennes *(Δ),* et des k-médianes *(∇)* (k = 2).

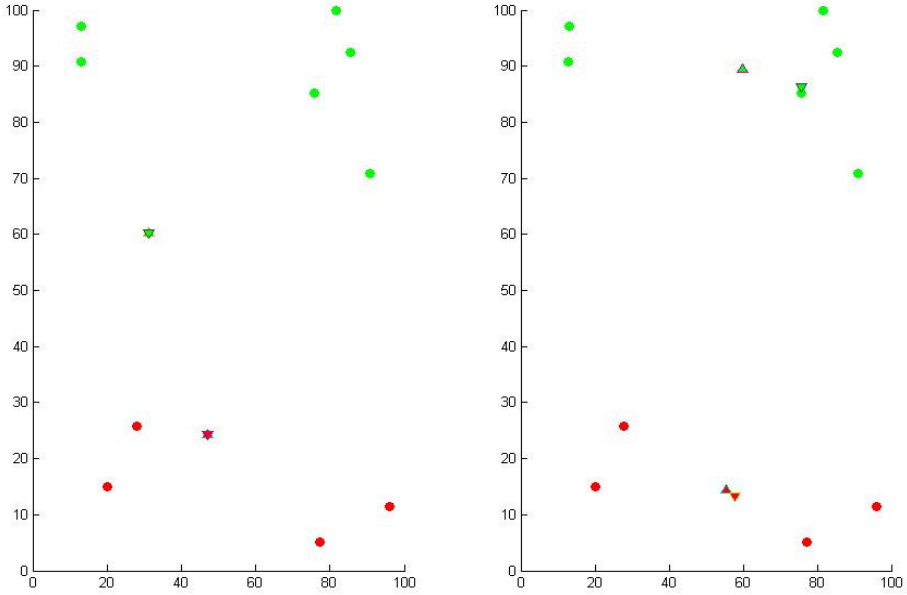


Figure 1 : La différence entre une médiane (∇) et une moyenne (Δ).

Cluster rouge	
95.9528	11.5125
77.3462	5.1341
20.0717	15.0624
27.8820	25.8053

Cluster Vert	
90.8916	70.8750
85.3390	92.5449
81.6746	99.9538
75.8808	85.1291
12.9666	97.1226
12.9274	90.7286

Médianes (triangles vers le bas)		Moyennes (triangles vers le haut)	
57.8138	13.2967	55.3132	14.3786
75.8183	86.2503	59.9467	89.3923

On remarque que les médianes (triangles vers le bas) se positionnent vers l'endroit où il y a le plus de points, pour minimiser la somme des distances tandis que les moyennes (triangles vers le haut) sont au centre de leur cluster.

2.3) K-modes

La méthode des k-modes est similaire à celle des k-moyennes mais adaptée pour les objets catégoriques, ce sont des objets qui ne contiennent pas de valeurs numériques mais des chaînes par exemple.

Le principe de la méthode des k-modes se déroule ainsi :

1) On initialise k modes (aléatoirement par exemple).

2) On associe chaque objet de l'ensemble de données au mode le plus proche ou le plus similaire à lui.

3) Puis on recalcule les modes de chaque ensemble, à partir de la fréquence des champs des objets.

4) Et on recommence les étapes 2 et 3, jusqu'à ce que les modes ne changent plus.

Le pseudo-code de l'algorithme des k-médianes se trouve en annexe 7.

La fonction de distance utilisée, retourne une valeur en fonction du nombre de champs identiques (La distance de Hamming).

Pour trouver un mode, à partir d'un cluster, idéalement on essaie de construire l'objet qui minimise la somme des distances par rapport à tous les objets du cluster.

Et de manière plus simple, on associe à chaque champ du mode la valeur du champ la plus fréquente parmi les éléments de son cluster.

Exemple :
Soit le cluster suivant :
Obj(rond, rouge, petit)
Obj(rond, vert, grand)
Obj(carré, vert, petit)

On obtiendra le mode suivant : Obj(rond, vert, petit)

2.4) K-prototypes

Il existe une variante de l'algorithme des k-modes qui s'appelle l'algorithme des k-prototypes, c'est un algorithme de clustering de type k-moyennes pour des objets de types mixtes qui contiennent à la fois des valeurs numériques et des valeurs catégorielles.

Dans ce cas, on peut utiliser une fonction de distance qui additionne le résultat des deux fonctions de distances, une pour les valeurs numériques et une pour les valeurs catégorielles. La deuxième distance sera par exemple celle utilisée dans l'algorithme des k-mode décrit ci-dessus.

Pour trouver le centre d'un ensemble, idéalement il faudrait trouver l'objet qui minimise les distances par rapport à tous les objets de l'ensemble. Cependant on peut aussi effectuer une moyenne sur les champs numériques et le champ le plus fréquent pour les champs catégoriels.

Les documents suivants [7][8][9] parlent de l'algorithme des k-modes et ont été utilisé pour rédiger cette partie.

2.5) K-médoïdes

La méthode des k-médoïdes [10] est une méthode de partitionnement qui se base sur des médoïdes pour créer des partitions (clusters).

Un médoïde est l'élément d'un ensemble qui minimise la somme des distances entre lui et chacun des autres éléments de cet ensemble.

K représente le nombre de médoïdes et de clusters que l'on souhaite.

L'algorithme PAM (Partionning Around Medoid) [12] se décompose en deux parties. Dans un premier temps l'initialisation et dans un deuxième temps la recherche des meilleurs médoïdes.

Initialisation (BUILD):

1) Le premier centre de gravité choisi est l'élément pour lequel la somme des distances par rapport à tous les autres éléments est la plus petite.

2) On définit chaque nouveau centre c_i comme l'élément parmi tous les éléments non-médoïdes, qui maximise :
$$\sum_{x_j \in X} \max(D(x_j) - d(x_j\ c_i,), 0)$$
3) On recommence en 2 après avoir choisi un nouveau centre c_i jusqu'à en avoir k.

La recherche des meilleurs médoïdes (SWAP)

On considère chaque échange possible entre un centre c_i et un élément x_h (non-médoïde), pour chaque couple (c_i, x_h).

Pour cela on va définir le coût de cet échange: $\mathrm{Cost}(c_i, x_h)$
Et la contribution d'un élément x_j(non-médoïde et différent de x_h) à ce coût: $\mathrm{Cont}(x_j, c_i, x_h)$.

On définit $\mathrm{Cont}(x_j, c_i, x_h)$ de la manière suivante :

a) Si x_j est plus loin de $c_i\ et\ x_h$ que d'un autre centre.
$\mathrm{Cont}(x_j, c_i, x_h) = 0$
b) Si x_j est plus proche de c_i que d'un autre centre (alors $d(x_j, c_i) = D(x_j)$)
   - Si x_j est plus proche de x_h que de son second plus proche centre.
     $\mathrm{Cont}(x_j, c_i, x_h) = d(x_j, x_h) - d(x_j, c_i)$
   - Si x_j est plus loin de x_h que de son second plus proche centre.
     $\mathrm{Cont}(x_j, c_i, x_h) = E(x_j) - D(x_j)$
c) Si x_j est plus loin de c_i mais plus proche de x_h que d'un autre centre.
   $\mathrm{Cont}(x_j, c_i, x_h) = d(x_j, x_h) - D(x_j)$

On définit ensuite $\mathrm{Cost}(c_i, x_h)$:

$$\mathrm{Cost}(c_i, x_h) = \sum_{x_j \in X} Cont(x_j, c_i, x_h)$$

On remarque que la fonction $\mathrm{Cost}(c_i, x_h)$ correspond à une somme de différence. Elle peut donc être nulle par moment.

L'algorithme se déroule ensuite ainsi :

1) Pour chaque couple (c_i, x_h), on calcule $\mathrm{Cost}(c_i, x_h)$.
2) On sélectionne le couple avec le coût minimum.
3) Si ce coût est négatif on procède à l'échange et on recommence en 1
4) Sinon si ce coût est positif ou nulle on arrête.

Algorithme 4 :

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = [c_1, c_2, \dots, c_k]$ L'ensemble des k centres.

Variables :
- $idcentre$: l'id du centre du couple sélectionné
- $idelt$: l'id de l'élément du couple sélectionné

Début

Initialisation (C) // On initialise les centres
Faire
$costmin = \infty$
$idcentre = -1$
$idelt = -1$
Pour $h = 1$ **jusqu'à** k **Faire** // Recherche du meilleurs couple
Pour $i = 1$ **jusqu'à** n **Faire**
Si $Cost(c_i, x_h) < costmin$ **Alors**
$costmin = Cost(c_i, x_h)$
$idcentre = h$
$idelt = i$
Fin Si
Fin Pour
Fin Pour
$swap(x_{idelt}, c_{idcentre})$
Tant que $costmin < 0$
Fin

$Cost(c_i, x_h)$ est la fonction définie plus haut.

$swap(x_{idelt}, c_{idcentre})$ est une fonction qui echange un centre et un élément x_{idelt} devient un centre et $c_{idcentre}$devient un élément de X).

Algorithme 4: L'algorithme PAM (Partition Around Medoids)

La méthode d'initialisation de PAM sera décrite dans la partie concernant les méthodes d'initialisation (voir partie 1 section 4.1).

Il existe plusieurs algorithmes différents pour implémenter la méthode des k-médoïdes.

Le premier est PAM (Partionning Around Medoid), c'est celui qui nous avons décrit en *Algorithme 4*.

La complexité de cet algorithme est en $O(k * (n-k)^2 * i)$
Avec n est le nombre d'instances dans l'ensemble de données.
Et i le nombre d'itérations.

CLARA [12] a ensuite été proposé comme une variante dans laquelle on découpe l'ensemble de données en échantillons et on applique l'algorithme PAM dessus.

La complexité de cet algorithme est en $O\big((ks + k(n-k)\ ) * i\big)$
Avec n est le nombre de données dans l'ensemble de données.
Avec i le nombre d'itérations.
Avec s la taille d'un échantillon.

Algorithme 5 :

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = [c_1, c_2, \dots, c_k]$ L'ensemble des k centres.

Variables :
- S : Un sous-ensemble de l'ensemble de données

Début

S = Sample(X, taille) //créer un échantillon à partir de X et de dimension taille.
C = Pam(S,k)
Fin

Algorithme 5: L'algorithme CLARA

Pour obtenir de meilleurs résultats avec CLARA, il est courant de l'utiliser plusieurs fois.

CLARANS [11] est une autre variante de l'implémentation de la méthode des k-médoïdes. Cet algorithme correspond à un algorithme de descente pour déterminer les k médoïdes, il est lancé plusieurs fois en stockant les meilleurs résultats à chaque fois.

Dans cet algorithme, on définit une solution comme un ensemble de k médoïdes. Et le voisin d'une solution le même ensemble avec un seul médoïde qui change.

Algorithme CLARANS :

Maxneighbor est un paramètre de l'algorithme. Il désigne le nombre maximum de voisins d'une solution qui vont être recherchés.

1) On initialise aléatoirement une solution S.

2) On initialise j = 1.

3) On génère une solution newS voisine de S.

4) On calcule la différence de coût entre newS et S par rapport à la fonction de coût (Cost) définit dans PAM (cf figure 1).

5) Si cette différence est négative on affecte newS à S et on retourne en 2.

6) Sinon on incrémente j de 1 et s'il n'a pas dépassé la valeur maxneighbor on retourne en 3

7) Si j a dépassé la valeur maxneighbor on renvoie le résultat.

Ceci est une version légèrement simplifiée de l'algorithme présenté par Ng & han.
Dans leurs cas, il définissait un paramètre numLocal en plus qui désigne le nombre de fois que l'algorithme (figure 2) doit tourner en stockant les meilleurs résultats à chaque fois.

La complexité de CLARANS est meilleure que celle de PAM et CLARA mais elle de l'ordre de $O(n^2)$.

Algorithme 6 :

Données :

- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = [c_1, c_2, \dots, c_k]$ L'ensemble des k centres.

Variables :

- $newC = [c_1, c_2, \dots, c_k]$ Un nouvel ensemble de centre.

Début

Initialisation (C) // On initialise les centres
$j = 1$
Tant que $j <$ maxneighbor **Faire**
    $newC = voisin(C)$ // Une solution voisine différente que d'un centre.
    Si Cost(C, $newC$) < 0 **Alors**
        $j = 1$
    Sinon
        $j = j + 1$
    Fin Si
Fin Tant que
Fin

La fonction $voisin(C)$ permet à partir de C d'obtenir une solution avec un centre différent.

La fonction Cost (C, $newC$) initialement introduite avec des paramètres c_i et x_h, On prend dans ce cas comme c_i le centre de C qui est différent dans $newC$ et x_h le point qui remplace ce centre.

Algorithme 6: L'algorithme CLARANS

2.6) Choix d'une méthode de clustering

2. 6.1) Complexité

Voici un résumé de la complexité des algorithmes que nous avons étudiés.

Algorithme	Complexité
K-moyennes	$O(kni)$
K-mode/ K-prototype	$O(kni)$ Si on fait des moyennes, ou fréquences
K-mode/ K-prototype	$O(n^{0(\log(n/\varepsilon)^{d-1})})$ Si on fait la somme des distances
K-médianes	$O(n^{0(\log(n/\varepsilon)^{d-1})})$
K-médoïdes(PAM)	$O(k*(n-k)^2*i)$
K-médoïdes(CLARA)	$O\big((ks+k(n-k)\ )*i\big)$
K-médoïdes(CLARANS)	De l'ordre de $O(n^2)$.
K-médoïdes(Fast k-medoids)	$O(kni)$

Tableau 1: Tableau de la complexité des différents algorithmes de clustering étudiés

2. 6.2) Discussion

L'algorithme des k-modes et celui des k-prototypes sont utilisés pour des données de types catégorielles ou mixtes. Les données que nous allons utilisées sont mixtes donc ces algorithmes semblent intéressant à première vue. Néanmoins nous allons utiliser une représentation supervisée de nos données (Nous verrons dans la partie II section 5 comment nous allons obtenir cette représentation)
Cette représentation de nos données aboutira à des données numériques, cela rendra l'algorithme des k-modes et celui des k-prototypes inappropriés à notre problème.

La méthode des k-médianes semble donner les meilleurs résultats, mais les complexités des différents algorithmes pour cette méthode sont beaucoup trop élevées pour que nous puissions utiliser cette méthode.
De plus nous souhaitons obtenir de « vrai clients » ou vrai éléments comme centre de cluster et cette méthode ne le permet pas.

L'algorithme des k-moyennes est un algorithme plutôt rapide comparé à d'autres algorithmes de clustering basé sur le partitionnement, et il donne des résultats relativement satisfaisant. Il donne des éléments artificiels comme centre de cluster, or nous souhaitons obtenir des centres faisant partie de notre ensemble de données. C'est pourquoi l'algorithme des k-moyennes ne semble pas adapté à notre problème.

On envisage de tester une variante basée sur l'algorithme des k-moyennes, qui consiste à appliquer un algorithme des k-moyennes puis à rechercher ensuite les éléments de l'ensemble de données les plus proches des centres et de les redéfinir comme centres. Cela devrait nous permettre d'une part de bénéficier de la rapidité de cet algorithme, et d'autre part de pouvoir obtenir de « vrai client » comme centre de cluster.

A des fins de comparaison, nous allons tester la méthode des k-médoïdes qui est supposée donner de meilleurs résultats que celle des k-moyennes, et permettre d'obtenir des éléments de l'ensemble de données comme centre de cluster.

Il existe plusieurs variantes pour implémenter cette méthode et voici les principaux algorithmes :

L'algorithme PAM est celui qui semble donner les meilleurs résultats, comparé à tous les autres algorithmes implémentant la méthode des k-médoïdes. C'est un algorithme qui a une complexité relativement grande, et une mauvaise capacité à travailler avec une grande quantité de données. Néanmoins cela reste un algorithme de référence de l'implémentation de la méthode des k-médoïdes. Si l'on veut tester la méthode des k-médoïdes, on se doit de tester cet algorithme.

Les algorithmes CLARA et CLARANS sont des variantes de l'algorithme PAM, qui visent à augmenter la vitesse de l'algorithme en s'autorisant à être un peu moins précis. Par conséquent ces algorithmes donnent des résultats bien moins bons que l'algorithme PAM. Nous n'allons pas tester ces deux variantes car nous ne pensons pas qu'elles pourront nous permettre d'obtenir des résultats suffisamment satisfaisants.

Nous sommes très intéressés par les possibilités que pourraient offrir le fast algorithme for k-medoids créé par Hae-Sang Park et Chi-Hyuck Jun [13]. En effet il est supposé avoir une complexité similaire à celle de l'algorithme des k-moyennes, tout en ayant des performances proches de celles de Pam. Nous souhaitons vérifier les résultats présentés dans l'article qui l'introduit en testant cet algorithme.

Donc suite à notre analyse des différents algorithmes à notre disposition pour mettre en place une méthode de clustering basée sur le partitionnement, nous avons choisi de tester les algorithmes suivant :

- K-moyennes avec recherche des centres.
- PAM (Partition Around Medoid)
- Fast k médoïdes

3) Paramétrage de la méthode de clustering

3.1) Choisir la méthode d'initialisation.

L'initialisation des algorithmes de clustering basés sur le partitionnement semble influencer 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 [16], il faut donc que nous choisissions une « bonne » méthode d'initialisation. Le but de cette partie est de faire un état de l'art des techniques existantes en matière d'initialisation. Certaines des techniques vues ici seront choisies pour être comparées.

Une partie de ces méthodes d'initialisation utilisées pour l'algorithme des k-médoïdes ou celui des k-moyennes ont été étudiées notamment dans le papier de Hae-Sang Park et Chi-Hyuck Jun [13], celui de Anna D. Peterson, Arka P. Ghosh and Ranjan Maitra encore en révision [14], et celui sur l'algorithme des k-médianes par A. Juan and E. Vidal [15].

Voici un petit état de l'art sur les différentes méthodes d'initialisation, ainsi qu'une comparaison de la complexité de leur implémentation:

3. 1.1) Méthode 1 (Aléatoire)

La plus simple des méthodes est d'initialiser les centres aléatoirement.

On a k centres à initialiser donc la complexité de cet algorithme est 0(k).
Pour utiliser correctement cette méthode d'initialisation il faut lancer plusieurs fois l'algorithme des k-moyennes ou des k-médoïdes en gardant la solution qui minimise le critère d'erreur choisi.

Cette méthode d'initialisation est une méthode de référence. Toutes les personnes introduisant de nouvelles méthodes d'initialisation se comparent à celle-ci. Nous allons tester cette méthode à titre de référence.

Algorithme 7 :

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres.

Pour i = 1 **jusqu'à** k **Faire**
$x =$ Un élément tiré aléatoirement sur X
$C = C \cup \{x\}$
$X = X \setminus \{x\}$
Fin Pour

Algorithme 7: Algorithme de l'initialisation aléatoire

3. 1.2) Méthode 2 (K-means++)

On initialise le premier centre au hasard puis on calcule les autres de la manière suivante :

On pose D(x) la distance entre un point x et son centroïde le plus proche.

1) On calcule pour chaque point x' qui n'est pas un centre la probabilité suivante :
$$\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$$
2) On tire un centre c_i parmi les x' suivant cette probabilité.
3) Et on recommence ces 2 étapes jusqu'à ce que l'on ait placé tous les centres.

On aura donc plus de chance de tirer un point éloigné des centres déjà présents.

Ce type de tirage aléatoire correspond à faire un tirage suivant une loi de probabilité pour des variables discrètes définies par leurs propres probabilités de tirage.
Et la somme cumulée des probabilités $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$ définie la fonction de répartition caractérisant cette loi.

Pour tirer un élément suivant la probabilité $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$ de manière simple on peut procéder de la manière suivante:

1) On tire un nombre aléatoire n_x entre 0 et $\sum_{x \in X} D(x)^2$
2) On initialise une variable cumul_som = 0 et i = 0
3) On ajoute $D(x_i)^2$ à cumul_som
4) Si n_x est supérieur ou égal à cumul_som, on incrémente i et on retourne en 3

5) Sinon (n_x est inférieur à cumul_som) on s'arrête et on retourne x_i le nombre tiré.

(Par sécurité on peut aussi limiter i < n mais théoriquement ce n'est pas utile)

Commentaires de cette méthode :

La complexité de la recherche de D(x) pour tous les éléments ainsi que le calcul de la somme des D(x) en même temps au fur et à mesure est de 0(k'n) pour les k' centres déjà défini.
Puis le tirage d'un élément avec la probabilité $\frac{D(x\prime)^2}{\sum_{x \in X} D(x)^2}$, correspond au pire à 0(n).

Donc l'initialisation de l'algorithme des kmeans++ est au pire de $0(\frac{(k-1)*(k-2)}{2} * n + n)$, et elle est donc de l'ordre de $0(k^2n)$

Algorithme 8:

Données :

- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).
- $D = [D_1, D_2, \dots, D_i, .., D_n]$, D_i : Distance de x_i à son plus proche centre.

Début

$x =$ Un élément tiré aléatoirement sur X
$C = C \cup \{x\}$
$X = X \setminus \{x\}$

Pour i = 2 **jusqu'à** k **Faire**
  somme_total = 0
  Pour j = 1 **jusqu'à** n **Faire** // Calcul de D
    $mise\_a\_jour(D_j)$
    somme_total = somme_total + $D_j{}^2$
  Fin Pour
  alea = $rand([0: somme\_total])$ //Un nombre réel aléatoire sur $[0: somme\_total]$
  somme_cumul = 0
  Pour j = 1 **jusqu'à** n **Faire** //Tirage d'un nouveau centre
    somme_cumul = somme_cumul + $D_j{}^2$
    Si alea < somme_cumul **Alors**
      $C = C \cup \{x_j\}$
      $X = X \setminus \{x_j\}$
      break
    Fin Si
  Fin Pour
Fin Pour
Fin

$mise\_a\_jour(D_j)$ D_j est la distance entre x_j et son plus proche centre. Donc cette fonction remet à jour la valeur de cette distance par rapport à tous les centres définis (en particulier les nouveaux).

On utilise dans cet algorithme un *break* qui est très fortement déprécié dans l'écriture d'algorithmes mais qui est utilisé ici pour un souci de lisibilité, il peut être remplacé par un « Tant que » et des booléens.

Algorithme 8: Algorithme de l'initialisation kmeans++

3. 1.3) Méthode 3 (Pam)

1) Le premier centre de gravité choisi est l'élément x_i pour lequel la somme des distances par rapport à tous les autres éléments est la plus petite.
   Donc
$$\min(\sum_{j=1}^{n} d(x_i, x_j))$$
2) On définit chaque nouveau centre c_i comme l'élément parmi tous les éléments non-médoïdes, qui maximise :
$$\sum_{x_j \in X} \max(D(x_j) - d(x_j\ c_i,), 0)$$
3) On recommence en 2 après avoir choisi un nouveau centre c_i jusqu'à en avoir k.

La complexité du calcul minimum de la somme des distances (1) entre un point et tous les autres est de 0(n).

Puis la complexité du calcul de la somme (2) pour un c_i est de 0(k'n) pour les k' centres déjà placés (donc cela correspondra à une somme de 1 à k-1 pour tous les centres).
Et le calcul du maximum de cette somme (2) pour chaque c_i est de l'ordre de $0(n^2 * \frac{(k-1)*(k-2)}{2})$
La complexité de cette méthode est donc de l'ordre de $0(k^2n^2)$.

Cette méthode d'initialisation est celle utilisée dans l'algorithme des k-médoïdes (PAM) introduit par Kaufman and Rousseeuw, nous allons donc tester cette technique.

Algorithme 9:

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).
- $D = [D_1, D_2, \dots, D_i, .., D_n]$, D_i : Distance de x_i à son plus proche centre.
- $S = [S_1, S_2, \dots, S_n]$ Variables contenant des sommes.

Début

Pour i=1 **jusqu'à** n **Faire** //Recherche du Premier centre
    $S_i = \sum_{j=1}^{n} d(x_i, x_j)$
Fin Pour
$idMin = Min(S)$
$C = C \cup \{x_{idMin}\}$
$X = X \setminus \{x_{idMin}\}$
Pour $l = 2$ **jusqu'à** k **Faire**
    $mise\_a\_jour(D)$ //(Par rapport aux centres définis)
    Pour i = 1 **jusqu'à** n **Faire**
        $S_i = \sum_{j=1}^{n} Max\ (D_i - d(x_i, x_j), 0)$
    Fin Pour
    $idMax = Max(S)$
    $C = C \cup \{x_{idMax}\}$
    $X = X \setminus \{x_{idMax}\}$
Fin Pour
Fin

La fonction $Min(S)$ renvoie l'index de l'élément minimum parmi les éléments de S, et $Max(S)$ renvoie l'index de l'élément maximum.
La fonction $mise\_a\_jour(D)$ recalcule l'ensemble D de distance entre tous les éléments et leurs centres.

Algorithme 9: Algorithme de l'initialisation PAM

3. 1.4) Méthode 4 (Médianes)

1) On calcule la variance de chaque dimension de l'ensemble des données.
2) On choisit la dimension (la colonne) avec la variance maximum.
3) On les trie dans n'importe quel ordre.
4) On découpe l'ensemble de donnée en k sous-ensemble.
5) On calcule la médiane de chaque sous-ensemble sur la dimension sélectionnée.
6) Et on prend comme centre les éléments (lignes) associés à chaque médiane.

Le calcul de la variance 1) a pour complexité 0(nd).
Le choix de la dimension maximum 2) est en 0(d).
Le tri de l'ensemble de données 3) est en 0(n* log(n)).
Le découpage en k partition 4) est en 0(k).
Le calcul des k médianes sur les partitions 5) est en 0(n).
La sélection des éléments associés à chaque médiane est en 0(1).

On obtient donc pour cette méthode d'initialisation une complexité de 0(nd + d + n*log(n) + k + n).
Cette complexité est de l'ordre de 0(n* (d+log(n))) si k est négligeable devant d. (Si d >> k >1)

Algorithme 10:

Données :

- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- $X[1]..X[i] \dots X[d]$ les éléments de X sur la $i^{ème}$ dimension.
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).
- $[E_1, E_2, \dots, E_k]$ Sous-ensemble de X.
- $V = [V_1, \dots, V_i, \dots, V_d]$ Variance sur la dimension i

Début

Pour i=1 **jusqu'à** d **Faire** //Recherche du Premier centre
    $V_i = variance\ (X[i])$ //Calcul de la variance pour chaque dimension
Fin Pour
$idMax = Max(V)$

$Tri\ (X)$ // On trie les données de X
$[E_1, E_2, \dots, E_k]$= sous_ensembles(X,k) // On crée k sous-ensembles de X.
Pour i = 1 **jusqu'à** k **Faire**
    $idmed = mediane(E_i[idMax])$ //calcul de l'id sur X de la médiane sur E_i
    $C = C \cup \{x_{idmed}\}$
Fin Pour
Fin

La fonction sous_ensembles(X,k) retourne k sous-ensembles de taille approximativement égale à partir de l'ensemble X.

Algorithme 10: Algorithme de l'initialisation des médianes

3. 1.5) Méthode 5 (park)

Cette méthode a été introduite par Park et Jun dans leur papier A simple and fast algorithm for k-medoids.

1) On calcule pour chaque x_j :

$$\sum_{i=1}^{n} \frac{d(x_i, x_j)}{\sum_{l=1}^{n} d(x_i, x_l)}$$

2) On trie les valeurs obtenues en ordre croissant.
3) Et on prend les éléments associés aux k premières valeurs comme centres.

Le calcul de cette double somme 1) pour chaque élément est en 0(n^3).
Le tri de l'ensemble des valeurs obtenues est en 0(n*log(n)).
La sélection des k première valeurs est en 0(k).

La complexité de cette méthode est donc de 0($n^3 + n\log(n) + k$)
Et elle est donc de l'ordre de 0(n^3).

Algorithme 11:

Données :

- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).
- $S = [S_1, \dots, S_j, \dots, S_n]$ variables des futures sommes

Début
Pour j = 1 **jusqu'à** n **Faire**
$S_j = \sum_{i=1}^{n} \frac{d(x_i, x_j)}{\sum_{l=1}^{n} d(x_i, x_l)}$
Fin Pour
idx = index_tri (S) //On trie S et on récupère les index (idx) dans un tableau

Pour i = 1 **jusqu'à** k **Faire**
$C = C \cup \{x_{idx_i}\}$
Fin Pour
Fin

La fonction index_tri(S) tri les éléments de S en ordre croissant et retourne la liste des index des valeurs triées.

Algorithme 11: Algorithme de l'initialisation park

3. 1.6) Méthode 6 (Points éloigné Minmax)

1) On calcule les distances entre tous les points.
2) On choisit les deux points les plus éloignés.
3) On calcule pour chaque point non-centroïdes x_i la distance à son plus proche centroïde. Et on prend le point x_i qui maximise cette distance. (Donc le point le plus loin des centres déjà présent)
4) On recommence en 3 jusqu'à avoir défini k centres.

Le calcul de la distance entre tous les points est en 0(n^2).
Puis le calcul du point qui maximise la distance à son plus proche centre est en 0(nk)
Donc la complexité de cette méthode est en 0(n^2 + nk) et elle est de l'ordre de 0(n^2)

Algorithme 12:

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).
- d_mat matrice des distances pré-calculées
- $d\_mat[i][j]$ distance entre x_i et x_j.
- $D = [D_1, D_2, \dots, D_i, .., D_n]$, D_i : Distance de x_i à son plus proche centre.

Début
$(x_1, x_2) = \max(d\_mat)$ // couple de points avec la distance maximum
$C = C \cup \{x_1, x_2\}$
$X = X \setminus \{x_1, x_2\}$

Pour i = 3 **jusqu'à** k **Faire**
  $mise\_a\_jour(D)$ //On met à jour D par rapport aux nouveaux centres
  Pour i = 1 **jusqu'à** n **Faire**
    $idMax = Max(D)$ //On récupère l'indice de l'élément le plus loin de son centre
    $C = C \cup \{x_{idMax}\}$
    $X = X \setminus \{x_{idMax}\}$
  Fin Pour
Fin Pour
Fin

La fonction $mise\_a\_jour(D)$ recalcule l'ensemble D de distance entre tous les éléments et leurs centres.
d_mat Représente la matrice des distances entre chacun des points.

$\max(d\_mat)$ Retourne les deux points les plus éloignés les uns des autres.

Algorithme 12: Algorithme de l'initialisation des points éloignés (minmax)

3. 1.7) Méthode 7 (Partitionnement avec densité avant)

1) On calcule un coefficient :
$$d1 = \frac{1}{n(n-1)} * \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|x_i - x_j\|$$
2) On compte pour chaque élément, le nombre d'éléments nb_i à une distance de maximum d1 (dans une sphère de rayon d1).
3) Et on choisit comme premier centre, comme l'élément qui a le nb_i le plus grand dans cette sphère de rayon d1.
4) Pour les autres centres, on choisit les éléments avec le nb_i le plus grand après les centres déjà choisis et qui sont au moins à une distance d1 de tous les centres déjà choisis.

Le calcul du coefficient 1) est en 0($\frac{(n-1)*(n-2)}{2}$).
Le calcul du nombre nb_i pour chacun des éléments est en 0(n²)
Le choix du premier centre est en 0(1).
Et le choix de tous les autres centres est au pire en 0(n).
Donc la complexité de cette méthode est de l'ordre de 0(n²)

Algorithme 13:

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).
- $nb = [nb_1, \dots nb_i, \dots, nb_n]$ tableau du nombre de voisins par élément

Début

$d1 = \frac{1}{n(n-1)} * \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|x_i - x_j\|$
//calcul de nb
Pour i = 1 **jusqu'à** n **Faire**
  $nb_i = 0$
  Pour j = 1 **jusqu'à** n **Faire**
    Si $d(x_i, x_j) < d1$ **Alors**
      $nb_i = nb_i + 1$
    Fin Si
  Fin Pour
Fin Pour
idx = index_tri (nb) //On trie nb et on récupère les index (idx) dans un tableau

j = 0
Pour i = 1 **jusqu'à** k+j **Faire**
  Si $au\_moins\_a\_d1(x_{idx_i})$ **Alors** // x_{idx_i} doit être au moins à d1 de tous les centres définis
    $C = C \cup \{x_{idx_i}\}$
  Sinon
    j=j+1
  Fin Si
Fin Pour
Fin

La fonction index_tri(nb) tri les éléments de nb en ordre croissant et retourne la liste des index des valeurs triées.

Algorithme 13: Algorithme de l'initialisation basé sur le partitionnement par densité

3. 1.8) Méthode 8 (Sample)

Cette méthode consiste à prendre un échantillon de l'ensemble de données (souvent 10% en général) et appliquer un algorithme de partitionnement sur cet échantillon puis de prendre les centres trouvés comme centres initiaux.

Si l'algorithme de partitionnement utilisé est l'algorithme des k-moyennes, alors la complexité de cette méthode est de 0(kn'i). Avec n' la taille du sous-ensemble et i le nombre d'itération de l'algorithme des k-moyennes.

Cette méthode d'initialisation semble assez rapide, et performante au vu des résultats présenté dans le document d'Anna D. Peterson, Arka P. Ghosh et Ranjan Maitra.
Elle pourra nous permettre d'utiliser la même fonction de distance pour initialiser les points et pour partitionner ensuite l'ensemble de données.
Elle mérite donc d'être testée pour notre analyse.

Algorithme 14:

Paramètre:
- taille : taille de l'échantillon (valeur courante 10%)

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).

Début

```
S = sous_ensemble (X,taille)  //On crée un sous-ensemble de X
C = Kmeans(S,k)               // On applique Kmeans dessus
```

Fin

La fonction sous_ensemble (X,taille) découpe l'ensemble X en un sous-ensemble avec «taille » éléments choisis de manière aléatoire parmi les éléments de X.

La fonction Kmeans représente l'algorithme des Kmeans initialisé de manière aléatoire.

Algorithme 14: Algorithme de l'initialisation basé sur un échantillon

3. 1.9) Méthode 9 (greedy)

1) On définit un nouveau centre c comme le point x_i (non-centroïde) qui minimise
$$\sum_{j=1}^{n} \min(D(x_j), d(x_i, x_j))$$
Avec x_j (non-centroïde)
2) Si aucun centre n'est définit D(x) = ∞
3) On recommence en 1 tant que l'on n'a pas défini k centres.

Le calcul de D(x) est de l'ordre de 0(k') avec k' le nombre de centroïdes définis.
Le calcul de la somme 1) est en 0(nk') pour un élément.
La sélection du minimum de cette somme pour chaque x_i est en 0(n²k')
La complexité de cette méthode est donc de $0\left(n^2 * \frac{(k-1)*k}{2}\right)$ et elle est de l'ordre de 0(n²k²)

D'après le document de référence [14], une implémentation direct de cet algorithme a pour complexité 0((n-k)²k²).
Pour obtenir une telle complexité, il faut supposer que l'on ne parcourt pas les k centres à chaque itération (donc n-k éléments).
Dans notre calcul, on considère que l'on parcourt bien toutes les données à chaque fois et on obtient par conséquent une complexité différente de celle présentée dans l'article [14].

Algorithme 15:

Données :
- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- k le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).
- $D = [D_1, D_2, \dots, D_i, .., D_n]$: Distance de x_i à son plus proche centre. (initialement à ∞).

Début

Pour t = 1 **jusqu'à** k **Faire**
    minz = ∞
    Pour i = 1 **jusqu'à** n **Faire**
        z' = $\sum_{j=1}^{n} \min\left(D(x_j), d(x_i, x_j)\right)$
        Si z' < minz **Alors**
            minz = z'
        Fin Si
    Fin Pour
    $C = C \cup \{x_{minz}\}$
    $X = X \setminus \{x_{minz}\}$
    $mise\_a\_jour(D)$ //On met à jour D par rapport aux nouveaux centres
Fin Pour
Fin

La fonction $mise\_a\_jour(D)$ recalcule l'ensemble D de distance entre tous les éléments et leurs centres.

Algorithme 15: Algorithme de l'initialisation greedy

3. 1.10) Complexité et choix des méthodes d'initialisations

Les complexités présentées ici sont des complexités au pire et en implémentation direct des algorithmes d'initialisation.

Algorithme d'initialisation	Complexité au pire	Nb paramètres
Méthode 1 (Aléatoire)	0(k)	0
Méthode 2 (Kmeans++)	0(kn)	0
Méthode 3 (Pam)	0(k^2n^2)	0
Méthode 4 (Médianes)	0($n(d + log(n))$)	0
Méthode 5 (Park)	0(n^3).	0
Méthode 6 (Minmax)	0(n^2)	0
Méthode 7 (Densité)	0(n^2)	0
Méthode 8 (Sample) en utilisant k-means.	0(kn'i) (avec n' environ 10% de n)	1 : Taille (en %) de n'
Méthode 9 (greedy)	0(k^2n^2)	0

Tableau 2: Tableau récapitulatif des complexités des méthodes d'initialisation

Suite à notre état de l'art sur la plupart des méthodes d'initialisation des algorithmes basées sur le partitionnement, il y a plusieurs méthodes qui seront testées et comparées :

- La méthode de Park car elle est utilisé dans la méthode des fast k-médoïdes, et nous allons tester cet algorithme donc nous allons également tester la méthode d'initialisation associée.
- La méthode de Pam car elle est utilisée dans PAM un des algorithmes que nous allons tester.
- La méthode Kmeans++ qui semble performante avec une complexité assez petite.
- La méthode Sample en utilisant l'algorithme des k-moyennes, qui semble données de bons résultats tout en ayant une complexité raisonnable.
- La méthode aléatoire qui est une référence parmi les différentes techniques d'initialisation.

3.2) Choisir la valeur de k.

Le choix de la valeur de k est le plus souvent laissé à l'utilisateur. Pour faire son choix l'utilisateur se base en général sur :

- une connaissance à priori sur les données (le problème est supposé contenir k classes)
- un nombre de partition qui l'arrange pour l'utilisation qu'il fera des partitions.
- le test de plusieurs valeurs de k et une sélection de celle qui optimise un certain critère (comme le critère BIC par exemple)

Des méthodes ont été introduites pour régler la valeur de k de manière automatique.
Parmi ces méthodes on retrouve G-means[18], X-means[19], S-means[20].
La méthode X-means est détaillée dans l'annexe 8.

Il existe également une méthode de réglage « automatique » de k par un algorithme des k-moyennes entropique :
"An Unsupervised Clustering Method using the Entropy Minimization" [28] de Gintautas Palubinskas et "Texture Analysis through a Markovian Modelling and Fuzzy Classification: Application to Urban Area Extraction from Satellite Images" [29] A. LORETTE.

Dans notre cas on réalise une étude pour ensuite se placer dans un contexte industriel. Dans ce 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 courriers. Dans ce contexte, il est préférable de laisser le choix de k à l'utilisateur.

3.3) Choisir une fonction de distance / une métrique.

3.3.1) Distances

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).

Cependant selon la fonction de distance choisie, on trouvera des clusters différents.

Il existe beaucoup de fonctions de distance, puisse qu'une fonction de distance doit simplement vérifier les propriétés suivantes:

- Elle doit prendre 2 arguments en paramètre à partir du même ensemble et retourner une valeur dans R^+.

- Elle doit vérifier une propriété de symétrie :
$$\forall x, \forall y \in X\ , d(x,y) = d(y\ , x)$$
- Elle doit vérifier une propriété de séparation :
$$\forall x, \forall y \in X\ , d(x,y) = 0 \Leftrightarrow x = y$$
- Elle doit vérifier l'inégalité triangulaire :
$$\forall x, \forall y, \forall z \in X\ , d(x,z) \leq d(x,y) = d(y,z)$$

Dans notre analyse nous allons considérer les fonctions de distance suivantes :

- La distance de Hamming

C'est une fonction qui à la base est définie pour des segments binaires, mais elle est aussi applicable sur des données catégorielles.

Elle se définie de manière très simple.
Soit x et y 2 données de X. soit x_c la valeur de la caractéristique c de x

$$d(x,y) = \sum_{c}^{C} \delta(x_c, y_c) \quad \begin{cases} \delta(x_c, y_c) = 1 \; si \; x_c = y_c \\ \delta(x_c, y_c) = 0 \; si \; x_c \neq y_c \end{cases}$$

- Distance de Levenstein ou Damerau-Levenstein

Elles définissent entre deux mots un nombre d'opérations pour passer d'un mot à l'autre.
Elles peuvent être utilisées de la même manière que la distance de Hamming sur chaque caractéristique à la place de retourner une valeur binaire, on renvois le nombre d'opérations nécessaires pour passer d'un mot à l'autre.

- La norme L2 ou distance euclidienne.

Elle est définie par :

$$d(x,y) = \sqrt{\sum_{c}^{C} |x_c - y_c|^2}$$

- La norme Lp (aussi appelé distance de Minkowski)

$$d(x,y) = \sqrt[p]{\sum_{c}^{C} |x_c - y_c|^p}$$

Quand la fonction euclidienne et celle de Minkowski sont utilisées avec un algorithme de partitionnement, cela crée des clusters en forme d'ellipses, ou d'hyper-ellipses en dimension supérieur à 2.

- Norme L1 (distance de Manhattan)

$$d(x,y) = \sum_{c}^{C} |x_c - y_c|$$

Cette distance permet de créer des clusters en forme de losanges lors de son utilisation dans un algorithme de partitionnement.

- Norme infinie (distance de Tchebychev)

$$d(x,y) = \sup_{1<c<C} (|x_c - y_c|)$$

Cette distance permet de créer des clusters en forme de carrés lors de son utilisation dans un algorithme de partitionnement.

- Distance de Mahalanobis

Elle est basée sur la matrice de covariance Σ de l'ensemble de données.

$$d(x,y) = \sqrt{(x-y)^T \, \Sigma^{-1} (x-y)}$$

3.3.2) Discussion

Certaines distances sont faites pour les chaînes de caractères ou les données catégorielles comme la distance de Levenstein, Damerau-Levenstein, ou celle de Hamming.

Ce type de distance ne nous intéresse pas car nous souhaitons travailler sur des données numériques (ou qui auront été mis sous un format numérique).

La distance de Mahalanobis s'adapte à la répartition de données par rapport à chaque variable. Elle correspond à une distance normalisée par rapport à nos données. Elle est donc particulièrement intéressante lorsque l'on travaille sur des données où les espaces de définition des variables sont très différents d'une variable à une autre.

D'une part nos données seront discrétisées, et donc l'espace de définition de leurs variables seront à peu près similaire d'une variable à une autre, ce qui rend cette fonction de distance moins intéressante pour nous.

Et d'autre part nous allons travailler sur des ensembles de données très grand, et calculer la matrice de covariance avec de tels ensembles est couteux en temps et en mémoire. Par conséquent la fonction de Mahalanobis n'est pas adaptée pour notre problème.

On sait que nos données seront la plus part du temps prétraitées avec MODL [25], avant de les utiliser pour créer une représentation supervisée ces données. Et MODL, fait une discrétisation par grilles des données qu'il traite. Par conséquent nos données seront localement contenues à l'intérieur de carrée ou de rectangle.

C'est pourquoi il semble peu judicieux d'utiliser des fonctions de distance formant des clusters en forme d'ellipses, et qu'il serait plus intéressant de privilégier des fonctions de distance permettant de réaliser des clusters carrées ou rectangulaires.

Les normes L2 et Lp nous donneraient des clusters en forme d'ellipses ou d'hyper ellipses, c'est pour cette raison que nous ne les utiliserons pas.

La distance de Tchebychev peut nous permettre aussi d'obtenir une forme de cluster satisfaisante, néanmoins avec cette distance on ne prend en considération que l'écart entre les éléments sur une seule variable (celle pour laquelle les deux éléments sont les plus éloignés l'un de l'autre). Il semble dommage de ne pas utiliser l'écart par rapport à toutes les variables d'une donnée pour déterminer sa distance à une autre, c'est pourquoi nous n'utiliserons pas cette distance.

La norme L1 permet de créer des clusters en forme de losanges lors de son utilisation dans un algorithme de partitionnement. Cette forme de cluster semble parfaitement adaptée à nos données. De plus la norme L1 permet de considérer chaque variable avec la même importance.

Il semble dès lors judicieux dans notre cas de prendre la **norme L1** comme fonction de distance.

4) Représentation supervisée des données

4.1) Notre définition d'une représentation supervisée des données

Une représentation supervisée des données est une manière de représenter des données initialement brutes grâce à une technique de groupage et de discrétisation qui utilise une information de classe sur les données.

Ces données brutes peuvent être continues, catégorielles, ou bien même les deux.

Dans le cas où les variables de ces données sont continues, elles seront discrétisées, et dans le cas où elles sont catégorielles, elles seront groupées. Ce prétraitement permet de représenter les données catégorielles et continues de la même manière.

Exemple:
Avec ce prétraitement une variable continue « âge » pourrait se représenter ainsi:
- (a) Les moins de 18 ans.
- (b) Les 18-25 ans
- (c) Les 25-45 ans
- (d) Les 45- 60 ans
- (e) Les plus de 60 ans.

Et avec ce prétraitement une variable catégorielle « Civilité » pourrait se représenter ainsi:
- (1) M.
- (2) Mme et Mlle

Une valeur de variable est maintenant représentée par la valeur du groupe ou de l'intervalle auquel elle appartient.

A partir de cette représentation des données et de leurs classes sur l'ensemble d'évaluation, on peut calculer pour chaque valeur x de groupe ou d'intervalle d'une variable X et pour chaque classe C la probabilité P(X=x/C).

Pour chaque variable explicative X d'un élément, on calcule la probabilité P(X=x/C) à partir d'un ensemble d'évaluation.
Cette probabilité correspond au fait qu'un élément est la valeur x pour la variable explicative X sachant qu'il est de la classe C.

Cette probabilité peut être transformée en quantité d'information telle que A=Log(P(X=x/C).

Voici comment on pourrait représenter 3 individus I_1, I_2, I_3 avec chacun une variable « âge » X_1 avec pour valeur $x_1 \in \{a, b, c, d, e\}$ et une variable « Civilité » X_2 avec pour valeur $x_2 \in \{1,2\}$.

Soit I_1, I_2, I_3 tels que $I_1 = (e, 1)$, $I_2 = (b, 2)$, $I_3 = (d, 2)$.

On obtiendrait la représentation suivante:

	P(X_1/C_1)	P(X_2/C_1)	P(X_1/C_2)	P(X_2/C_2)
I_1	Log(P($X_1 = e/C_1$))	Log(P($X_2 = 1/C_1$))	Log(P($X_1 = e/C_2$))	Log(P($X_2 = 1/C_2$))
I_2	Log(P($X_1 = b/C_1$))	Log(P($X_2 = 2/C_1$))	Log(P($X_1 = b/C_2$))	Log(P($X_2 = 2/C_2$))
I_3	Log(P($X_1 = d/C_1$))	Log(P($X_2 = 2/C_1$))	Log(P($X_1 = d/C_2$))	Log(P($X_2 = 2/C_2$))

Tableau 3: Tableau d'un exemple de représentation supervisée de 3 éléments
De cette manière, on garde une connaissance de l'apprentissage effectué sur l'ensemble d'évaluation lors de la représentation de nos données.

Dans les sections suivantes, nous allons détailler d'autres types de représentation que nous testerons, et la manière de les obtenir.

4.2) La représentation brute

Nous appelons dans ce document représentation brute des données, la représentation native des données.

L'application d'un algorithme de clustering basé sur le partitionnement à ce type de représentation implique une fonction de distance adaptée.

Une fonction de distance adaptée à une représentation est par exemple la distance euclidienne pour des données numériques, et la distance de Hamming pour des données catégorielles.

Dans le cadre de nos tests nous avons choisi la norme L1 car elle est adaptée aux données numériques des représentations supervisées.

Nous souhaitons conserver la même fonction de distance tout au long des tests. Ceci explique que lorsque nous utiliserons une représentation brute de nos données, nous ne testerons que les ensembles de données qui ne contiennent que des données numériques.

4.3) La représentation issue des connaissances d'une classification par Kxen(K2R)

Kxen est un outil qui peut permettre de réaliser une représentation. Il est actuellement utilisé par Orange dans un contexte industriel pour réaliser des groupes pour les campagnes markéting.

Nous l'étudions pour comparer la représentation supervisée actuellement utilisée par Orange avec celle que l'on propose (Voir partie II section 4.4).

Kxen est composé de plusieurs modules. L'un d'entre eux, K2R permet de faire de la classification et de la régression. Un autre de ses modules est K2S, il permet de faire de la segmentation.

Il est possible d'utiliser le module K2R pour réaliser une représentation supervisée de données.

Pour cela il faut appliquer Kxen sur un jeu de données en lui spécifiant une valeur cible.
Kxen va découper le jeu de données en 3 ensembles par défaut (estimation, validation et test). L'ensemble d'estimation sera utilisé pour réaliser l'apprentissage et créer différents modèles, l'ensemble de validation sera utilisé pour sélectionner le meilleur modèle parmi ceux créés lors de l'apprentissage.
Et enfin l'ensemble de test sera utilisé pour calculer des indicateurs de généralisation et de qualité de la solution trouvée.

Il semblerait que la première étape du processus d'apprentissage de Kxen soit la discrétisation des données à partir de la variable cible.

Et que la deuxième étape soit le calcule des P(X=x/C) à partir de la discrétisation effectuée.
Avec « X » une variable.
Avec « C » la valeur d'une classe.
Et « x » la valeur d'une variable (discrétisée)

Kxen fonctionne un peu comme une boite noire donc il est difficile d'être certain de son fonctionnement, néanmoins il semble qu'il fonctionne comme cela.

Il est ensuite possible d'exprimer chaque élément à partir de ces probabilités comme l'exemple du tableau 3 (Partie II section 4.1) en réalisant une projection.

L'utilisation standard de Kxen suppose de le laisser réaliser ces prétraitements en interne sans visibilité dessus et sans faire de projection.

4.4) La représentation issue des connaissances d'une classification par khiops

Il est possible d'utiliser Khiops et Kawab pour réaliser une représentation supervisée.

Khiops est un outil de classification créé par l'équipe. Il est basé sur MODL. MODL permet de réaliser une discrétisation ou un groupage. Elle permet également de moyenner un prédicteur Bayesien Naïf Sélectif.

Les articles suivants " MODL: Une méthode quasi-optimale de discrétisation supervisée " [25], " MODL: Une méthode quasi-optimale de groupage des valeurs d'un attribut symbolique " [26], et " Moyennage du prédicteur Bayesien Naïf Sélectif, évaluation sur un challenge international " [27] de Marc Boullé permettent de comprendre son fonctionnement.

Khiops prend une Base de données en entrée. Il discrétise et groupe les données avec la méthode MODL. Il produit en sortie un classifieur Bayesien Naïf Sélectif.

Kawab est un outil qui prend en entrée un classifieur Khiops et une base de données.
Il produit en sortie les Log(P(X/C)) de chacun des éléments de la base de données.

Ces contributions de la forme Log(P(X=x/C)) sont celles de l'exemple du tableau 3 (Partie II section 4.1)

On pondère ensuite ces contributions en fonction de l'information portée par les variables.

Ces contributions prennent alors la forme de w_X x Log(P(X=x/C)).

Et c'est cette représentation (sous forme de quantité d'information) que nous proposons d'utiliser.

4.5) L'algorithme supervisé utilisé en comparaison

Il nous semble intéressant de comparer les algorithmes de clustering choisis dans la partie II section 2 avec un algorithme de classification, le classifieur Bayesien Naïf Sélectif.

Ce classifieur sera un objectif intéressant de comparaison, car en général les techniques de clustering ne permettent pas d'obtenir de meilleurs résultats en classification que des algorithmes de classification.

Le Baseline (classifieur majorité) sera un autre objectif de comparaison intéressant. C'est l'un des plus mauvais classifieur, et l'on devrait

Il pourrait être intéressant en fin de stage s'il reste du temps de comparer les algorithmes de clustering choisis avec un algorithme de clustering supervisé. Dans ce cas nous choisirions celui présenté par Al-Harbi et Rayward-smith dans son article *Adapting k-means for supervised clustering.* Cet algorithme est décrit dans l'annexe 44.

5) Analyse de l'influence de la représentation sur la qualité du clustering

5.1) Méthodologie d'évaluation du clustering.

Quelque soit l'algorithme qui va être testé, quelque soit la méthode d'initialisation, quelque soit les critères de qualité nous allons utiliser la même méthodologie pour réaliser les tests.

- **Découpage des données**

Nous découperons nos ensembles de données pour faire de la 10-foldcross validation dessus (voir annexe 2).

Les tests sont ensuite fait sur les 10 découpes afin d'obtenir à chaque fois un résultat moyen muni de son écart-type.

- **Application d'un algorithme de clustering et affectation d'une classe à chaque élément**

On applique sur les 10 ensembles d'entrainement l'algorithme de clustering choisi.
L'algorithme de clustering produit pour chacun des ensembles d'entrainement k groupes d'éléments (clusters) et produit k centres de groupes.

Ces éléments possèdent une classe. On peut ensuite observer la classe des différents éléments contenus dans un groupe et déterminer la classe dominante d'un groupe.

Pour chaque élément des ensembles de test, on pourra calculer quel est le centre de groupe c le plus proche et associer à cet élément la classe dominante du groupe représenté par le centre c.

On pourra définir un score par la probabilité d'être d'une classe donnée C sachant qu'on appartient à un groupe donné CL, c'est la probabilité P(C/CL).

La figure 23 illustre la méthode d'évaluation sur les ensembles d'entrainement et la figure 24 illustre la méthode d'évaluation sur les ensembles de tests.

On pose:

- CL i : Le $i^{ème}$ cluster (1<= i <= k)
- Algorithme de partitionnement : (k-moyennes+recherche de points de l'ensemble comme centre, PAM, fast k-médoïd, Kxen un k-moyennes supervisé)
- J le nombre de classe de notre ensemble de données.

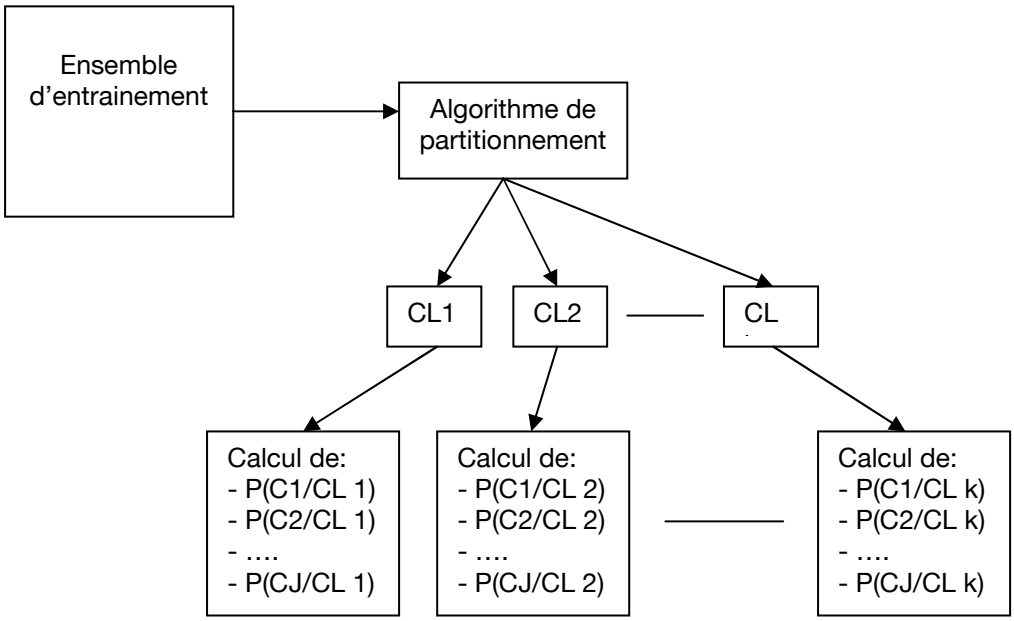


Figure 23: Organigramme de la méthode d'évaluation (apprentissage)

Pour chaque élément de l'ensemble de test, on va l'associer à un cluster en fonction de sa distance par rapport à tous les clusters (on l'affecte au cluster qui est le plus proche de lui). Puis on va associer à chaque élément une classe en fonction de la classe majoritaire au sein du cluster c'est-à-dire la classe i qui maximise la probabilité d'appartenir à cette classe sachant qu'il appartient à un cluster.

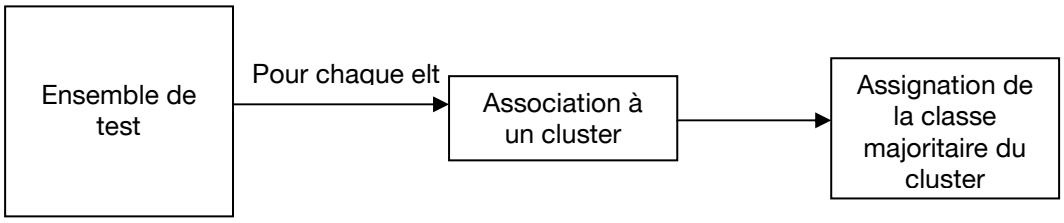


Figure 24: Organigramme de la méthode d'évaluation (test)

Ainsi les éléments de l'ensemble de test et ceux de l'ensemble d'apprentissage peuvent être mis dans un cluster. Muni de cette information on détaille ci-après les critères d'évaluation de la qualité du clustering obtenu.

5.2) Critères d'évaluation de la qualité du clustering

Il existe de nombreux critères pour évaluer la qualité du clustering obtenu. Nous nous proposons ici de détailler les critères que nous allons utiliser.

5. 2.1) Le SSE

Le SSE (Sum Squared Error) correspond à la somme des distances au carrée entre un centre et chacun des points de son cluster.

Soit :

- $X = [x_1, x_2, \dots, x_n]$ un ensemble de n données
- K le nombre de partitions souhaitées
- $C = \emptyset$ L'ensemble des k futurs centres ($[c_1, c_2, \dots, c_n]$).

$$\text{SSE (X)} = \sum_{j=1}^{K} \sum_{i=1}^{n} \|x_i - C_j\|^2$$

Ce critère peut donner des valeurs comprises entre 0 et$+\infty$. Un clustering A sera considéré meilleur par ce critère qu'un clustering B si le clustering A donne une valeur plus faible par ce critère que le clustering B.

5. 2.2) Le critère de Sylvain Ferrandiz [21]

$$c = log(N) + log\binom{N+K-1}{K} + \sum_{k=1}^{K} log\binom{N_k + J - 1}{J-1} + \sum_{k=1}^{K} log\left(\frac{N_k!}{N_{k1}! \dots N_{kJ}!}\right)$$

Dans ce critère :
N est le nombre d'éléments de l'ensemble testé (pour nous l'ensemble de test).
K est le nombre de clusters
J est le nombre de classe au sein de l'ensemble.
N_k est le nombre d'élément appartenant au cluster k
N_{kj} est le nombre d'élément de classe j appartenant au cluster k

Un problème se pose lorsque l'on implémente ce critère en effet :

$$\binom{N+K-1}{K} = \frac{(N+K-1)!}{K! * (N-1)!}$$

Or on a souvent un nombre de données N très grand et on ne peut pas calculer le factoriel de grand nombre. (170 ! = 7.25741562 × 10^{306})

Mais en fait il y a une astuce puisse que l'on cherche le log :

$$log\, n! = \sum_{i=1}^{n} \log i$$

On obtient donc :

$$log\binom{N+K-1}{K} = \sum_{i=1}^{N+K-1} \log i - \sum_{i=1}^{K} \log i - \sum_{i=1}^{N-1} \log i$$

En sachant cela, l'implémentation de ce critère devient beaucoup plus évidente.

Ce critère peut donner des valeurs comprises entre 0 et+∞. Un clustering A sera considéré meilleur par ce critère qu'un clustering B si le clustering A donne une valeur plus faible par ce critère que le clustering B.

5. 2.3) Le taux de bonne classification (Accuracy)

Pour trouver le taux de bonne classification des données, on calcule le nombre d'éléments correctement classés sur le nombre total d'éléments.

On l'exprime généralement en pourcentage.

Ce critère peut donner des valeurs comprises entre 0 et 100. Si le critère vaut 100, cela signifie que tous les éléments ont été correctement classés.

5. 2.4) L'AUC sous la courbe Roc

La définition d'une courbe ROC (Receiver Operating Characteristics) et le calcul de l'AUC correspondant est très bien décrite dans le document de Tom Fawcett [17].

Une explication détaillée de la définition d'une courbe ROC, et la manière de la calculer, et la manière de calculer l'AUC d'une courbe ROC sera disponible en Annexe 1.

Ce critère peut donner des valeurs comprises entre 0.5 et 1. Une très bonne classification donnera un AUC proche de 1 tandis qu'une classification complètement aléatoire donnera un AUC proche de 0.5.

5.3) Les tests mis en place

Au cours de notre étude différents algorithmes de clustering (Partie II section 2), nous avons choisi de tester les algorithmes suivant :

- K-moyennes
- PAM (Partition Around Médoïd)
- Fast K-medoïds

Nous souhaitons aussi comparer ces algorithmes à deux autres algorithmes.
Le premier est K2S, l'algorithme de Kxen utilisé par l'équipe de la cellule score.
Le deuxième est le naïf bayes sélectif, il nous permettra d'étudier la performance de l'algorithme en représentation supervisée par rapport à une méthode de classification.

Lors de notre étude des méthodes d'initialisation des algorithmes de clustering basés sur le partitionnement (Partie II section 3), nous avons choisi de tester les méthodes d'initialisations suivantes :

- la méthode aléatoire
- la méthode des kmeans++
- la méthode Sample
- la méthode PAM
- la méthode Fast K médoïdes.

Après notre présentation des différentes méthodes pour réaliser une représentation supervisée, nous avons décidé d'étudier 2 méthodes de représentation des données :

- La représentation brute.
- La représentation supervisée basée sur Khiops et Kawab.

Nous allons tester le résultat du clustering obtenu par 4 critères de qualité vu dans la partie II section 5.2 :

- Le taux de bonne classification (Accuracy)
- L'AUC
- Le critère de sylvain ferrandiz
- Le SSE

Le but de ces tests est d'observer en quoi une représentation supervisée des données peut elle permettre d'améliorer la qualité du clustering pour des valeurs de k différentes (De 2 à $\sqrt{n}$). Les tests seront effectués sur les 42 bases décrites dans l'annexe 3 et pour une partie des tests sur les 9 bases décrites dans l'Annexe 46.

-L'initialisation

Nous appliquerons l'algorithme des k-moyennes sur les données brutes, pour choisir la méthode d'initialisation à utiliser pour cet algorithme.

Les méthodes d'initialisation suivantes vont être testées : Aléatoire, Kmeans++, sample, Pam, fast-kmédoïdes. (Voir partie II section 3 pour le détail des méthodes). Une fois la méthode d'initialisation choisie on utilisera toujours cette méthode dans l'algorithme des k-moyennes.

-Représentation brute.

La série de tests suivante à pour but de tester la qualité du clustering de chacune des méthodes retenues. A savoir PAM, fast-kmédoïdes, et l'algorithme des k-moyennes.

Les méthodes ne seront testées avec une représentation brute que sur des bases contenant des valeurs numériques.

-Représentations supervisées.

Nous testerons ensuite l'influence des représentations supervisées en comparaison à la représentation brute.

On commencera par tester la représentation basée sur K2S décrite dans la partie II section 4.3. Elle sera testée sur toutes les bases.

Puis nous évaluerons la représentation supervisée basée sur khiops et kawab décrite dans la partie II section 4.4. On testera les 3 algorithmes retenus avec cette représentation, à savoir PAM, les fast-kmédoïdes, et l'algorithme des k-moyennes avec recherche des centres sur toutes les bases de données.

- Méthodes de classification

Nous testerons la qualité d'une méthode de classification en comparaison, aux algorithmes de clustering testés et parfois utilisés comme technique de classification.

Pour réaliser cette série de tests nous avons choisi d'utiliser l'algorithme du naïf bayes sélectif, et celui du classifieur majorité.

Ces algorithmes nous serviront de référence pour tester nos méthodes de clustering.

Le tableau 4 présente de manière spécifique les différents tests qui seront réalisés.

Test à réaliser	Initialisation	Nombre de bases concernées	Valeurs de k
Algorithme K-moyenne puis recherche des centres en non-supervisé.	Aléatoire	26 (bases numériques)	De 2 à $\sqrt{n}$
Algorithme K-moyenne puis recherche des centres en non-supervisé.	Kmeans++	26 (bases numériques)	De 2 à $\sqrt{n}$
Algorithme K-moyenne puis recherche des centres en non-supervisé.	Sample	26 (bases numériques)	De 2 à $\sqrt{n}$
Algorithme K-moyenne puis recherche des centres en non-supervisé.	Pam	26 (bases numériques)	De 2 à $\sqrt{n}$
Algorithme K-moyenne puis recherche des centres en non-supervisé.	Fast k-médoïdes	26 (bases numériques)	De 2 à $\sqrt{n}$
Algorithme PAM non-supervisée sur des données numériques.	Standard (PAM)	4 (bases numériques)	De 2 à $\sqrt{n}$
Algorithme fast-k-médoïdes non-supervisé sur des données numériques.	Standard (fast kmédoïdes)	4 (bases numériques)	De 2 à $\sqrt{n}$
Algorithme K-moyenne puis recherche des centres en non-supervisé sur des données numériques.	Choisie	4 (bases numériques)	De 2 à $\sqrt{n}$
Kxen(K2S) en technique non-supervisée.	Standard	9	De 2 à $\sqrt{n}$
Algorithme PAM basé la représentation khiops et kawab.	Standard	9	De 2 à $\sqrt{n}$
Algorithme fast-kmédoïdes basé la représentation khiops et kawab.	Standard	9	De 2 à $\sqrt{n}$
Algorithme K-moyenne puis recherche des centres basé la représentation khiops et kawab.	Choisie	9	De 2 à $\sqrt{n}$
Classifieur Naïf Bayes sélectif	Standard	9	De 2 à $\sqrt{n}$

Tableau 4: Tableau des tests à réaliser.

5.4) Résultats expérimentaux

5.4.1) Les techniques d'initialisation

Voici les résultats que nous avons obtenus sur avec les techniques d'initialisations suivantes :

- Aléatoire
- Sample
- Kmeans++

Nous avons essayé de tester les techniques d'initialisation de PAM et de celle des fast k-médoïdes. Malgré la puissance de calcul disponible il s'est avéré que ces méthodes d'initialisation ne peuvent être utilisées que pour des petites bases de données (faible nombre d'instances et faible nombre de variables explicatives). Il faut aussi noter que les bases de l'UCI utilisées dans cette section sont des « petites » bases de données comparé aux données d'Orange qui sont par nature en grande volumétrie (plusieurs dizaines de milliers d'instances x plusieurs centaines de variables explicatives). Par conséquent lors des tests les calculs pour ces méthodes ont été interrompus ; les résultats ne sont donc pas présentés.

Les deux tableaux suivants présentent respectivement sur train et test, les résultats du nombre de fois où une méthode gagne sur 24 bases pour chacun des critères. (En cas d'égalité de deux méthodes pour une base donnée et un critère donné, les deux méthodes se voient attribuées le même rang)

Ces deux tableaux ont été réalisés à partir du tableau en Annexe 42 présentant les valeurs moyennes trouvées pour les critères de qualité avec des valeurs de k différentes (k allant de 2 à $\sqrt{n}$).

Train	Sylv	Sylv écart-type	ACC	ACC écart-type	SSE	SSE écart-type	AUC	AUC écart-type
Aléatoire	11	7	13	10	9	11	14	8
Sample	18	8	17	10	11	8	16	9
kmeans++	5	10	7	5	5	6	7	8

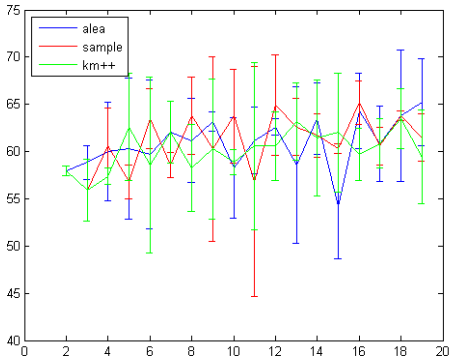
Tableau 5 : Nombre de fois où la méthode gagne (sur 24 bases) sur Train

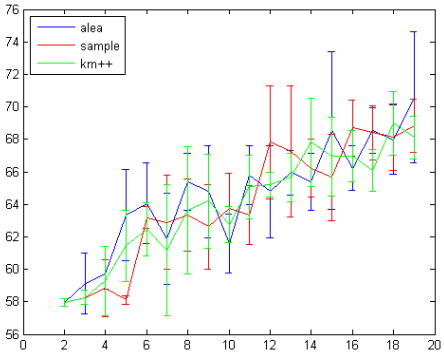
Test	Sylv	Sylv écart-type	ACC	ACC écart-type	SSE	SSE écart-type	AUC	AUC écart-type
Aléatoire	13	10	10	12	12	7	13	8
Sample	19	8	15	9	8	12	13	11
kmeans++	9	6	6	4	4	7	6	5

Tableau 6 : Nombre de fois où la méthode gagne (sur 24 bases) sur Test

Pour réaliser les tableaux présentant les résultats du nombre de fois où une méthode gagne sur 24 bases pour chacun des critères, on calcule dans un premier temps l'aire sous chacune des courbes (comme celles-ci-dessous) à partir de leurs valeurs. L'aire sous une courbe correspond pour nous à la moyenne des valeurs de cette par rapport à k (en abscisse). Dans un deuxième temps, on calcule le nombre de fois où une méthode a été la première par rapport aux autres.

Taux de bonne classification (Accuracy) (Test/Train) sur BUPA





Toutes les courbes pour les méthodes d'initialisation sont disponibles dans les annexes 11 à 35.

Les valeurs moyennes des critères de qualité sont ensuite présentées dans les Annexes (L'Annexe 42 pour les méthodes d'initialisation avec k allant de 2 à $\sqrt{n}$).

On refait les mêmes tableaux pour k allant de 2 à 10 ce qui correspondra vraisemblablement plus à l'usage qui sera fait par l'usage industriel des k-moyennes à Orange.

Les deux tableaux suivants présentent respectivement sur train et test, les résultats du nombre de fois où une méthode gagne sur 9 bases pour chacun des critères. (En cas d'égalité de deux méthodes pour une base données et un critère donné, les deux méthodes se voient attribuées le même rang)

Ces deux tableaux ont été réalisés à partir du tableau en Annexe 43 présentant les valeurs moyennes trouvées pour les critères de qualité avec des valeurs de k différentes (k allant de 2 à 10).

Train	Sylv	Sylv écart-type	ACC	ACC écart-type	SSE	SSE écart-type	AUC	AUC écart-type
Aléatoire	16	9	13	15	10	8	15	11
Sample	17	9	11	8	7	9	10	10
kmeans++	10	6	8	2	7	7	5	3

Tableau 7 : Nombre de fois où la méthode gagne (sur 24 bases) sur Train avec k allant de 2 à 10

Test	Sylv	Sylv écart-type	ACC	ACC écart-type	SSE	SSE écart-type	AUC	AUC écart-type
Aléatoire	15	10	11	9	12	8	12	9
Sample	14	8	13	10	8	8	11	8
kmeans++	8	6	4	6	3	8	4	9

Tableau 8 : Nombre de fois où la méthode gagne (sur 24 bases) sur Test avec k allant de 2 à 10

5.4.2) Les méthodes de clustering sur une représentation brute en comparaison à la représentation supervisée.

Les 4 tableaux suivants présentent les résultats moyens (par rapport à k) des méthodes Pam et Kmeans en supervisée et non-supervisée obtenus.
Lors de nos tests l'algorithme PAM n'a pas fonctionné sur Shuttle et Letter donc nous ne présenterons pas les résultats de cette méthode sur ces deux ensembles de données.

L'algorithme des Fast k médoïdes n'a fonctionné que pour Iris en non-supervisée et cette méthode tourne encore pour la représentation supervisée donc nous ne présenterons pas les résultats de la méthode des fast-kmédoïdes.

Iris	Sylv	ACC	SSE	AUC
Pam (Train)	8,34E+01	8,92E+01	**8,31E+01**	9,63E-01
Pam supervisée (Train)	**7,61E+01**	**9,27E+01**	9,20E+01	**9,72E-01**
Kmeans (Train)	8,63E+01	8,88E+01	**1,86E+02**	9,53E-01
Kmeans supervisée (Train)	**8,09E+01**	**9,10E+01**	3,64E+03	**9,60E-01**
Pam (Test)	2,54E+01	8,96E+01	**8,91E+00**	**9,59E-01**
Pam supervisée (Test)	**2,51E+01**	**9,07E+01**	1,13E+01	9,51E-01
Kmeans (Test)	2,55E+01	8,78E+01	**2,69E+00**	9,46E-01
Kmeans supervisée (Test)	**2,47E+01**	**9,31E+01**	7,33E+01	**9,66E-01**

Tableau 9 : Résultats moyens par rapport à k sur une représentation brute et sur une représentation supervisée de la base Iris

Phonème	Sylv	ACC	SSE	AUC
Pam (Train)	**1,00E+03**	**8,31E+01**	**2,62E+04**	**9,36E-01**
Pam supervisée (Train)	1,09E+03	7,82E+01	7,17E+04	9,35E-01
Kmeans (Train)	1,24E+03	7,73E+01	1,14E+06	9,15E-01
Kmeans supervisée (Train)	**1,12E+03**	**7,84E+01**	**6,75E+04**	**9,25E-01**
Pam (Test)	2,11E+02	**8,15E+01**	**4,15E+02**	9,26E-01
Pam supervisée (Test)	**2,07E+02**	7,86E+01	8,92E+02	**9,35E-01**
Kmeans (Test)	2,26E+02	7,61E+01	1,37E+04	9,10E-01
Kmeans supervisée (Test)	**2,16E+02**	**7,69E+01**	**8,60E+02**	**9,19E-01**

Tableau 10 : Résultats moyens par rapport à k sur une représentation brute et sur une représentation supervisée de la base Phonème

Shuttle	Sylv	ACC	SSE	AUC
Kmeans (Train)	1,30E+04	8,84E+01	5,83E+09	9,02E-01
Kmeans supervisée (Train)	**1,11E+04**	**9,26E+01**	**3,32E+08**	**9,41E-01**
Kmeans (Test)	2,42E+03	8,85E+01	7,49E+07	9,02E-01
Kmeans supervisée (Test)	**1,86E+03**	**9,24E+01**	**3,67E+06**	**9,29E-01**

Tableau 11 : Résultats moyens par rapport à k sur une représentation brute et sur une représentation supervisée de la base Shuttle

Letter	Sylv	ACC	SSE	AUC
Kmeans (Train)	4,99E+04	1,67E+01	1,74E+07	7,30E-01
Kmeans supervisée (Train)	**4,44E+04**	**2,28E+01**	**9,17E+06**	**7,98E-01**
Kmeans (Test)	5,96E+03	1,56E+01	2,45E+05	7,11E-01
Kmeans supervisée (Test)	**5,43E+03**	**2,22E+01**	**1,07E+05**	**7,87E-01**

Tableau 12: Résultats moyens par rapport à k sur une représentation brute et sur une représentation supervisée de la base Letter

5.4.3) Les méthodes de clustering sur une représentation supervisée.

Voici les résultats que nous avons obtenus sur avec les techniques de clustering suivantes :

- Pam (Partition Around Medoid)
- K-moyennes
- Kxen
- Fast k médoïde
- SNB (Algorithme de classification Bayesien Naïf Sélectif)

Nous avons essayé l'algorithme des Fast k médoïdes. Il s'est avéré que cette méthode est adaptée qu'aux « petites » bases. Dans le cadre de nos tests, nous n'avons pas été en mesure de collecter tous les résultats pour cette méthode. Les résultats avec la méthode des Fast k médoïdes ne seront donc pas présentés.
De plus lors de nos tests, on a pu remarquer que la convergence de cette implémentation des k-médoïdes mérite d'être prouvée, ou l'algorithme mérite d'être corrigé (ce que nous avons fait).

La méthode Kxen ne nous permet pas d'obtenir les centres des clusters (calculés en internes), donc nous ne seront pas en mesure d'afficher le critère SSE pour cette méthode.

De plus il faut noter que PAM a échoué sur les 2 bases Shuttle et Letter et que Kxen a échoué sur la base Phonème.

Les deux tableaux suivants présentent respectivement sur train et test, les résultats du nombre de fois où une méthode gagne sur 9 bases pour chacun des critères de qualité. (En cas d'égalité de deux méthodes pour une base données et un critère donné, les deux méthodes se voient attribuées le même rang)

Train	Sylv	ACC	SSE	AUC
Pam	1	5	6	4
Kmeans	1	3	2	4
Kxen	7	1	/	1

Tableau 13 : Nombre de fois où la méthode gagne (sur 9 bases) sur Train

Test	Sylv	ACC	SSE	AUC
Pam	6	6	4	5
Kmeans	3	2	4	2
Kxen	2	1	/	3

Tableau 14 : Nombre de fois où la méthode gagne (sur 9 bases) sur Test

5.5) Discussion sur les résultats

5.5.1) Les techniques d'initialisation

Lors de nos tests sur les techniques d'initialisation des k-moyennes, nous avons été dans l'impossibilité de tester les méthodes d'initialisations PAM et celle des Fast-k médoïdes car elles ont pris beaucoup trop de temps pour se terminer.
Ces deux méthodes avaient les complexités les plus élevées et le fait de ne pas pouvoir les tester nous conduit à rejeter ces deux méthodes comme techniques d'initialisation.

Les résultats sur les méthodes d'initialisation présentent ceux des 3 méthodes suivantes :
- La méthode aléatoire (Partie II section 3.1.1)
- La méthode Sample (Partie II section 3.1.8)
- La méthode Kmeans++ (Partie II section 3.1.2)

Au vu des premiers résultats (Tableaux 5 et 6), on peut constater qu'initier un algorithme des k-moyennes par l'une de ces trois méthodes produit un clustering d'une qualité relativement équivalente.

A la suite de ces premiers résultats, il nous a fallut choisir une méthode pour poursuivre les tests. La méthode Sample semble donner de meilleurs résultats que les autres méthodes, c'est pourquoi nous avons choisi cette méthode pour continuer nos tests avec la technique des k-moyennes.

Nous avons pu ensuite obtenir de nouveaux résultats (Tableaux 6 et 7) sur les techniques d'initialisations qui correspondent à ceux des 3 même méthodes mais avec des valeurs de k plus petites (entre 2 et 10 au lieu de 2 et $\sqrt{n}$).

Ces résultats ont montré que la technique aléatoire et la technique Sample sont relativement équivalentes. La technique Sample est visiblement un peu meilleure avec des valeurs sur les valeurs de k comprise entre 10 et $\sqrt{n}$. Cela aura permis à cette méthode d'obtenir de meilleurs résultats lors de nos premiers tests et d'être choisie.

Cependant l'utilisation faite par l'équipe marketing d'un algorithme de clustering se fera sûrement avec des valeurs de k assez faibles. La méthode aléatoire est celle qui semble obtenir les meilleurs résultats pour de faibles valeurs de k. Donc il serait peut être préférable de reconsidérer notre choix pour la partie de réponse au problème industriel et de choisir la méthode aléatoire.

5.5.2) La représentation brute face à la représentation supervisée.

Nous observons dans le tableau 9 que la méthode PAM et la méthode des k-moyennes donnent de meilleurs résultats en utilisant une représentation supervisée plutôt qu'une représentation brute pour le critère de sylvain, le taux de bonne classification et l'AUC.

Dans les tableaux 10 à 12, on observe que la méthode des k-moyennes utilisée sur une représentation supervisée donne de meilleurs résultats qu'utilisée sur une représentation brute pour le critère de sylvain, le taux de bonne classification et l'AUC.

On ne peut pas ici comparer la représentation supervisée et la représentation brute par le critère SSE car le critère SSE dépend de la représentation des données choisie.

On en déduit que l'utilisation d'une représentation supervisée lors d'un clustering peut permettre d'obtenir souvent de meilleurs résultats que si on avait utilisée une représentation brute.

Il semble vraiment intéressant d'utiliser cette représentation pour réaliser un clustering si l'on possède un ensemble d'entrainement.

5.5. 3) Les méthodes de clustering sur une représentation supervisée

Nous commençons dans cette partie par l'observation des courbes présentées en Annexe 35 à 41.
Sur ces courbes, il semble que la méthode PAM soit meilleure que les autres méthodes sur tous les critères sauf celui de sylvain à partir d'un k suffisamment grand.

Les courbes de la méthode des k-moyennes sur une représentation supervisée se placent entre celles de la méthode PAM sur une représentation supervisée et celles de Kxen.

La méthode Kxen se place en dessous de toutes les méthodes sauf pour le critère de sylvain ou elle obtient de meilleurs résultats.

L'algorithme de classification du SNB fonctionne sans définir k. Il est nettement meilleur que les algorithmes de clustering testés si k est petit. Mais elle se fait battre assez souvent par PAM et le k-moyennes avec un k assez grand (10 ou 20 par exemple). On voit très bien ce phénomène dans l'Annexe 36 (Hepatitis) et sur l'Annexe 37 (Hypothyroid).

Il n'y a aucune méthode qui fasse moins bien que le Baseline (le classifieur majorité) en ce qui concerne le taux de classification.

Nous poursuivons en critiquant les deux algorithmes qui n'ont pas fonctionné sur l'ensemble entier de bases testées.

Le premier de ces deux algorithmes est PAM qui n'a pas fonctionné sur Shuttles (58000 instances) et sur Letter (20 000 instances).

Cet algorithme est adapté pour de « petites » bases mais visiblement à partir de 20 000 instances à traiter cet algorithme semble montrer ses limites. Ceci peut être expliqué par le fait que l'algorithme commence par calculer les distances entre toutes les instances et les met en mémoire avant de commencer la partie principale de l'algorithme.

Cet algorithme ne pourra donc pas convenir à une utilisation dans le contexte industriel d'Orange, car les bases d'Orange sont par nature très grandes (plusieurs milliers d'instances et plusieurs milliers de variables explicative).

Le deuxième de ces deux algorithmes est Kxen qui n'a pas fonctionné sur Phonème.

Ce problème est dû à l'utilisation de Kxen mode « script » en java. Lors du processus Kxen, par moment et pour certaines bases de données les noms variables sont renommés. Dans ce cas le résultat fourni par Kxen contient une instance de trop, ne sachant pas si l'instance supplémentaire est au début, à la fin, ou n'importe où ailleurs, on ne peut utiliser les résultats fourni par Kxen dans ces cas particuliers. On aurait pu utiliser Kxen en mode « manuel » (Avec l'interface graphique) pour obtenir les résultats cela aurait fonctionné, mais au vu de l'ensemble des bases testées et des valeurs de k différentes cela aurait été une tache trop fastidieuse pour être réalisée. De plus pour un usage industriel, il est nécessaire de pouvoir travailler en mode « script ».

Poursuivons en observant les tableaux de l'annexe 45 présentant les résultats moyens (par rapport à k) des méthodes testées et les deux tableaux résumés (13 et 14 de la section 5.4). Ces deux tableaux présentent le nombre de fois qu'une méthode gagne par rapport aux autres.

Sur ces tableaux, on observe qu'en moyenne le SNB obtient de meilleurs résultats que toutes les autres méthodes en taux de bonne classification (ACC) et en AUC. Ceci peut expliquer par le fait que le SNB est un algorithme de classification. Il est donc meilleur que les méthodes de clustering ayant un k petit. C'est pourquoi en réalisant la moyenne pour des valeurs de k allant de 2 à 10, (puis 20, 40, 80, 160 en s'arrêtant à $\sqrt{n}$) le SNB obtient de meilleurs résultats.

On observe également sur ces tableaux que Kxen finit 7 fois premier sur le critère de Sylvain alors que l'algorithme des k-moyennes et celui de Pam ne finissent qu'une fois premier.

Cela rejoint l'observation des courbes et cela tend à montrer que le clustering réalisé par Kxen est de meilleure qualité que celui réaliser par d'autres méthodes. C'est un résultat assez surprenant. Il serait intéressant de mieux comprendre pourquoi la méthode Kxen obtient de si bon résultats avec le critère de Sylvain alors qu'elle donne des résultats décevant sur les critères de classification.

Pour conclure, la méthode PAM donne des résultats très satisfaisant sur de petite base de données mais n'est pas adapté aux grandes bases de données.

La méthode des k-moyennes permet d'avoir des résultats très proche de ceux de PAM, et peut être utilisé sur de grandes bases de données.

Nous conclurons cette étude sur le clustering sur une représentation supervisée dans la section suivante.

6) Apport de notre étude sur le clustering utilisant une représentation supervisée.

Les méthodes d'initialisations de l'algorithme des k-moyennes testées sont équivalentes en termes de qualité du clustering.

Dans le cadre de notre étude, la méthode Sample est celle qui nous semble donner des résultats un peu meilleurs que les autres. C'est cette méthode que nous avons choisi d'utiliser pour la suite de nos tests avec la technique des k-moyennes.
Si nous utilisons un clustering pour la partie industrielle nous choisirons la méthode Sample comme méthode d'initialisation.

Notre comparaison entre l'utilisation de deux méthodes de clustering sur une représentation brute et sur une représentation supervisée nous a permis de constater que l'utilisation d'une représentation supervisée peut très souvent améliorer la qualité d'un clustering.

Nous avons vu que nous pouvions trouver de meilleurs résultats que ceux de Kxen en termes de classification en utilisant une représentation supervisée avec un algorithme de clustering comme PAM ou les k-moyennes.

Une technique de clustering utilisée utilisant une représentation supervisée avec un k élevé peut être une technique de classification, souvent meilleure que le classifieur Bayesien Naïf Sélectif (SNB).

La méthode PAM n'est pas adaptée aux grandes volumétries. Il n'est pas envisageable d'utiliser cette méthode pour répondre à la problématique industrielle d'Orange.

Au cours de cette étude sur le clustering utilisant une représentation supervisée, nous avons été très déçus par la méthode des Fast K médoïdes. Elle semblait avoir un très grand potentiel en ayant la rapidité de la technique des k-moyennes et la robustesse de la méthode des k médoïdes. Mais cette méthode a été très lente, et aucun résultat avec cette méthode n'aura pu être présenté dans ce rapport.

Pour conclure cette étude sur le clustering utilisant une représentation supervisée, si nous devions choisir une méthode pour réaliser un clustering de qualité sur des bases de données de grande volumétrie, nous choisirions d'utiliser un algorithme des k-moyennes utilisant la norme L1, initialisé avec la méthode Sample et appliqué sur une représentation supervisée des données faites par khiops et kawab (Si nous disposons d'un ensemble d'entraiment).

Partie III) Application à une problématique industrielle.

1) Problème industriel.

1.1) Le contexte industriel

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 est calculée à partir d'un modèle prédictif (dans notre cas un classifieur naïf de Bayes) pour un ensemble de clients (le périmètre de la campagne).

Les clients sont ensuite triés par rapport à leur probabilité d'appétence au produit. Le service marketing ne contacte ensuite que les clients les plus appétant c'est-à-dire ceux ayant la plus forte probabilité d'acheter le produit. En parallèle et avant le avant le contact commercial, il peut être intéressant de réaliser une segmentation des clients qui seront contactés. L'idée est de pouvoir réaliser des campagnes marketing personnalisées pour les segments de clients.

Si ces campagnes personnalisées sont réalisées pour un mois M, l'équipe marketing souhaiterait ne pas devoir recommencer la préparation des campagnes au mois M+i pour de nouvelles données. Ils aimeraient que l'on puisse identifier les nouveaux clients du mois M+i, par rapport aux segments calculés et interprétés au mois M.

Pour le moment il existe une solution logicielle au sein de l'entreprise pour réaliser ce type de campagne. Mais la solution actuelle n'est quasiment pas utilisée car les groupes obtenus deviennent trop différents de mois en mois.

On se propose d'étudier la solution actuelle, et de chercher dans un premier temps s'il est possible de l'améliorer en utilisant une représentation supervisée des données puis dans un deuxième temps d'envisager d'autres pistes de recherche possible si nécessaire pour améliorer la solution existante.

1.2) La solution actuelle

- Le processus actuel

La solution actuelle repose sur Kxen. D'une part les clients sont classés par K2R le module de régression et de classification de Kxen. Et d'autre part la segmentation est effectuée par K2S le module de segmentation de Kxen.

On différentie deux parties dans le processus de création des groupes pour faire des campagnes marketing personnalisées. Il y a la création des premiers groupes de clients que l'on appelle la phase d'apprentissage et il y a l'identification des nouveaux clients au mois M+i à des groupes déjà existant que l'on appelle la phase d'exécution.

Nous appelons base de modélisation du classifieur (BDMC) une petite base de clients extraite de la base de déploiement. Elle correspond à une base d'apprentissage.

Nous appelons base de déploiement (BDD), la base des clients sur lesquels se porte l'analyse.

1) On entraine le classifieur K2R à partir de la BDMC. A la suite de cet apprentissage, on le sauvegarde dans un modèle K2R.

2) A partir de ce modèle et de la BDD, des scores sont générés pour chacun des éléments de la BDD.

3) Les éléments de la BDD possédant les meilleurs scores sont sélectionnés pour former la base des meilleurs clients (Top Scores).

4) Le module de segmentation K2S est appliqué sur la base Top Score. K2S permet d'obtenir des groupes à partir des clients de la base Top Score et de sauvegarder les centres des groupes dans un modèle K2S.

La figure suivante présente un organigramme illustrant ce processus :

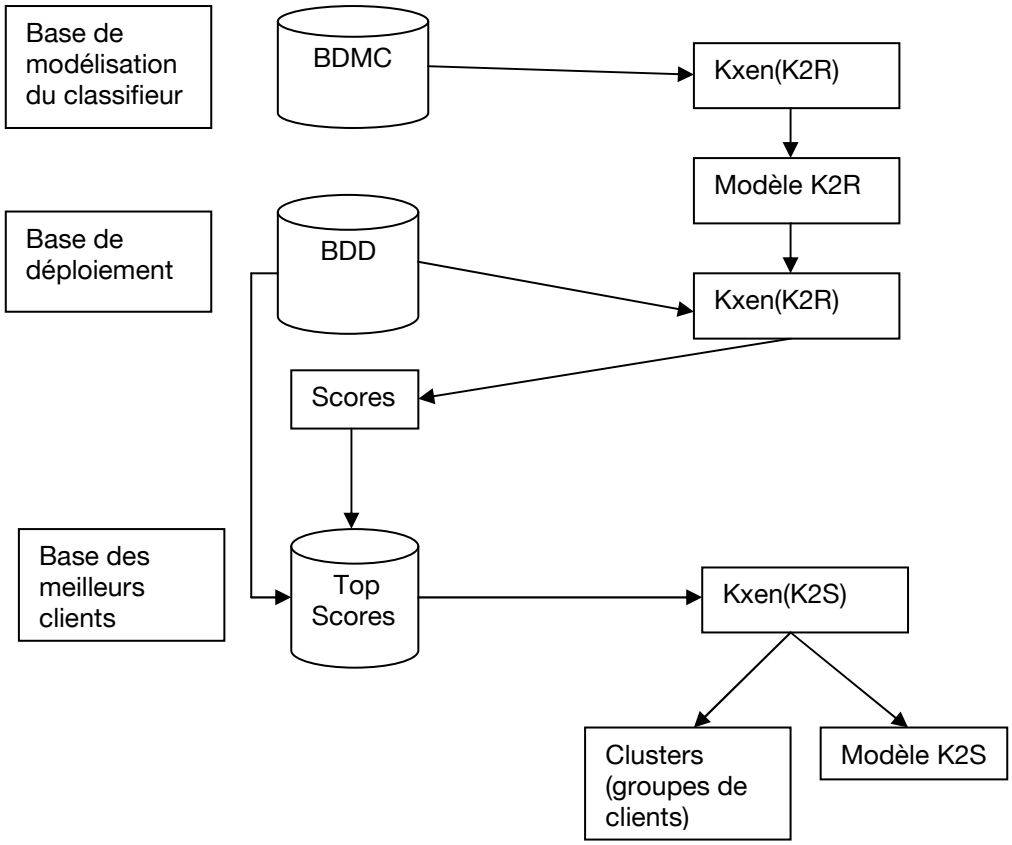


Figure 25: Organigramme de la solution actuelle (apprentissage)

La phase d'exécution permet de traiter une nouvelle base de clients au mois M+i en réutilisant les modèles crées lors de la phase d'apprentissage.

La phase d'exécution se déroule ainsi :

1) A partir du modèle K2R et de la nouvelle base de client, des scores sont générés pour chacun des éléments de la nouvelle base.
2) A partir de ces scores, les meilleurs clients sont sélectionnés pour former la base des Top Scores.
3) A partir du modèle K2S (les centres), chaque élément de la nouvelle base est assigné par K2S à un groupe.

La figure suivante présente un organigramme illustrant ce processus :

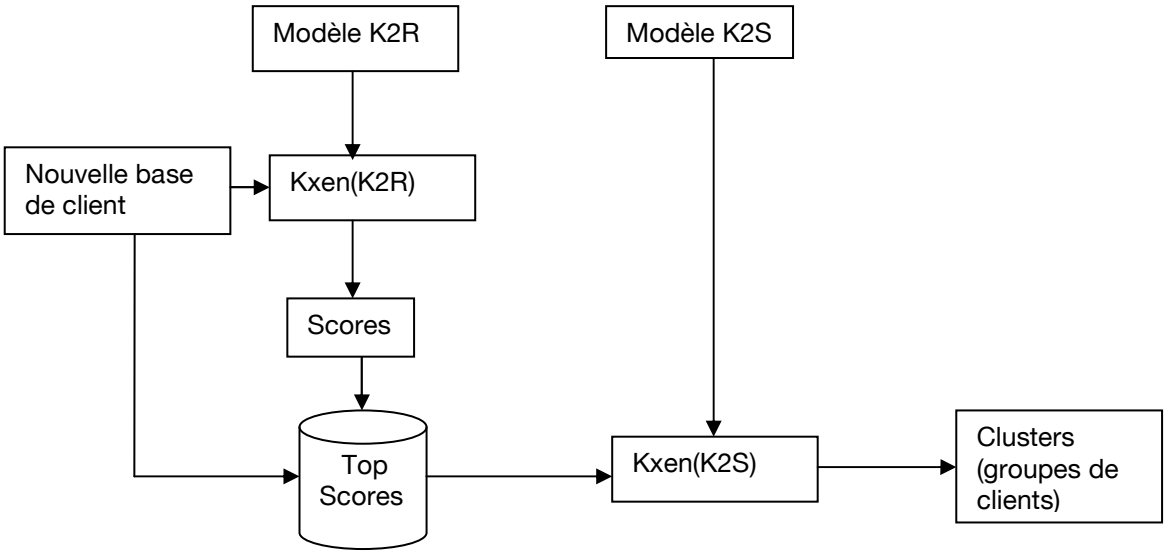


Figure 26: Organigramme de la solution actuelle (exécution)

- Le type de données traitées

Un client est représenté par des milliers de variables comme le type d'abonnement qu'il possède, ses consommations, son âge, et beaucoup d'autres. Ces variables sont de type mixte, certaines sont catégorielles et d'autres continues.

Cet ensemble de variables est de l'ordre de 500 à 1000 variables. C'est un nombre de variables trop important pour pouvoir être traité correctement par un algorithme de clustering. En effet un grand nombre de variables implique un temps beaucoup plus long de l'exécution de la technique de clustering. Et de plus les techniques de clustering basées sur le partitionnement perdent de leur efficacité avec des grandes dimensions car les éléments tendent à se ressembler dans de très grandes dimensions.

Kxen réalise un prétraitement sur les données avant d'appliquer son module de classification K2R, ou son module de segmentation K2S.

Le prétraitement effectué dépend des données sur lesquels il est appliqué.
K2R et K2S seront appliqués sur des bases de tailles différentes et si les données ne sont pas stationnaires alors il se peut que les données ne se ressemblent pas d'une base à l'autre.

On pense que les prétraitements effectués pourront être vraiment différents entre K2R et K2S puisse que les bases sont différentes.

- La manière d'utiliser les résultats obtenus

Le but de l'équipe markéting est de pouvoir regrouper les données en 3 ou 4 clusters. L'équipe marketing essaiera ensuite de choisir, d'une part le type de campagnes qui vont être réalisées, puis d'autre part la manière de réaliser ces campagnes.

Le « type de campagnes » correspond à la manière de joindre les clients (Envoi d'emails, appel téléphonique, envoi de lettres)

La « manière de réaliser les campagnes » correspond à l'argumentaire qui va être choisi lors des campagnes. Le but est de pouvoir reconnaitre des types de clients, et leurs proposer un message qui leurs serait plus adapté, de manière à avoir plus de chance de les toucher.

Il est vraiment intéressant pour l'équipe marketing de pouvoir analyser simplement les variables qui définissent un cluster (exemple: les hommes, entre 41 et 52 ans qui ont une offre triple play, et pas d'abonnement TV).

Une analyse des variables au sein d'un cluster pourrait être envisagée pour répondre à ce type de besoin.

1.3) Le problème avec la solution actuelle

La solution actuelle fonctionne correctement sur une petite période. Mais après cette période les groupes trouvés d'un mois à l'autre deviennent trop différents.

L'équipe marketing souhaiterait que les clusters soient plus stables au cours du temps. Il faut donc définir cette notion de stabilité du clustering au cours du temps.

La notion de stabilité

Pour le moment, on définit cette notion de stabilité par deux critères.

Le premier est l'évolution du pourcentage d'appartenance à un cluster. Au mois M, 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 devrait rester les même pour que l'on considère la solution comme stable par rapport à ce critère.

Le deuxième critère 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 M, on observe dans les clusters la répartition des clients appartenant à une classe. On recommence l'opération aux 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.

Exemple :
Au mois M :
Soit un ensemble X_1 avec 2 classe (0 et 1)
On réalise un clustering sur X_1 avec k = 2.
On mesure la répartition des classes, et on obtient :

- Pour le cluster 1, 20% des éléments sont de la classe 0 et 80% sont de la classe 1.
- Pour le cluster 2, 60% des éléments sont de la classe 0 et 40% sont de la classe 1.

Au mois M+i :
On projette les nouvelles données sur les clusters appris au mois M. Puis on mesure la répartition des classes avec un autre ensemble de données X_2, à un autre mois.

Si la répartition des classes reste à peu près les même d'un mois à l'autre, on considère la solution comme stable par rapport au deuxième critère.

Remarque :
Ces critères sont proches des souhaits de l'équipe marketing mais ils ne correspondent peut être pas à ce que l'on devrait attendre de la stabilité d'un clustering. En effet si les données changent radicalement d'un mois à l'autre il est normal que le résultat du clustering soit différent d'un mois à l'autre.

Le problème d'interprétabilité des données.

En fait au-delà du problème de stabilité des clusters, c'est un problème de capacité à interpréter les données qui se pose.

En effet une fois que la phase d'apprentissage a été effectuée, l'équipe marketing va chercher à identifier quels profils ont les clients appartenant à un cluster. Puis l'équipe marketing va réaliser une campagne personnalisée aux profils de clients identifiés.
Si le profil des clients d'un cluster change trop, la campagne construite ne leurs correspondra plus. Il faudra donc la refaire, et c'est cela qui conduit l'équipe marketing à ne pas utiliser la solution actuelle.

2) Amélioration du processus de clustering.

2.1) Méthodes envisagées pour résoudre le problème.

2.1.1) K-moyennes basé sur les Top Scores représentés à partir de connaissance supervisée

Cette méthode est celle que l'on envisage de réaliser et de tester pour tenter de répondre au problème posé.

Description de la méthode

Cette méthode est composée comme la méthode actuelle en deux parties, une partie d'apprentissage et une partie d'exécution.

Nous appelons base de modélisation du classifieur (BDMC) une petite base de clients extraite de la base de déploiement. Elle correspond à une base d'apprentissage.

Nous appelons base de déploiement (BDD), la base des clients sur lesquels se porte l'analyse.

1) On entraine le classifieur Naïf Bayes sélectif (SNB) de Khiops à partir de la BDMC. A la suite de cet apprentissage, on le sauvegarde dans un modèle Khiops.

2) A partir de ce modèle et de la BDD, des scores sont générés pour chacun des éléments de la BDD.

3) Les éléments de la BDD possédant les meilleurs scores sont sélectionnés pour former la base des meilleurs clients (Top Scores).

4) A partir du modèle Khiops et de la base Top Scores, on génère une base des meilleurs clients dans une représentation supervisée en utilisant Kawab. (Voir Partie II section 4.4)

5) Un algorithme de clustering basé sur le partitionnement est appliqué sur la base Top Score en représentation supervisée. Cet algorithme permet d'obtenir des groupes à partir des clients de la base Top Score.

6) On sauvegarde les centres des clusters trouvés dans un fichier.

La figure suivante présente un organigramme illustrant ce processus :

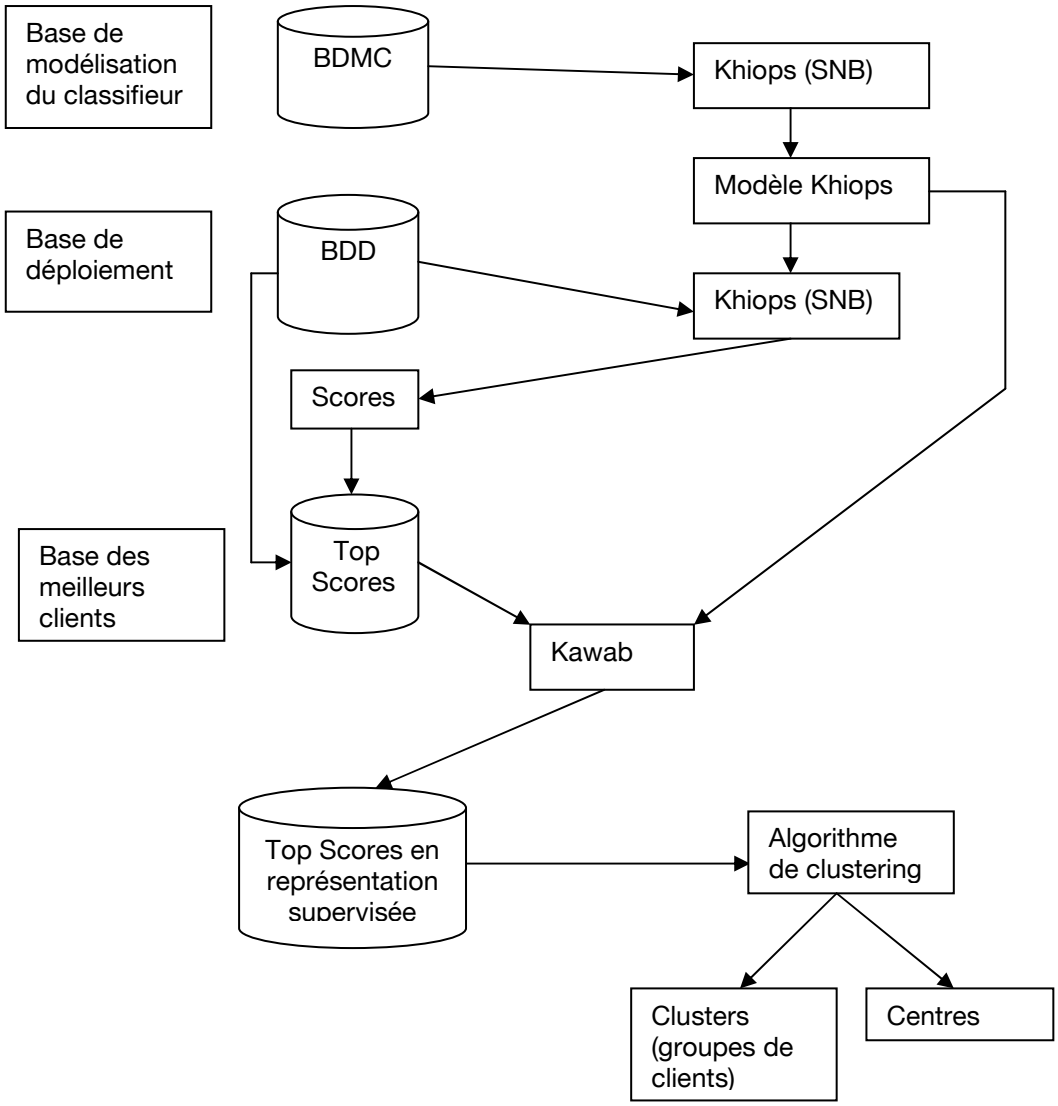


Figure 25: Organigramme de la solution proposée (apprentissage)

La phase d'exécution se déroule ainsi :

1) A partir du modèle Khiops et de la nouvelle base de client, des scores sont générés pour chacun des éléments de la nouvelle base.

2) A partir de ces scores, les meilleurs clients sont sélectionnés pour former la base des Top Scores.

3) A partir du modèle Khiops et de la base Top Scores, on génère une base des meilleurs clients dans une représentation supervisée en utilisant Kawab.

4) A partir des centres trouvés lors de l'apprentissage, un groupe est assigné par l'algorithme de clustering choisi à chaque élément de la nouvelle base.

La figure suivante présente un organigramme illustrant ce processus :

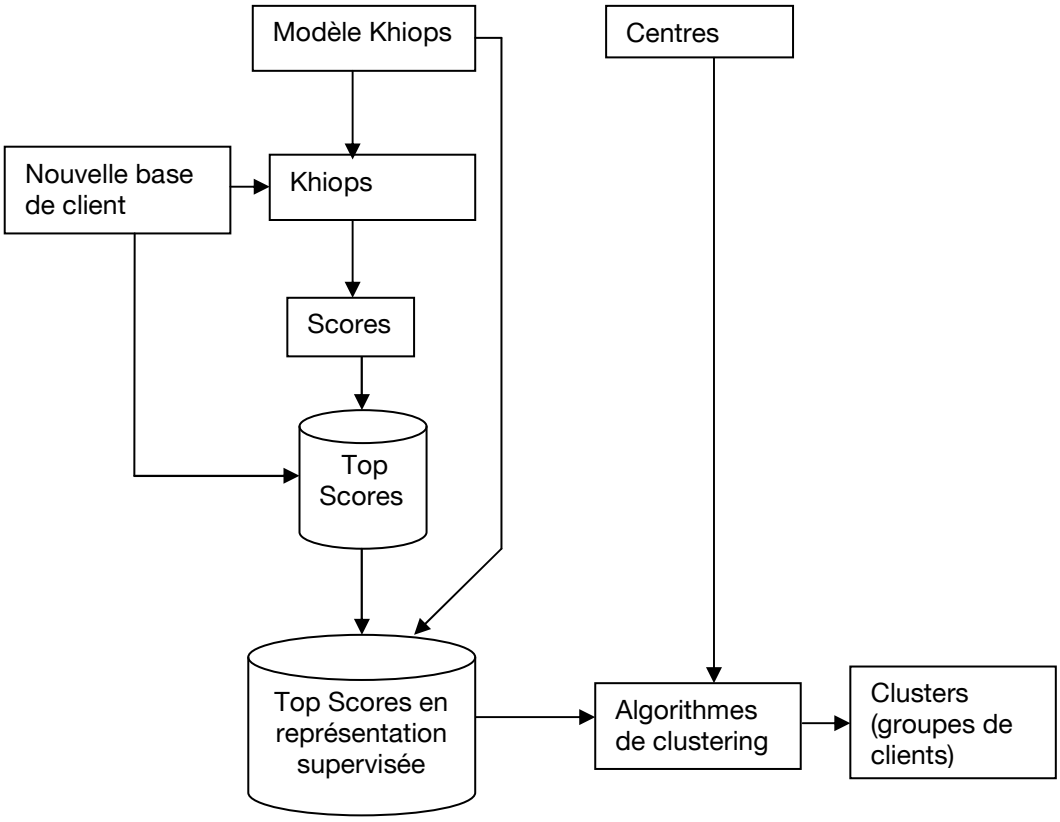


Figure 26: Organigramme de la solution proposée (exécution)

Les différences de cette méthode par rapport à la méthode actuelle.

Les prétraitements effectués par la solution actuelle et par celle proposée sont différents.

Le premier prétraitement effectué lors du processus est celui appliqué sur les données de l'ensemble d'entrainement (BDMC) avant de créer un modèle de classifieur.

Pour la solution actuelle il est réalisé par Kxen (K2R).
Pour la solution proposée, il est réalisé par Khiops en utilisant la discrétisation MODL.

Nous pensons que le prétraitement MODL, discrétisera mieux les données que celui de kxen.

Le deuxième prétraitement effectué est celui appliqué sur les Top Scores avant d'appliquer une méthode de clustering dessus.

La méthode actuelle traite les données indépendamment de premier traitement.
Alors que la méthode proposée vise à conserver une information sur la classification pour réaliser ce prétraitement. Dans la méthode proposée, ce prétraitement correspond transformer les données sous la forme d'une représentation supervisée.

Il serait possible d'envisager d'autres solutions pour résoudre les problèmes de la solution actuelle (Partie III section 1.2). Ces méthodes sont décrites ci-dessous.

2.1.2) K-moyenne basé sur les valeurs positives

Voici le principe de cette méthode :

- On créer un classifieur à partir de l'ensemble d'entraiment (BDM).

- On applique ce classifieur à l'ensemble de données (BDD) et on obtient une classification de nos données (certains clients seront classés comme positifs et d'autres négatifs). Les clients classés comme positifs sont ceux qui sont les plus susceptibles de répondre à une campagne marketing.

- On sélectionne ensuite seulement les clients qui ont été classés comme positifs et on les met dans une base.

- On applique une méthode de clustering sur cette base. Et on obtient des groupes de clients.

A première vue cette méthode semble pouvoir fonctionner mais elle suppose que suffisamment de clients soient classés comme positif. Or dans les bases clients il y a beaucoup plus de client négatif que de client positif. (C'est à dire qu'il y a plus de client que l'on n'a pas contacter que de client qui ont été contactés et qui ont répondu positivement)
C'est pourquoi cette méthode ne semble pas pouvoir fonctionner.

2.1.3) Gonfler les clusters

Cette méthode a pour but d'essayer d'améliorer l'évolution du pourcentage d'appartenance à un cluster. Elle est basée sur le processus décrit dans la Partie III section 2.1.1) en apportant une idée supplémentaire pour la stabilité.

Voici le principe de cette méthode:
- Au Mois M, on applique le processus standard (décrit dans la Partie III section 2.1.1)

- Aux mois suivants, si le pourcentage d'appartenance à un cluster est différent de celui du mois M, on change un peu la fonction d'appartenance à un cluster pour que le pourcentage

d'appartenance à un cluster soit respecté. Habituellement cette fonction d'appartenance est basée sur une distance par rapport au centre le plus proche. Ici il s'agira d'utiliser une fonction pondérée progressivement jusqu'à avoir atteint le pourcentage d'appartenance aux clusters le plus identique possible à celui du mois M.

Nous testerons cette méthode en fin de stage s'il nous reste du temps pour le faire.

2.1.4) K-moyennes sur tout le monde puis effectuer une sélection top score ensuite.

Cette méthode est fondée sur l'idée que l'application d'un k-moyennes serait plus performante si la base sur laquelle il est appliqué est la plus grande possible.

Le principe de cette méthode est le suivant:

- On effectue dans un premier temps un k-moyennes sur la bases de donnés (BDD).
- On crée un classifieur à partir de la base d'entrainement (BDM).
- On applique ce classifieur à la base de données (BDD) et on obtient des scores pour chacun des clients.
- On sélectionne ensuite les clients avec les meilleurs scores pour former la base des Top Scores.

Les clients auront été répartis en groupes utilisables par l'équipe Markéting, et seulement les meilleurs clients auront été sélectionnés (donc ceux qui ont la plus forte probabilité de répondre positivement à la campagne).

Nous testerons cette méthode en fin de stage s'il nous reste du temps pour le faire.

2.2) La démarche d'évaluation.

2.2.1) Les méthodes évaluées

Nous avons choisi d'évaluer deux méthodes dans cette partie.
La première méthode qui sera testée est celle que nous proposons. (Elle est décrite dans la Partie III section 2.1.1)
La deuxième méthode testée sera la méthode actuelle basée sur l'utilisation de Kxen (Elle est décrite dans la Partie III section 1.2).

2.2.2) Les données utilisées lors des tests.

Plusieurs bases de données ont été mises à notre disposition pour cette phase de tests.
Nous possédons 3 bases de 200 000 clients datant de Mars, Mai, et Août 2009.

Ces bases sont constituées d'un grand nombre de variables (environs 1000).

Nous utiliserons la base de Mars comme ensemble d'apprentissage pour le classifieur.
Nous utiliserons les tops scores de la base Mars comme ensemble d'apprentissage pour la segmentation.
Les bases de Mai et Août correspondront à des ensembles de tests.

Les critères d'évaluations seront calculés pour chacun des mois (Mars, Mai et Août).

2.2.3) Les critères d'évaluation de la qualité des méthodes.

Nous évaluerons cette méthode par les deux critères de stabilité introduit dans la Partie III section 1.1:
- L'évolution du pourcentage d'appartenance à un cluster.
- L'évolution de la répartition des classes ou des valeurs cibles au sein des clusters.

Ces critères seront présentés sous forme de courbe au cours des mois.

2.3) Présentation des résultats obtenus

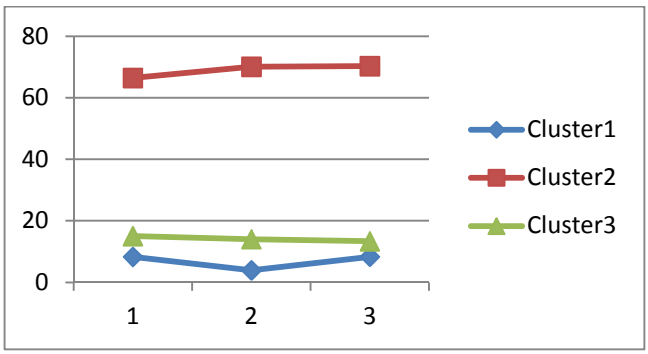
Je vous présente ici les 2 types de résultats que l'on a pu obtenir au cours de ces tests :

2.3.1) Test avec k=3 et Top Score =5%

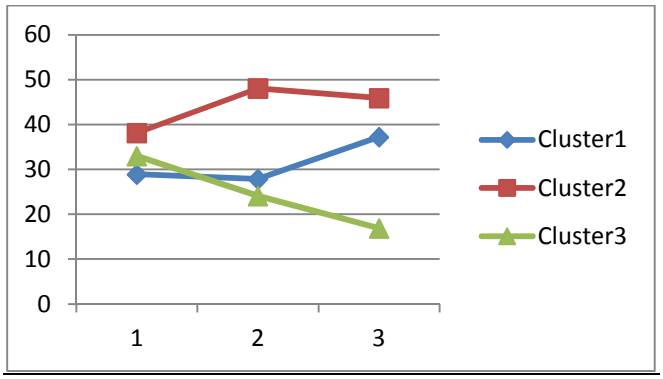
Les résultats suivants sont ceux obtenus à la suite du processus kxen et de celui des K^3.
Par rapport à nos deux critères de stabilité :
- Le pourcentage d'élément par cluster
- La proportion d'élément d'une classe donnée au sein d'un cluster.

Ils sont également disponibles dans l'annexe 47.

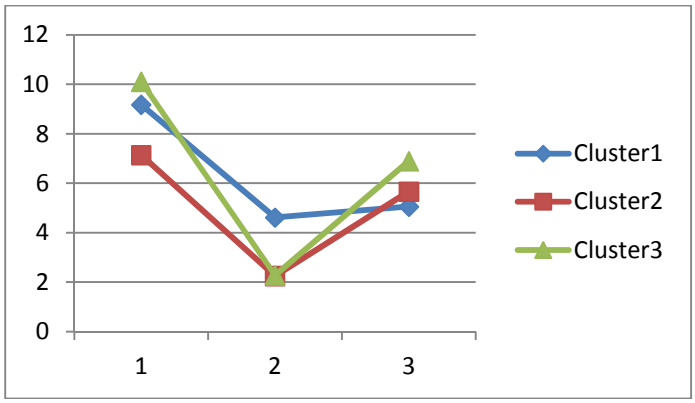
Pourcentage d'éléments par cluster (Kxen)



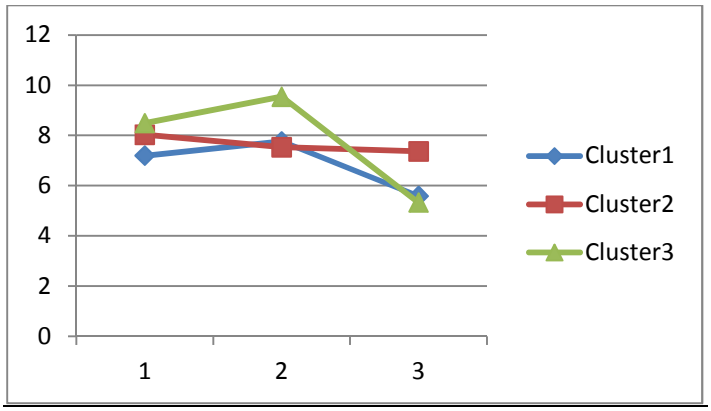
Pourcentage d'éléments par cluster (K^3)



Proportion d'éléments (CIBLE_CHURN=1) par cluster (Kxen)



Proportion d'éléments (CIBLE_CHURN=1) par cluster (K^3)



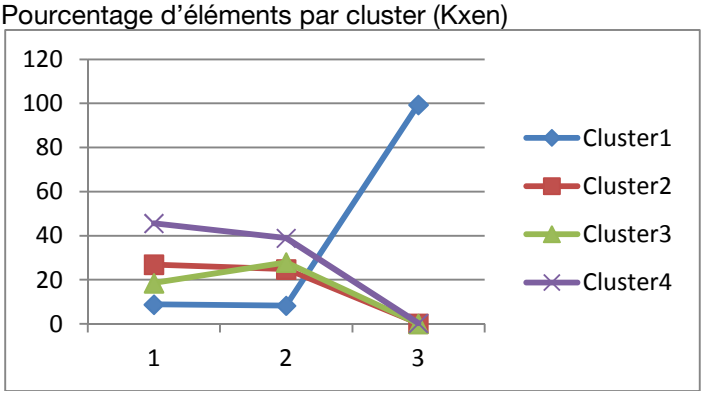
2.3.2) Test avec k=4 et Top Score =5%

Les résultats suivants sont ceux obtenus à la suite du processus kxen et de celui des K^3.
Par rapport à nos deux critères de stabilité :

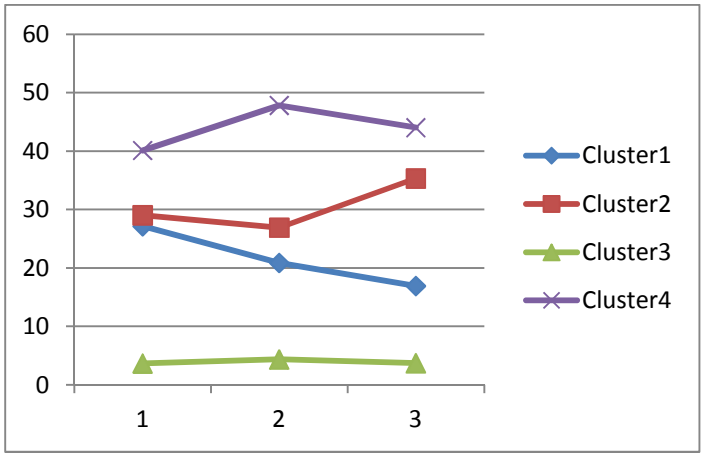
- Le pourcentage d'élément par cluster
- La proportion d'élément d'une classe donnée au sein d'un cluster.

Ils sont également disponibles dans l'annexe 48.

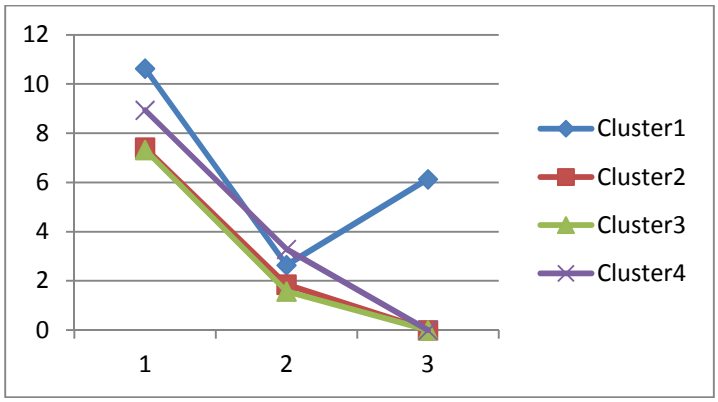
Pourcentage d'éléments par cluster (Kxen)



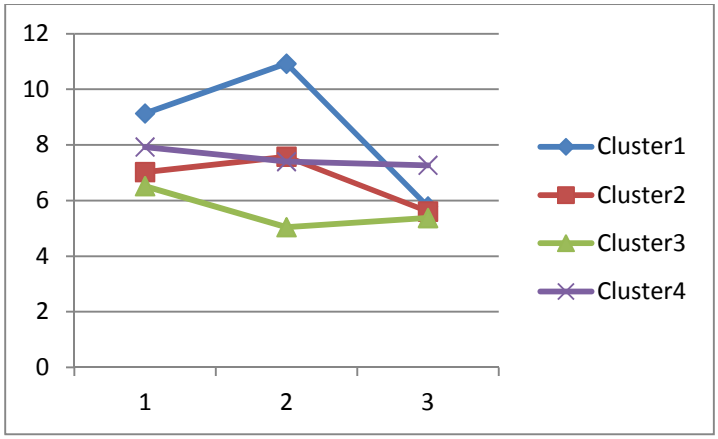
Pourcentage d'éléments par cluster (K^3)



Proportion d'éléments (CIBLE_CHURN=1) par cluster (Kxen)



Proportion d'éléments (CIBLE_CHURN=1) par cluster (K^3)



2.3.3) Test avec k=5 et + et Top Score =5%

Les résultats suivants sont ceux obtenus à la suite du processus kxen et de celui des K^3.

Tous les résultats trouvés pour une valeur de k supérieur ou égale à 5 ressemblent à ceux avec k=4.
Ils sont disponibles dans les annexes 49 à 51.

2.4) Interprétations et critiques des résultats obtenus.

Les résultats obtenus avec k=3, semble faire apparaître que le processus basé sur Kxen est tout à fait stable voir même plus que K^3 (Khiops, Kawab, K-moyennes) par rapport aux proportions d'éléments par cluster.
En revanche il semble que par rapport à la proportion d'éléments de classe 1 au sein d'un cluster le processus K^3 est un peu plus stable.

Ob observe également dans ces tests que le processus K^3 repartit les données de façon plus homogène à l'intérieur des clusters alors que le processus Kxen crée un gros cluster avec plus de 60% des éléments et 2 petits avec moins de 20 % des éléments.

Les résultats obtenus avec k=4 et un k plus grand montrent un phénomène étrange pour le processus Kxen.
En effet pour le dernier mois, ce processus associe plus de 90% des éléments à un cluster.
On observe bien que dans ces conditions les clusters trouvés ne peuvent plus être interprétés.

Donc à partir de k=4, le processus K^3 est plus stable au cours du temps que celui basé sur kxen par rapport aux deux critères de stabilités choisis.

Aucune des solutions ne nous permet d'obtenir une stabilité parfaite, ce qui est tout à fait cohérent puisse que les données ne sont pas stationnaires, il est normal que les clusters changent un peu au cours du temps.

Remarque 1 :
Le processus testé n'est pas exactement celui utilisé dans l'équipe marketing. En effet lorsque ce processus est utilisé dans l'équipe marketing, il est utilisé de manière itérative c'est-à-dire qu'on applique plusieurs fois, et qu'à chaque itération on retire les variables les moins informatives jusqu'à obtenir un nombre de variables très réduit. Mais on pourrait tout aussi bien réaliser cette manière de faire avec K^3. Donc cela n'a pas d'influence sur les tests réalisés.

Remarque 2 :
Il y a une différence importante entre le processus Kxen et le processusK^3. Cette différence concerne la gestion des variables.
En effet la partie Khiops de K^3 élimine automatiquement un certain nombre de variables non informatives. Le nombre de variables passe donc de 800 à 300. Puis avec l'application de kawab le nombre de variables passe à 600.

Dans le cas de Kxen, certaine variables sont discrétisées et regroupées mais le nombre de variables reste constant tout au long du processus. Il travaille donc certainement sur 800 variables. Lors de l'application de la segmentation.

Ces résultats tendent à montrer que la solution proposée fonctionne de manière suffisamment satisfaisante pour être utilisée par l'équipe marketing et qu'elle est plus stable que celle basée sur kxen.

Il serait très intéressant d'utiliser ces deux processus de manière itérative, et de comparer les résultats obtenus.

Partie IV) Premier bilan du stage

Il reste une partie du stage à terminer cependant il déjà possible de réaliser un premier bilan de ce stage. Nous aborderons dans un premier temps l'intérêt qu'a pu représenter ce stage pour l'entreprise. Dans un deuxième temps nous dégagerons l'intérêt personnel de ce stage sur le plan humain et technique. Nous conclurons ensuite sur les perspectives ouvertes par ce stage.

1) Intérêt du stage pour l'entreprise

Ce stage aura eu un intérêt particulier pour l'équipe PROF. L'équipe PROF au cours de ses recherches en Data Mining et en Profiling, constitue une grande base de connaissances. Ce stage aura permis à l'équipe d'approfondir ses connaissances en particulier au sujet de l'influence d'une représentation supervisée pour la réalisation d'un clustering basé sur le partitionnement.

La deuxième partie de ce stage permettra certainement de trouver une méthode qui sera stable dans le temps et qui permettra de réaliser une solution satisfaisante pour l'équipe markéting.

Si les recherches dans cette deuxième partie s'avéraient non-concluantes dans cette voie, nous aurons tout de même exploré l'une des voies envisagées pour répondre au problème de l'équipe Markéting. Lors de ma présentation et grâce à l'équipe PROF, nous avons envisagé d'autres méthodes pour répondre au problème de l'équipe Marketing.
Ces autres méthodes pourront éventuellement être testées à la fin de mon stage si le temps restant le permet, ou bien pourront donner lieu à un autre stage.

On aura donc fait avancer la recherche de l'équipe PROF et de l'entreprise sur l'influence d'une représentation supervisée de connaissance lors d'un clustering, et sur les méthodes qui peuvent fonctionner ou pas pour répondre au problème de stabilité de l'équipe Marketing.

2) Intérêt personnel

D'un point de vue humain, ce stage a été très enrichissant. Voici les points que je voudrais mettre en avant:

- Cela aura été l'occasion de découvrir le fonctionnement d'une équipe de Recherche et Développement dans une grande entreprise.
- Il aura été très enrichissant de s'intéresser au travail de la plus part des membres de l'équipe, découvrir les points sur lesquels ils cherchent et leurs objectifs.
- Autonomie
- Au cours de mon stage, j'ai eu l'occasion d'échanger des idées avec Mohamed Dermouche sur nos stages respectifs. Nous nous sommes aidés durant ce stage sur nos problèmes respectifs, et cela nous aura beaucoup profité.

Sur le plan technique, ce stage a également été très instructif. J'ai notamment pu enrichir mes connaissances en technique de clustering. J'ai découvert des techniques que je ne connaissais pas comme les K-médoïdes, les k-médianes, et les k-modes. Au cours de mes recherches j'ai découvert des algorithmes de classification comme le classifieur Bayesien Naïf Sélectif, ou les plus proche voisins basé sur des partitions de Voronoï.

3) Conclusion

Perspectives ouvertes

- Les recherches effectuées lors de ce stage aboutiront à la création d'une solution logicielle qui permettra de traiter les données client de manière stable au cours du temps.
Pour l'équipe markéting, cela permettra à terme de leurs proposer une solution viable pour réaliser des campagnes markéting personnalisées aux clients. Cela leurs permettra également d'appliquer un discours adapté aux clients, et un mode de contact (téléphonique, e-mail, lettres) adapté.

- Une fois cette solution logicielle réalisée, elle pourrait être intégré à la solution PAC (Plateforme d'analyse clients) actuellement en fin de développement.

- Ce stage pourrait déboucher sur la rédaction et la publication d'un article sur l'influence d'une représentation supervisée lors d'une clustering.

- Une seule méthode pour répondre au problème industriel va être testée lors de ce stage. Mais d'autres méthodes avaient été proposées par l'équipe PROF au cours de ma présentation. Il serait intéressant de tester les méthodes qui ont été proposées en comparaison à la notre.

- A postériori, j'aurais pu être plus réactif lors de la phase de test, et décider plus vite de diminuer la quantité de tests effectués.

La pertinence de la formation par rapport au stage

Ma formation m'a permis d'acquérir les connaissances en statistiques et en mathématiques, et l'autonomie nécessaire à ce stage.

Au cours de la majeure SCIA, j'ai pu connaître des méthodes de classification comme le classifieur Bayesien Naïf et la technique des k-moyennes qui constitue une base vraiment utile pour ce stage.

J'ai également pu utiliser mes compétences en programmation Matlab, Java, et Python acquises durant ma formation à EPITA.

La prochaine étape

Nous avons montré que la méthode proposée marche bien de manière individuelle. Et quelle se comporte mieux que la méthode actuelle basée sur Kxen.

Cette méthode semble donc pouvoir être utilisée par l'équipe marketing et pourra amener à la création d'un logiciel l'implémentant.

Cependant il reste certains points qu'il serait intéressant d'approfondir avant de créer un logiciel implémentant cette méthode.

Ces points sont les suivants :

- Comparer les deux processus de manières itératives
- Tester les méthodes proposées par l'équipe
- Réaliser les tests de stabilités sur plus de base clients.
- Utiliser la somme des distances en L1 au lieu du SSE car le SSE n'est un critère approprié vu qu'on utilise la norme L1
- Prendre en compte le fait que faire des clusters en L1 ne garanti pas le compromis variance inter/variance intra

Il aurait surement été possible pour moi de prolonger ce stage, et de développer le logiciel qui en résultera, ce qui aurait vraiment pu être intéressant.
Cependant je veux retourner en Ile-de-France, et trouver un travail là-bas ci-possible dans le datamining ou dans la gestion de projet.

Bibliographie

[1] - k-Moyennes, *N. Gast, E. Gaussier, Cours L3 math/info*
Ce document présente l'algorithme des k-moyennes et il a été utilisé pour rédiger la partie concernant les k-moyennes.

[2] - k-means++: The Advantages of Careful Seeding, *David Arthur, Sergei Vassilvitskii.*
Technical Report. Stanford. (2006)
Ce document a été utilisé pour parler de kmeans++ dans la partie sur l'algorithme des k-moyennes. Il a également été utilisé pour la partie sur les différentes méthodes d'initialisation de l'algorithme des k-moyennes.

[3] - Approximation Algorithms for Facility Location Problems, *Jens Vygen, Research Institute for Discrete Mathematics, University of Bonn. (*2004/2005*)*
Ce document décrit le problème de Weber-Fermat, et une manière pour approximer une solution à ce problème, on le cite pour permettre au lecteur de trouver plus d'information sur ce problème que ce qui a été dis dans le rapport.

[4] - A generalized Weiszfeld method for the multi-facility location problem, *Cem Iyigun, Adi Ben-Israel,* Operations Research Letters 38 (2010)
Ce document présente une méthode basée sur l'algorithme de Weiszfeld, il est intéressant pour comprendre la méthode de Weiszfeld. Il est cité pour permettre au lecteur de trouver plus d'information sur le calcul de la médiane.

[5] - An Iterative Algorithm for a Capacitated MultiFacility Weber Problem with Fuzzy Demands, *S.M.H. Manzour-al-Ajdad, S.A. Torabi and R. Tavakkoli-Moghaddam.*
E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference
Ce document présente une méthode basée sur de la programmation linéaire pour trouver la médiane d'un groupe de points. Il est cité pour permettre au lecteur de trouver plus d'information sur le calcul de la médiane.

[6] - Approximation schemes for Euclidean k-medians and related problems, *Sanjeev Arora, Prabhakar Raghavant, Satish Rae.*
In Proc. 30th Annu. ACM Sympos. Theory Comput. (1998)
Ce document présente une méthode pour trouver la médiane d'un groupe de points basé sur des arbres quaternaires. Il est cité pour permettre au lecteur de trouver plus d'information sur le calcul de la médiane.

[7] - The Effects of Ties on Convergece in K-Modes Variants for Clustering Categorical Data, *N. Orlowski, D. Schlorff, J. Blevins, D. Ca˜nas, M. T. Chu, R. E. Funderlic*
Department of Computer Science (2004)
Ce document a été utilisé pour comprendre le fonctionnement de la méthode des k-modes.

[8] - Attribute Value Weighting in K-Modes Clustering for Y-Short Tandem Repeats (Y-STR) Surname, *Ali Seman,Zainab Abu Bakar, Azizian Mohd. Sapawi*
Information Technology (ITSim), 2010 International Symposium
Ce document a été utilisé pour comprendre le fonctionnement de la méthode des k-modes.

[9]- Extending K-Means Clustering to First-Order Representations, *Mathias Kirsten and Stefan Wrobel*
Proceeding ILP '00 Proceedings of the 10th International Conference on Inductive Logic Programming (2000)

Ce document a été utilisé pour comprendre le fonctionnement de la méthode des k-modes.

[10] - Spatial Clustering Methods in Data Mining: A Survey, *Jiawei Han, Micheline Kamber and Anthony K.H.Tung* (2001)
Ce document a été utilisé pour comprendre le fonctionnement des différents algorithmes de la méthode des k-médoïdes.
[11] - Efficient and Effective Clustering Methods for Spatial Data Mining, *Raymond T.Ng, Jiawei Han*
Proceedings of the 20th VLDB Conference Santiago, Chile, 1994
Ce document introduit pour la première fois l'algorithme Clarans une variante des k-médoïdes.

[12] - Finding Groups in Data: An Introduction to Cluster Analysis, *Leonard Kaufman, Peter J. Rousseeuw (pages 68-126)*
Ce livre introduit l'algorithme PAM pour la première fois. Il a été utilisé pour se baser sur la définition officielle de l'algorithme PAM.

[13] - A simple and fast algorithm for K-medoids clustering, *Hae-Sang Park, Chi-Hyuck Jun*
Expert Systems with Applications, Vol. 36, No. 2. (09 March 2009)
Cet article présente un nouvel algorithme pour la méthode des k-médoïdes et il présente plusieurs méthodes d'initialisation. Il a été utilisé pour la partie sur l'initialisation de l'algorithme des k-moyennes ou des k-médoïdes.

[14] - Comparison of Four Initialization Techniques for the *K*-Medians Clustering Algorithm,
A. Juan and E. Vidal
Advances in Pattern Recognition (2000)
Cet article présente quatre techniques d'initialisation et il a servit pour écrire la partie sur la méthode 9 (greedy).

[15] - A systematic evaluation of different methods for initializing the K-means clustering algorithm, *Anna D. Peterson, Arka P. Ghosh and Ranjan Maitra*
(Encore en révision, 2011)
Ce document présente 11 méthodes d'initialisation et il a été étudié en vérifiant chacune de ses références quand cela était possible pour rédiger une explication de certaines méthodes d'initialisations. Quand il y avait une différence entre ce document et ses références, ce sont ses références qui ont été utilisées.

[16] – An Experimental Comparison of Several Clustering and Initialization Methods, *Marina Meila, David Heckerman*
Machine Learning (1998)
Ce document présente plusieurs techniques d'initialisation et nous nous en sommes servis pour réaliser la section 3 de la partie II.

[17] – An introduction to ROC analysis, *Tom Fawcett*
Pattern Recognition Letters 27 (2006)
Ce document présente la définition d'une courbe ROC, la manière de la calculer, et la manière de calculer l'AUC à partir d'une courbe ROC. Nous avons utilisé ce document pour rédiger l'Annexe 1.

[18] – G-means: A Clustering Algorithm for Intrusion Detection, *Zhonghua Zhao, Shanqing Guo, Qiuliang, et Tao Ban*
In ICONIP (2008)
Ce document introduit la méthode G-means, il a été cité dans le cadre du réglage de la valeur de k (Partie II section 3.2).

[19] – X-means: Extending K-means with efficient estimation of the number of clusters, *Dan Pelleg et Andrew Moore*
Proceedings of the Seventeenth International Conference on Machine Learning (2000)

Ce document introduit la méthode X-means, il a été cité dans le cadre du réglage de la valeur de k (Partie II section 3.2).

[20] – S-means: Similarity Driven Clustering and Its application in Gravitational-Wave Astronomy Data, *Hansheng Lei, Lappoon R.Tang, Juan R.Iglesias, Soma Mukherjee, et Soumya Mohanty*
Ce document introduit la méthode S-means, il a été cité dans le cadre du réglage de la valeur de k (Partie II section 3.2).

[21] – Bayesian instance selection for the nearest neighbor rule, Sylvain Ferrandiz · Marc Boullé
Mach Learn (2010)
Ce document introduit le critère de Sylvain Ferrandiz. Ce critère est utilisé dans nos tests.

[22] – Data Mining Exploration, Sélection, Compréhension, Vincent Lemaire
Habilitation à diriger des recherches de l'Université Paris Sud (2008)
Ce document est l'un des articles publiés par mon maître de stage, il a été utilisé pour l'introduction section 3.

[23] – Une nouvelle stratégie d'Apprentissage Bayésienne, Vincent Lemaire, Alexis Bondu, Marc Boullé
EGC (2010)
Ce document est l'un des articles publiés par mon maître de stage, il a été utilisé pour l'introduction section 3.

[24] – Elaboration d'une représentation basée sur un classifieur et son utilisation dans un déploiement basé sur un k-ppv, Vincent Lemaire, Marc Boullé, Pascal Gouzien
Conference d'apprentissage (CAP) (2010)
Ce document est l'un des articles publiés par mon maître de stage, il a été utilisé pour l'introduction section 3.

[25] – MODL: Une méthode quasi-optimale de discrétisation supervisée, Marc Boullé
Research Report France Telecom R&D, No 8444 (2004)
Cet article est une référence sur ce que peut permettre MODL en matière de discrétisation. Il est utilisé dans le cadre de la section 4.1 de la partie II.

[26] – MODL: Une méthode quasi-optimale de groupage des valeurs d'un attribut symbolique, Marc Boullé
Research Report France Telecom R&D, No 8611 (2004)
Cet article est une référence sur ce que peut permettre MODL en matière de groupage. Il est utilisé dans le cadre de la section 4.1 de la partie II.

[27] – Moyennage du prédicteur Bayesien Naïf Sélectif, évaluation sur un challenge international, Marc Boullé
In 8ème Conférence francophone sur l'Apprentissage automatique (Cap) (2006)
Cet article est une référence sur ce que peut permettre MODL en matière de moyennage. Il est utilisé dans le cadre de la section 4.1 de la partie II.

[28] - An Unsupervised Clustering Method using the Entropy Minimizations, *Gintautas Palubinskas, Xavier Descombes, Frithjof Kruggel.*
Pattern Recognition (1998)
C'est article décrit une méthode qui pourrait être utilisée pour régler la valeur de k (Partie II section 3.2).

[29] - Texture Analysis through a Markovian Modelling and Fuzzy Classification:
Application to Urban Area Extraction from Satellite Images, *A. LORETTE*
International Journal of Computer Vision, 36(3) (2000)
C'est article décrit une méthode qui pourrait être utilisée pour régler la valeur de k (Partie II section 3.2).