Travail encadré de recherche :

Profilage des consommateurs à l'aide des k-means++ semi-supervisés

Par :

Destin ASHUZA CIRUMANGA
Elvina GOVENDASAMY

Master 1 Mathématiques et Applications
Parcours Data Science

Sous la direction de Mme Eunice OKOME OBIANG

2019-2020

Description

La segmentation des populations est un problème rencontré dans de nombreuses disciplines. En marketing par exemple, cette technique permet aux entreprises d'identifier et de caractériser les différents profils de consommateurs, afin de leur offrir des produits et des services adaptés à leurs besoins.

Le clustering par l'algorithme des k-means est la méthode d'apprentissage non supervisé la plus utilisée pour répondre à ce type de problématique. Son objectif est de segmenter la population en k groupes disjoints sur base d'un critère de similarité. Toutefois, la qualité de ce modèle varie selon le choix des centres initiaux des clusters. L'idée des k-means++ semi supervisés est de contrôler l'initialisation des centres de clusters en intégrant des données supervisées dans le processus d'apprentissage pour ainsi améliorer les performances du clustering en termes de coût et de durée d'exécution.

Dans ce travail, nous nous intéressons dans un premier temps à la présentation de différents résultats théoriques sur les différents algorithmes des k-means et à la démonstration d'un critère de qualité associé aux k-means++ semi-supervisés. Dans un second temps, nous programmons et testons l'algorithme des k-means++ semi-supervisés sur un jeu de données clients.

Table des matières

Introduction

A l'ère du big data et de l'intelligence artificielle, le recours aux outils d'analyse et d'exploitation des données est de plus en plus fréquent, et ce dans tous les secteurs désormais. L'apprentissage automatique - machine learning en anglais - est la science au cœur de l'exploitation de ces données et de l'automatisation des méthodes dévolues à cette fin. Il existe différents types d'apprentissage, chacun ayant ses points forts et ses inconvénients. La distinction la plus courante se fait entre l'apprentissage supervisé et l'apprentissage non supervisé qui regroupent la plupart des problèmes. Et un mariage entre ces deux types donne l'apprentissage semi-supervisé.

Dans l'apprentissage supervisé, nous disposons d'un jeu de données où toutes les observations sont déjà étiquetées. Concrètement, nous avons d variables notées X mesurées sur N individus et une variable réponse notée Y qui contient l'étiquette de chaque observation. Ainsi, avec ce type d'apprentissage nous cherchons à trouver une relation entre la variable réponse (la variable que nous cherchons à expliquer) et les autres variables utilisées dans les données. L'objectif est de guider l'algorithme en se basant sur l'information disponible dans Y de sorte que les résultats prédits soient les plus proches possible de la vraie réponse. Pour comprendre ce dont il est question, pronons un exemple : supposons que notre jeu de données porte sur la clientèle d'une entreprise. Nos variables X concernent par exemple les achats de clients et la variable Y sur la catégorisation de chaque client : client riche ou client pauvre. L'apprentissage supervisé permettra de pouvoir décider à partir des achats d'un client s'il est riche ou pauvre.
Les avantages de ce type d'apprentissage sont entre autres :

- des résultats facilement interprétables : dans notre exemple, nous savons exactement à quoi correspond nos résultats (le client est soit riche, soit pauvre) ;
- de meilleures performances prédictives : nous pouvons évaluer facilement les performances. Dans notre exemple, il est facile de vérifier combien de clients ont été classés dans la bonne catégorie ;
- des modèles facilement comparables : si dans notre exemple nous avons obtenus plusieurs relations permettant de catégoriser un client sur base de ses achats, nous nous contenterons de garder celle qui classe le plus grand pourcentage de clients dans la bonne catégorie.

Cependant l'acquisition de l'information a un coût. En effet, l'information contenue dans la variable réponse nécessite une mobilisation des moyens pour son obtention. Dans notre exemple toujours, nous pouvons penser à des questionnaires d'enquête qui ont été envoyés aux clients pour avoir l'idée de leurs revenus et l'embauche du personnel supplémentaire pour traiter les informations recueillies afin d'assigner chaque client à l'une des deux catégories. De plus, dans la catégorisation des individus, il n'est pas impossible que l'humain y insère un biais pour telle ou telle autre raison. Dans le cas illustratif que nous avons donné ici exemple, nous pouvons nous demander comment seront catégorisés les clients de la classe moyenne s'il y en a. Rien ne garantit qu'il y a juste deux catégories possibles.

Dans l'apprentissage non supervisé, la variable réponse n'est plus disponible. Les données ne sont pas étiquetées et ainsi l'apprentissage ne sera pas basé sur des catégories connues préétablies. Le but sera alors d'identifier des patterns ou des relations entre les variables, de regrouper les individus présentant des similitudes (clustering) ou encore de mettre en évidence les variables les plus significatives pour caractériser les individus. L'in-

convénient majeur de ce type d'apprentissage est la difficulté à interpréter les modèles ou à évaluer leurs performances. Mais il a comme avantages une facilité de mise en œuvre, des coûts réduits pour l'acquisition de l'information et une possible correction du biais qui serait dû aux a priori sur la catégorisation des individus.
En reprenant l'exemple sur les clients donné ci-haut, l'apprentissage non supervisé donnera un regroupement de clients en différentes catégories (donc pas nécessairement deux catégories). Cependant, nous serons incapables de dire directement à quoi correspond chaque catégorie ou de pouvoir juger de la qualité de cette catégorisation.

Pour ce qui est de l'apprentissage semi-supervisé, il est obtenu comme un jumelage de deux types précédents : l'on dispose de l'information Y juste pour une partie des données. Dès lors, ses avantages sont un mélange entre les avantages du supervisé et du non supervisé : la correction du biais sur les catégories, des performances significatives par rapport au non supervisé et augmentant avec la quantité des données supervisées, l'acquisition de l'information moins coûteuse par rapport au supervisé. Toutefois, dans certains cas, l'interprétation du modèle pose toujours problème.

Notre étude porte non pas sur tous ces types d'apprentissage mais juste sur l'algorithme des k-means qui est un algorithme simple et très utilisé relevant de l'apprentissage non supervisé. Il est par exemple utilisé en marketing pour répondre à la problématique de la segmentation des consommateurs. Il permet de pouvoir séparer les individus en k groupes homogènes appelés clusters. Les performances du modèle dépendent du choix des centres initiaux de ces clusters. L'objectif de notre étude est d'évaluer la performance des k-means++ semi-supervisés qui utilisent des données supervisées pour contrôler l'initialisation des centres.
Ainsi, dans une première partie théorique, nous présentons l'algorithme des k-means et son principe de fonctionnement ; ensuite nous présentons sa version améliorée, les k-means++, en démontrant un critère de qualité qui lui est associé ; et enfin nous étendons cette dernière version au cas où l'on dispose de certaines données qui sont labellisées, ce qui donne l'algorithme des k-means++ semi-supervisés.
Dans une deuxième partie appliquée, nous implémentons chacun des trois algorithmes étudiés grâce au langage python et nous comparons leurs performances en termes de coût et de temps d'exécution en les testant sur une base réelle des données clients. Nous appliquons enfin les résultats des k-means++ semi-supervisés à la segmentation des consommateurs de cette base.

I Partie théorique

Cette partie est consacrée à la présentation et à l'étude théorique de différents algorithmes de k-means. Les résultats énoncés et démontrés sont essentiellement tirés de [2] et de [6]. Ils sont suffisamment détaillés pour être facilement accessibles à toute personne ayant des bases en probabilités et en statistique.

I.1 Les k-means standards

Nous désignons par « k-means standards » la version de base de la méthode des k-means. C'est une méthode de classification non supervisée utilisée lorsque les données

ne sont pas étiquetées. Elle est basée sur une approche itérative qui permet de former k groupes disjoints en regroupant les observations présentant des similitudes. Pour quantifier cette similitude entre observations, on calcule la distance entre chaque observation et le centre de son groupe puis on fait la somme de toutes les distances ainsi calculées. L'objectif est alors de trouver un ensemble de k centres qui minimise cette somme des distances.

Présentation de l'algorithme

Soient $d \geq 1$ le nombre de variables mesurées sur nos observations, $X \subset \mathbb{R}^d$ l'ensemble de ces observations et k le nombre de centres voulus.
L'algorithme des k-means est utilisé pour résoudre le problème suivant : étant donné l' entier naturel k fixé et l'ensemble de points $X \subset \mathbb{R}^d$ (avec $d \geq 1$), trouver un ensemble de centres $C = \{c_i \in \mathbb{R}^d, i = 1, 2, ..., k\}$ tel que

$$C = \underset{\{A \subset \mathbb{R}^d, card(A)=k\}}{\operatorname{argmin}} \sum_{x \in X} \min_{c \in A} \|x - c\|^2 \tag{1}$$

Par la suite, en utilisant ces centres, le label de chaque élément x est donné par :

$$\ell(x) = \underset{1 \leq i \leq k}{\operatorname{argmin}} \|x - c_i\|$$

Résoudre l'équation (1) de manière exacte pour trouver C est "NP-hard" ; cela implique qu'il est difficile de trouver une solution exacte avec un algorithme de complexité polynomiale. Ainsi, en pratique, l'on se contente d'approcher la solution de manière locale. L'algorithme de Lloyd est le plus utilisé pour cette fin. Il prend en entrée l'ensemble des données X, un ensemble de centres initiaux C et retourne les centres finaux résolvant localement l'équation (1). Il a été proposé par Stuart Lloyd en 1957 à partir des idées de Hugo Steinhaus en 1956 et a été utilisé pour la première fois par James MacQueen en 1967. Cet algorithme fonctionne de façon itérative sur les deux tâches suivantes :

1. Assigner chaque observation au centre le plus proche en calculant sa distance à tous les centres.
2. Mettre à jour les centres en remplaçant chaque centre par la moyenne des observations qui lui sont assignées.

Le processus est répété jusqu'au moment où les centres ne changeront plus. Concrètement, cet algorithme s'implémente comme suit :

Algorithme de Lloyd
Entrées : X (le dataset) et C (l'ensemble des k centres initiaux)
Sorties : C (l'ensemble des centres définitifs)
1 faire
2 assigner chaque $x_i \in X$ au centre le plus proche $c(x_i) \in C$ ($c(x_i) = c_{\ell(x_i)}$)
3 remplacer chaque $c_j \in C$ par le point moyen des $x_i \in X$ tels que $c(x_i) = c_j$ ($\ell(x_i) = j$)
4 Tant que C change
5 sortir C

L'algorithme des k-means standards est alors obtenu en exécutant l'algorithme de Lloyd avec les centres initiaux qui sont choisis de manière uniformément aléatoire parmi les individus de notre base de données X.

Pour finir, signalons que le choix du nombre de centres (ne faisant pas parti du problème des k-means) se fait en utilisant diverses techniques dont la méthode du coude qui sera présentée plus tard dans ce travail.

Convergence

Après avoir compris le fonctionnement de l'algorithme de Lloyd (et donc celui des k-means standards qui en découle), une question légitime se pose : l'algorithme converge-t-il ?
Avant de répondre à cette question, introduisons d'abord quelques définitions et notations qui nous seront utiles dans la suite de ce travail.

Définition 1. ***(Clustering et cluster)***
Soit $1 \leq m \leq k$ un entier (k désignant le nombre de centres recherchés dans le problème des k-means).

Un clustering $C = \{c_1, c_2, ..., c_m\}$ est un ensemble de m centres qui sont utilisés pour déterminer le label de chaque point du jeu de données X.

Pour tout $1 \leq i \leq m$, un cluster X_i désigne alors l'ensemble des points ayant le label i, c'est-à-dire

$$X_i = \left\{ x \in X, i = \ell(x) = \underset{1 \leq j \leq m}{\operatorname{argmin}} \|x - c_j\| \right\}$$

Par la suite, par abus de langage, nous parlerons d'un cluster X_i dans C pour signifier que c'est un cluster défini à partir du clustering C. Cela ne signifie en aucun cas qu'un cluster est un centre.

Définition 2. ***(Fonction potentielle)***
Étant donné un clustering C , la fonction potentielle ϕ associée à C est la fonction à valeurs dans $\mathbb{R}^+$ définie dans l'ensemble des parties de X par :

$$\forall A \subset X\,, \quad \phi(A) = \sum_{a \in A} \min_{c \in C} \|a - c\|^2$$

Par la suite, par abus de notation, nous écrirons tout simplement $\phi(X) = \phi$ et uniquement pour X.

Définition 3. ***(Clustering optimal et cluster optimal)***
Soit C_{OPT} le clustering qui serait la solution exacte du problème des k-means tel que formulé par l'équation (1). C_{OPT} sera alors appelé clustering optimal et tout cluster dans C_{OPT} sera appelé cluster optimal.

Définition 4. ***(Poids D^2)***
Étant donné un clustering C, le poids D^2 est défini par :

$$\forall x \in X\,, \quad D^2(x) = \phi(\{x\}) = \min_{c \in C} \|x - c\|^2$$

C'est donc le carré de la distance de x au centre dans C qui lui est le plus proche.

A la lumière de ces définitions, nous pouvons résumer le problème de k-means comme suit :

— Le problème de k-means tel que formulé par l'équation (1) consiste à trouver le clustering qui minimise la fonction potentielle ϕ
— Théoriquement, on sait qu'il existe une solution exacte C_{OPT} à ce problème mais en pratique il est difficile de la calculer.
— L'algorithme des k-means fait donc appel à l'algorithme de Llyod pour trouver un minimum local de ϕ atteint en une solution approchée donnée par un clustering C.

Nous pouvons maintenant énoncer le résultat qui nous permet d'affirmer que l'algorithme converge :

Lemme 1. *Soient S un ensemble de points, $c(S)$ son isobarycentre et z un point quelconque.*
Alors, on a :

$$\sum_{x\in S} \|x-z\|^2 - \sum_{x\in S} \|x-c(S)\|^2 = card(S)\cdot\|c(S)-z\|^2$$

où $card(S)$ désigne le cardinal de S.

Démonstration.
Nous avons :

$$\begin{aligned}
\sum_{x\in S} \|x-z\|^2 &= \sum_{x\in S} \|\,(x-c(S)) + (c(S)-z)\,\|^2 \\
&= \sum_{x\in S} \|x-c(S)\|^2 + 2\cdot\sum_{x\in S}\langle x-c(S),\ c(S)-z\rangle + \sum_{x\in S}\|c(S)-z\|^2 \\
&= \sum_{x\in S} \|x-c(S)\|^2 + 2\cdot\langle\sum_{x\in S} x - card(S)\cdot c(S),\ c(S)-z\rangle + card(S)\cdot\|c(S)-z\|^2 \\
&= \sum_{x\in S} \|x-c(S)\|^2 + 2\cdot card(S)\cdot\langle c(S)-c(S),\ c(S)-z\rangle + card(S)\cdot\|c(S)-z\|^2 \\
&= \sum_{x\in S} \|x-c(S)\|^2 + 2\cdot card(S)\cdot\langle 0,\ c(S)-z\rangle + card(S)\cdot\|c(S)-z\|^2 \\
. &= \sum_{x\in S} \|x-c(S)\|^2 + card(S)\cdot\|c(S)-z\|^2
\end{aligned}$$

d'où le résultat. □

Le lemme 1 nous permet d'affirmer que remplacer le centre de chaque cluster par le point moyen de ses éléments à chaque tour de boucle de l'algorithme de Llyod permet de décroître nécessairement ϕ. Ainsi dès que ϕ atteint un minimum local, l'algorithme s'arrête.

Savoir que l'algorithme converge est une bonne chose. Malheureusement rien ne garantit que la solution locale obtenue est un bon clustering. En effet, ϕ peut être arbitrairement grand malgré l'optimum local atteint. Ainsi, même si l'algorithme des k-means est exécuté plusieurs fois sur un même jeu des données, il arrive qu'il donne de très mauvais résultats

et les éléments ne sont donc pas bien classés. Il n'y a donc pas de garantie de précision. C'est le défaut principal de l'algorithme des k-means standards. Dans la section suivante, nous présentons les k-means++ qui résolvent ce problème.

I.2 Les k-means++

Présentation de l'algorithme

L'initialisation de l'algorithme des k-means est un élément clef de la détermination d'un optimum local de bonne qualité. Dans cette section, nous cherchons à étudier une autre forme d'initialisation qui permettrait de résoudre le problème évoqué précédemment lors de l'application des k-means standards.

L'algorithme des k-means++ est obtenu en changeant uniquement la manière dont sont initialisés les centres qui seront utilisés dans l'algorithme de Lloyd. Au lieu de les choisir uniformément, ces centres initiaux sont choisis avec une probabilité proportionnelle au poids D^2. Ensuite le reste se passe exactement comme pour les k-means standards en faisant appel à l'algorithme de Lloyd.

Ainsi, le fonctionnement de l'algorithme des k-means++ se résume de la manière suivante :

1. Nous choisissons aléatoirement et de façon uniforme une observation de notre dataset X. Nous l'ajoutons à C comme premier centre.
2. Pour chaque individu x du dataset, nous calculons sa distance euclidienne au carré par rapport à chaque centre déjà trouvé et nous définissons son poids $D^2(x)$ comme étant le minimum des distances calculées ; mathématiquement ça donne :

$$\forall x \in X, D^2(x) = \min_{c \in C} \|x - c\|^2$$

3. Choisir un individu du dataset X avec une probabilité proportionnelle à la distance calculée à l'étape 2. Pour un individu x_0, cette probabilité correspond à $\frac{D^2(x_0)}{\sum_{x \in X} D^2(x)}$. L'individu choisi est alors ajouté à l'ensemble C de centres

Les étapes 2 et 3 sont répétées jusqu'à ce que les k centres initiaux aient été choisis. Une fois que ces centres sont obtenus, nous utilisons l'algorithme de Lloyd afin de déterminer les centres et clusters finaux.

Voici donc le pseudo-code correspondant à cet algorithme :

Algorithme d'initialisation des centres pour les k-means++
Entrées : X (le dataset) et k (le nombre de centres)
Sorties : C (l'ensemble des centres initiaux)
1 choisir un $x \in X$ suivant la loi uniforme
2 initialiser $C = \{x\}$
3 Tant que $card(C) \leq k$ **faire**
4 choisir un $x \in X$ avec probabilité proportionnelle à $D^2(x)$
5 actualiser $C = C \cup \{x\}$
6 Fin Tant que

7 sortir C

Cette nouvelle initialisation permet de résoudre le problème évoqué pour les k-means. Concrètement, elle permet d'avoir une idée de la précision de l'algorithme en fournissant une majoration de la valeur moyenne de la fonction potentielle. C'est à dire qu'en roulant plusieurs fois l'algorithme des k-means++, contrairement aux k-means standards, la moyenne de toutes les fonctions potentielles obtenues pour les différentes tentatives ne peut pas excéder une certaine valeur définie par rapport au clustering optimal ; ce qui implique une certaine garantie en termes de précision. Ce résultat est l'objet de la section suivante.

Critère de qualité

Dans cette section, nous présentons le résultat qui montre que l'algorithme des k-means++ est meilleur que celui des k-means standards dans le sens où l'espérance de la fonction potentielle est majorée. Plus exactement, le théorème 1 présenté à la fin de cette section montre que la fonction potentielle obtenue à fin de l'étape de l'initialisation de l'algorithme des k-means++ est un $O(\ln k)$ par rapport au clustering optimal théorique. Et cela est suffisant car dans l'algorithme de Lloyd, cette fonction ne peut faire que baisser. Voici l'énoncé précis du théorème en question :

Théorème 1 : Soient C' un clustering obtenu lors de l'initialisation de l'algorithme des k-means++ et ϕ' sa fonction potentielle associée. Alors on a :

$$\mathbb{E}[\phi'] \leq 8 \cdot (\ln k + 2) \cdot \phi_{OPT}$$

Nous présentons trois lemmes qui permettent de démontrer ce théorème : le premier permet de contrôler la fonction potentielle pour le choix du premier centre ; le deuxième montre que cette fonction reste toujours majorée lorsque l'on choisit les centres restants dans les clusters optimaux théoriques proportionnellement au poids D^2. Enfin le troisième lemme généralise cela pour n'importe quel choix des centres, ce qui conduit à la démonstration du théorème.

Lemme 2. *Soient A un cluster arbitraire dans C_{OPT} et $c(A)$ son point moyen. Soit C un clustering avec un unique centre choisi aléatoirement dans A et de fonction potentielle ϕ.*
Alors, on a :

$$\mathbb{E}\left[\phi(A)\right] = 2 \cdot \phi_{OPT}\left(A\right)$$

Démonstration.
Supposons que $C = \{a_0\}$ avec a_0 qui a été choisi aléatoirement dans A ; c'est-à-dire :

$$\mathbb{P}\left(\{choisir\ a_0\}\right) = \frac{1}{card(A)}$$

En faisant varier le choix de a_0, nous voyons clairement que $\phi(A)$ est une variable aléatoire discrète à valeurs dans $\mathbb{R}$. En effet, $card\left(\phi(A)\left(\Omega\right)\right) = card(A)$ et pour chaque

choix de a_0, la valeur prise par $\phi(A)$ est donnée par :

$$\begin{aligned}\phi_{a_0}(A) &= \sum_{a\in A}\min_{c\in\{a_0\}}\|a-c\|^2 \\ &= \sum_{a\in A}\|a-a_0\|^2\end{aligned}$$

De plus, par le lemme 1, nous savons que l'isobarycentre du cluster A, $c(A)$, est nécessairement le centre utilisé pour ce cluster dans le clustering optimal C_{OPT} sinon ϕ_{OPT} ne serait pas minimale. De tout ce qui précède, nous pouvons alors écrire :

$$\begin{aligned}\mathbb{E}\left[\phi(A)\right] &= \sum_{a_0\in A}\phi_{a_0}(A)\cdot\mathbb{P}\left(\phi(A)=\phi_{a_0}(A)\right) \\ &= \sum_{a_0\in A}\sum_{a\in A}\|a-a_0\|^2\cdot\frac{1}{card(A)} \\ &= \frac{1}{card(A)}\cdot\sum_{a_0\in A}\sum_{a\in A}\|a-a_0\|^2\end{aligned}$$

En appliquant le lemme 1 avec $S=A$ et $z=a_0$, nous obtenons :

$$\begin{aligned}\mathbb{E}\left[\phi(A)\right] &= \frac{1}{card(A)}\sum_{a_0\in A}\left(card(A)\cdot\|c(A)-a_0\|^2+\sum_{a\in A}\|a-c(A)\|^2\right) \\ &= \sum_{a_0\in A}\|c(A)-a_0\|^2+\frac{1}{card(A)}\cdot\sum_{a_0\in A}\sum_{a\in A}\|a-c(A)\|^2 \\ &= \sum_{a_0\in A}\|c(A)-a_0\|^2+\frac{1}{card(A)}\cdot card(A)\cdot\sum_{a\in A}\|a-c(A)\|^2 \\ &= 2\text{`}\cdot\sum_{a\in A}\|c(A)-a\|^2 \\ &= 2\cdot\phi_{OPT}(A)\end{aligned}$$

ce qui conclut la preuve. □

Lemme 3. *Soient A un cluster arbitraire dans C_{OPT} et C un clustering quelconque. Supposons que l'on ajoute aléatoirement un centre à C choisi dans A avec une probabilité proportionnelle à D^2, la distance euclidienne au carré. Soit ϕ la fonction potentielle du clustering ainsi obtenu.*
Alors, on a :

$$\mathbb{E}\left[\phi(A)\right]\leq 8\cdot\phi_{OPT}\left(A\right)$$

Démonstration.
Soit $1\leq l<k$ (ici k représente le nombre de centres utilisés pour le clustering optimal C_{OPT}).
Supposons que $C=\{c_1,c_2,...,c_l\}$ et considérons un cluster A dans C_{OPT}.
Rappelons que la distance D^2 associée au clustering C vérifie :

$$\forall a\in A, D^2(a)=\min_{c\in C}\|a-c\|^2$$

Soit $a_0 \in A$ choisi avec probabilité $\dfrac{D^2(a_0)}{\sum_{a \in A} D^2(a)}$

Notons $C(a_0) = C \cup \{a_0\}$ et ϕ_{a_0} respectivement le clustering et la fonction potentielle obtenus pour ce choix de a_0. Nous pouvons alors écrire :

$$\begin{aligned}
\mathbb{E}\left[\phi(A)\right] &= \sum_{a_0 \in A} \phi_{a_0}(A) \cdot \mathbb{P}\left(\phi(A) = \phi_{a_0}(A)\right) \\
&= \sum_{a_0 \in A} \sum_{a \in A} \min_{c \in C(a_0)} \|a - c\|^2 \cdot \frac{D^2(a_0)}{\sum_{a \in A} D^2(a)} \\
&= \sum_{a_0 \in A} \sum_{a \in A} \min\left(\min_{c \in C} \|a - c\|^2, \|a - a_0\|^2\right) \cdot \frac{D^2(a_0)}{\sum_{a \in A} D^2(a)} \\
&= \sum_{a_0 \in A} \sum_{a \in A} \min\left(D^2(a), \|a - a_0\|^2\right) \cdot \frac{D^2(a_0)}{\sum_{a \in A} D^2(a)}
\end{aligned}$$

De plus pour tout $a \in A$, nous avons :

$$\begin{aligned}
D(a_0) &= \min_{c \in C} \|a_0 - c\| \\
&\leq \|a_0 - c\| = \|(a_0 - a) + (a - c)\| \qquad \forall c \in C \\
&\leq \|a_0 - a\| + \|a - c\| \qquad \forall c \in C \quad \text{(par inégalité triangulaire)}
\end{aligned}$$

En particulier pour $c_j \in C$ tel que $D(a) = \|a - c_j\|$, nous avons :

$$D(a_0) \leq \|a_0 - a\| + D(a)$$

D'une part, en élevant au carré les membres de cette inégalité, nous obtenons :

$$D^2(a_0) \leq \|a_0 - a\|^2 + D^2(a) + 2 \cdot \|a_0 - a\| \cdot D(a)$$

D'autre part,

$$\begin{aligned}
\left(\|a_0 - a\| - D(a)\right)^2 &\geq 0 \\
\Leftrightarrow \|a_0 - a\|^2 + D^2(a) &\geq 2 \cdot \|a_0 - a\| \cdot D(a)
\end{aligned}$$

Ainsi, nous en déduisons que

$$\forall a \in A, \quad D^2(a_0) \leq 2 \cdot \|a_0 - a\|^2 + 2 \cdot D^2(a)$$

En sommant sur toutes les valeurs de a, nous obtenons alors

$$D^2(a_0) \leq \frac{2}{card(A)} \cdot \sum_{a \in A} \|a_0 - a\|^2 + \frac{2}{card(A)} \cdot \sum_{a \in A} D^2(a)$$

Nous utilisons ensuite cette inégalité pour majorer $D^2(a_0)$ dans l'expression de $\mathbb{E}\left[\phi(A)\right]$ obtenue précédemment, ce qui conduit à

$$\begin{aligned}
\mathbb{E}\left[\phi(A)\right] &\leq \frac{2}{card(A)} \cdot \sum_{a_0 \in A} \sum_{a \in A} \min\left(D^2(a), \|a - a_0\|^2\right) \cdot \frac{\sum_{a \in A} \|a_0 - a\|^2}{\sum_{a \in A} D^2(a)} \\
&+ \frac{2}{card(A)} \cdot \sum_{a_0 \in A} \sum_{a \in A} \min\left(D^2(a), \|a - a_0\|^2\right) \cdot \frac{\sum_{a \in A} D^2(a)}{\sum_{a \in A} D^2(a)}
\end{aligned}$$

Dans le second membre de cette inégalité, nous utilisons le fait que

$$\min\left(D^2(a), \|a-a_0\|^2\right) \leq D^2(a)$$

pour le premier bloc, et

$$\min\left(D^2(a), \|a-a_0\|^2\right) \leq \|a-a_0\|^2$$

pour le second bloc, nous obtenons :

$$\begin{aligned}\mathbb{E}\left[\phi(A)\right] &\leq \frac{2}{card(A)} \cdot \sum_{a_0 \in A} \sum_{a \in A} \|a_0 - a\|^2 \\ &+ \frac{2}{card(A)} \cdot \sum_{a_0 \in A} \sum_{a \in A} \|a - a_0\|^2\end{aligned}$$

C'est-à-dire

$$\mathbb{E}\left[\phi(A)\right] \leq 4 \cdot \frac{1}{card(A)} \cdot \sum_{a_0 \in A} \sum_{a \in A} \|a_0 - a\|^2$$

L'expression $\frac{1}{card(A)} \cdot \sum_{a_0 \in A} \sum_{a \in A} \|a_0 - a\|^2$ est celle que nous avons obtenue pour l'espérance lors de la démonstration du lemme 2. Ainsi, en appliquant le lemme 2, nous obtenons donc

$$\mathbb{E}\left[\phi(A)\right] \leq 8 \cdot \phi_{OPT}\left(A\right)$$

ce qui est le résultat escompté. □

Lemme 4. *Soient C un clustering arbitraire de fonction potentielle ϕ et $u > 0$ un entier naturel. Choisissons u clusters non couverts par C dans C_{OPT}. Notons X_u l'ensemble des points de ces clusters et $X_c = X - X_u$ son complémentaire. Supposons que nous ajoutons $t \leq u$ centres à C choisis de manière aléatoire proportionnellement au poids D^2. Notons C' le nouveau clustering ainsi obtenu et ϕ' sa fonction potentielle associée. Alors, on a :*

$$\mathbb{E}\left[\phi'\right] \leq \left(\phi\left(X_c\right) + 8 \cdot \phi_{OPT}\left(X_u\right)\right) \cdot \left(1 + H_t\right) + \frac{u-t}{u} \cdot \phi\left(X_u\right);$$

avec $H_t = \sum_{i=1}^{t} \frac{1}{i}$ la $t^{ème}$ somme partielle de la série harmonique.

Démonstration. Elle se fait par une récurrence à deux niveaux en montrant que si l'inégalité est vérifiée pour les couples d'entiers $(t-1, u)$ et $(t-1, u-1)$, alors elle l'est également pour le couple (t, u).

Regardons d'abord de plus près la fonction ϕ' et sa relation avec ϕ :
Notons c'_1, c'_2, ..., c'_t les t centres ajoutés à C. Pour toute partie $A \subset X$, on a alors :

$$\begin{aligned}\phi'\left(A\right) &= \sum_{x \in A} \min_{c' \in C'} \|x - c'\|^2 \\ &= \sum_{x \in A} \min\left(D^2\left(x\right), \|x - c'_1\|^2, ..., \|x - c'_t\|^2\right) \\ &\leq \sum_{x \in A} D^2\left(x\right) = \phi\left(A\right).\end{aligned}$$

Passons à la démonstration proprement dite maintenant :

- Initialisation : il suffit de vérifier pour $t = 0$, $u > 0$ et $t = 1$,$u = 1$.

Pour $t = 0$, $u > 0$ on a :

$$\begin{aligned}
\mathbb{E}[\phi'] &= \phi \qquad \text{(car } \phi' = \phi \text{ qui est constante)} \\
&= \phi(X_c) + \phi(X_u) \\
&= \left(\phi(X_c)\right) \cdot (1 + H_0) + \frac{u-0}{u} \cdot \phi(X_u) \quad \text{avec } H_0 = 0 \\
&\leq \left(\phi(X_c) + 8 \cdot \phi_{OPT}(X_u)\right) \cdot (1 + H_0) + \frac{u-0}{u} \cdot \phi(X_u)
\end{aligned}$$

Pour $t = 1$, $u = 1$ on distingue deux cas : soit le nouveau centre est choisi dans le seul cluster non couvert, soit il est choisi dans l'un des clusters couverts.
Si le nouveau centre c'_1 est choisi dans le cluster non couvert, il l'est avec probabilité $\frac{\phi(X_u)}{\phi}$ et on a :

$$\begin{aligned}
\phi' &= \phi'(X_c) + \phi'(X_u) \\
&\leq \phi(X_c) + \phi'(X_u)
\end{aligned}$$

D'où,

$$\begin{aligned}
\mathbb{E}[\phi' \mid c'_1 \in X_u] &\leq \phi(X_c) + \mathbb{E}[\phi'(X_u) \mid c'_1 \in X_u] \\
&\leq \phi(X_c) + 8 \cdot \phi_{OPT}(X_u) \quad \text{(par le lemme 3).}
\end{aligned}$$

Sinon, le nouveau centre c'_1 est choisi dans X_c avec probabilité $\frac{\phi(X_c)}{\phi}$ et on a :

$$\mathbb{E}[\phi' \mid c'_1 \in X_c] \leq \phi \quad \text{(car } \phi' \leq \phi \text{).}$$

Finalement,

$$\begin{aligned}
\mathbb{E}[\phi'] &= \mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in X_u\}}\right] + \mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in X_c\}}\right] \\
&= \mathbb{P}(c'_1 \in X_u)\, \mathbb{E}[\phi' \mid c'_1 \in X_u] + \mathbb{P}(c'_1 \in X_c)\, \mathbb{E}[\phi' \mid c'_1 \in X_c] \\
&\leq \frac{\phi(X_u)}{\phi}\left(\phi(X_c) + 8\phi_{OPT}(X_u)\right) + \frac{\phi(X_c)}{\phi}\, \phi \\
&\leq 2 \cdot \phi(X_c) + 8 \cdot \phi_{OPT}(X_u) \\
&\leq \left(\phi(X_c) + 8 \cdot \phi_{OPT}(X_u)\right) \cdot (1 + H_1) + \frac{1-1}{1} \cdot \phi(X_u)
\end{aligned}$$

- Hypothèse de récurrence (HR) et hérédité : soient $t \geq 1$ et $u \geq 1$.
Supposons l'inégalité vérifiée pour les couples $(t-1, u)$ et $(t-1, u-1)$ et montrons qu'elle est vraie pour le couple (t, u) également :
Comme précédemment, on considère deux cas suivant le choix du premier centre c'_1.

$\underline{1^{er}}$ **cas :** choix du premier centre dans un cluster inclus dans X_c.
On a :

$$\begin{aligned}\mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in X_c\}}\right] &= \mathbb{P}\left(c'_1 \in X_c\right) \mathbb{E}\left[\phi' \mid c'_1 \in X_c\right] \\ &= \frac{\phi\left(X_c\right)}{\phi} \mathbb{E}\left[\phi' \mid c'_1 \in X_c\right]\end{aligned}$$

Notons ϕ'' la fonction potentielle associée au clustering $C \cup \{c'_1\}$:

$$\phi'' = \sum_{x \in X} \min\left(D^2(x), \|x - c'_1\|^2\right) \leq \phi$$

En appliquant l'hypothèse de récurrence avec le couple $(t-1, u)$ car il reste $t-1$ centres à choisir et u clusters toujours non couverts, on obtient :

$$\begin{aligned}\mathbb{E}\left[\phi' \mid c'_1 \in X_c\right] &\leq \left(\phi''\left(X_c\right) + 8 \cdot \phi_{OPT}\left(X_u\right)\right) \cdot \left(1 + H_{t-1}\right) + \frac{u-t+1}{u} \cdot \phi''\left(X_u\right) \\ &\leq \left(\phi\left(X_c\right) + 8 \cdot \phi_{OPT}\left(X_u\right)\right) \cdot \left(1 + H_{t-1}\right) + \frac{u-t+1}{u} \cdot \phi\left(X_u\right)\end{aligned}$$

Ainsi,

$$\mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in X_c\}}\right] \leq \frac{\phi\left(X_c\right)}{\phi} \left[\left(\phi\left(X_c\right) + 8 \cdot \phi_{OPT}\left(X_u\right)\right) \cdot \left(1 + H_{t-1}\right) + \frac{u-t+1}{u} \cdot \phi\left(X_u\right)\right]$$

$\underline{2^{ème}}$ **cas :** choix du premier centre dans un cluster $A \subset X_u$.
On a :

$$\begin{aligned}\mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in A\}}\right] &= \mathbb{P}\left(c'_1 \in A\right) \mathbb{E}\left[\phi' \mid c'_1 \in A\right] \\ &= \frac{\phi\left(A\right)}{\phi} \mathbb{E}\left[\phi' \mid c'_1 \in A\right]\end{aligned}$$

Notons p_a la probabilité de choisir l'élément $a \in A$ comme premier centre et $\phi_a = \phi''(A)$ où ϕ'' désigne la fonction potentielle associée au clustering $C \cup \{c'_1\}$ comme précédemment mais avec $c'_1 = a$ ici. Alors, on a :

$$\begin{aligned}\mathbb{E}\left[\phi' \mid c'_1 \in A\right] &= \sum_{a \in A} \mathbb{P}\left(c'_1 = a\right) \mathbb{E}\left[\phi' \mid c'_1 = a\right] \\ &= \sum_{a \in A} p_a \, \mathbb{E}\left[\phi' \mid c'_1 = a\right]\end{aligned}$$

En appliquant l'hypothèse de récurrence avec le couple $(t-1, u-1)$ car il reste $t-1$ centres à choisir et $u-1$ clusters non couverts, on obtient :

$$\mathbb{E}\left[\phi' \,|\, c'_1 \in A\right] \leq \sum_{a \in A} p_a \left[\left(\phi''\left(X_c \cup A\right) + 8\phi_{OPT}\left(X_u - A\right)\right) \cdot \left(1 + H_{t-1}\right) + \frac{u-t}{u-1} \cdot \phi''\left(X_u - A\right)\right]$$

Comme

$$\begin{aligned}\phi''\left(X_c \cup A\right) &= \phi''\left(X_c\right) + \phi''\left(A\right) \\ &\leq \phi\left(X_c\right) + \phi_a \ ,\end{aligned}$$

Et

$$\phi''(X_u - A) + \phi(A) \leq \phi(X_u - A) + \phi(A) = \phi(X_u)$$

$$\Rightarrow \quad \phi''(X_u - A) \leq \phi(X_u) - \phi(A) \ ,$$

On aboutit à :

$$\begin{aligned}
\mathbb{E}[\phi' \mid c'_1 \in A] &\leq \sum_{a \in A} p_a \left[\left(\phi(X_c) + \phi_a + 8 \cdot \phi_{OPT}(X_u) - 8 \cdot \phi_{OPT}(A) \right) \cdot (1 + H_{t-1}) \right. \\
&\qquad \left. + \frac{u-t}{u-1} \cdot \left(\phi(X_u) - \phi(A) \right) \right] \\
&\leq \left(\phi(X_c) + \sum_{a \in A} p_a \phi_a + 8 \cdot \phi_{OPT}(X_u) - 8 \cdot \phi_{OPT}(A) \right) \cdot (1 + H_{t-1}) \\
&\qquad + \frac{u-t}{u-1} \cdot \left(\phi(X_u) - \phi(A) \right)
\end{aligned}$$

Par le lemme 3, on sait que :

$$\sum_{a \in A} p_a \phi_a \leq 8 \cdot \phi_{OPT}(A)$$

En appliquant ce résultat, on obtient :

$$\mathbb{E}[\phi' \mid c'_1 \in A] \leq \left(\phi(X_c) + 8 \cdot \phi_{OPT}(X_u) \right) \cdot (1 + H_{t-1}) + \frac{u-t}{u-1} \cdot \left(\phi(X_u) - \phi(A) \right).$$

D'où,

$$\begin{aligned}
\mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in A\}}\right] &= \frac{\phi(A)}{\phi} \mathbb{E}[\phi' \mid c'_1 \in A] \\
&\leq \frac{\phi(A)}{\phi} \left[\left(\phi(X_c) + 8 \cdot \phi_{OPT}(X_u) \right) \cdot (1 + H_{t-1}) + \frac{u-t}{u-1} \cdot \left(\phi(X_u) - \phi(A) \right) \right]
\end{aligned}$$

Comme

$$\mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in X_u\}}\right] = \sum_{A \subset X_u} \mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in A\}}\right] \ ,$$

on obtient :

$$\mathbb{E}\left[\phi' \mathbb{1}_{\{c'_1 \in X_u\}}\right] \leq \frac{\phi(X_u)}{\phi} \left(\phi(X_c) + 8 \cdot \phi_{OPT}(X_u) \right) \cdot (1 + H_{t-1}) + \frac{1}{\phi} \frac{u-t}{u-1} \cdot \left(\phi(X_u)^2 - \sum_{A \subset X_u} \phi(A)^2 \right)$$

Or, par l'inégalité des moyennes [1] (moyenne arithmétique et moyenne quadratique), on a :

$$\sum_{A \subset X_u} \phi(A)^2 \geq \frac{1}{u} \left(\sum_{A \subset X_u} \phi(A) \right)^2 = \frac{1}{u} \phi(X_u)^2$$

1. L'énoncé et la démonstration de cette inégalité sont présentés en annexe 2 à la page 29

$$\Rightarrow \quad -\sum_{A\subset X_u}\phi\left(A\right)^2 \leq -\frac{1}{u}\phi\left(X_u\right)^2$$

Ainsi,

$$\begin{aligned}\mathbb{E}\left[\phi'\mathbb{1}_{\{c_1'\in X_u\}}\right] &\leq \frac{\phi\left(X_u\right)}{\phi}\left(\phi\left(X_c\right)+8\phi_{OPT}\left(X_u\right)\right)\left(1+H_{t-1}\right)+\frac{1}{\phi}\frac{u-t}{u-1}\left(\phi\left(X_u\right)^2-\frac{1}{u}\phi\left(X_u\right)^2\right)\\ &\leq \frac{\phi\left(X_u\right)}{\phi}\left[\left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\cdot\left(1+H_{t-1}\right)+\frac{u-t}{u}\cdot\phi\left(X_u\right)\right]\end{aligned}$$

Finalement, en regroupant les deux cas, on obtient :

$$\begin{aligned}\mathbb{E}\left[\phi'\right] &= \mathbb{E}\left[\phi'\mathbb{1}_{\{c_1'\in X_c\}}\right]+\mathbb{E}\left[\phi'\mathbb{1}_{\{c_1'\in X_u\}}\right]\\ &\leq \frac{\phi\left(X_c\right)}{\phi}\left[\left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\cdot\left(1+H_{t-1}\right)+\frac{u-t+1}{u}\cdot\phi\left(X_u\right)\right]\\ &\quad+\frac{\phi\left(X_u\right)}{\phi}\left[\left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\cdot\left(1+H_{t-1}\right)+\frac{u-t}{u}\cdot\phi\left(X_u\right)\right]\\ &\leq \left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\cdot\left(1+H_{t-1}\right)+\frac{u-t}{u}\cdot\phi\left(X_u\right)\\ &\quad+\frac{1}{u}\frac{\phi\left(X_u\right)}{u}\phi\left(X_c\right)\\ &\leq \left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\cdot\left(1+H_{t-1}\right)+\frac{u-t}{u}\cdot\phi\left(X_u\right)\\ &\quad+\frac{1}{u}\left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\\ &\leq \left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\cdot\left(1+H_{t-1}+\frac{1}{u}\right)+\frac{u-t}{u}\cdot\phi\left(X_u\right)\\ &\leq \left(\phi\left(X_c\right)+8\cdot\phi_{OPT}\left(X_u\right)\right)\cdot\left(1+H_t\right)+\frac{u-t}{u}\cdot\phi\left(X_u\right)\qquad\left(\text{car } t\leq u\Rightarrow\frac{1}{u}\leq\frac{1}{t}\right).\end{aligned}$$

L'inégalité est ainsi établie pour le couple (t,u). □

Théorème 1. *Soient C' un clustering obtenu lors de l'initialisation de l'algorithme des k-means++ et ϕ' sa fonction potentielle associée. Alors on a :*

$$\mathbb{E}[\phi'] \leq 8\cdot(\ln k+2)\cdot\phi_{OPT}$$

Démonstration.
Considérons le clustering C avec un seul centre obtenu après avoir choisi le premier centre de manière uniforme lors de la première étape de l'algorithme des k-means++ et notons ϕ sa fonction potentielle associée. Soit A le cluster de C_{OPT} dans lequel ce premier centre a été choisi. Le clustering C' de fonction potentielle ϕ' est obtenu alors en ajoutant à C les $k-1$ centres restants. Ainsi, en appliquant le lemme 3 avec $t=u=k-1$ et $X_c=A$ (donc $X_u=X-A$), nous obtenons :

$$\mathbb{E}[\phi'] \leq \left(\phi\left(A\right)+8\cdot\phi_{OPT}\left(X-A\right)\right)\cdot\left(1+H_{k-1}\right)+\frac{(k-1)-(k-1)}{k-1}\cdot\phi\left(X-A\right)$$

Comme

$$\phi_{OPT}(X - A) = \phi_{OPT}(X) - \phi_{OPT}(A) = \phi_{OPT} - \phi_{OPT}(A) \ ,$$

l'inégalité équivaut à

$$\mathbb{E}[\phi'] \leq \Big(\phi(A) + 8 \cdot \phi_{OPT} - 8 \cdot \phi_{OPT}(A)\Big) \cdot (1 + H_{k-1})$$

En passant à l'espérance dans les deux membres, l'inégalité est préservée grâce à la propriété de positivité et nous obtenons :

$$\mathbb{E}\Big[\mathbb{E}[\phi']\Big] \leq \mathbb{E}\left[\Big(\phi(A) + 8 \cdot \phi_{OPT} - 8 \cdot \phi_{OPT}(A)\Big) \cdot (1 + H_{k-1})\right]$$

Remarquons que $\mathbb{E}[\phi']$ est une constante. De même, dans le second membre de l'inégalité ci-dessus, tous les termes sont toujours constants sauf $\phi(A)$. Ainsi, par les propriétés de l'espérance, cette inégalité revient à :

$$\mathbb{E}[\phi'] \leq \Big(\mathbb{E}[\phi(A)] + 8 \cdot \phi_{OPT} - 8 \cdot \phi_{OPT}(A)\Big) \cdot (1 + H_{k-1})$$

Par le lemme 2, nous savons que

$$\mathbb{E}[\phi(A)] = 2 \cdot \phi_{OPT}(A)$$

En utilisant ce résultat, nous aboutissons donc à :

$$\begin{aligned} \mathbb{E}[\phi'] &\leq \Big(8 \cdot \phi_{OPT} - 6 \cdot \phi_{OPT}(A)\Big) \cdot (1 + H_{k-1}) \\ &\leq 8 \cdot \phi_{OPT} \cdot (1 + H_{k-1}) \end{aligned}$$

Finalement, en utilisant l'inégalité [2] $H_{k-1} < H_k \leq 1 + \ln k$, nous obtenons

$$\mathbb{E}[\phi'] \leq 8 \cdot \phi_{OPT} \cdot (2 + \ln k)$$

ce qui est le résultat attendu. □

I.3 Les k-means++ semi-supervisés

Dans cette section, nous nous intéressons au cas semi supervisé où certaines observations sont déjà labellisées. Nous introduisons deux nouvelles notations : $X^{(u)}$ et $X^{(s)}$ désignant respectivement les données non labellisées et les données supervisées (labellisées).

2. La démonstration de cette inégalité entre la série harmonique et le logarithme est rappelée en annexe 3 à la page 29

Présentation de l'algorithme

Ici aussi, seule l'initialisation change pour tenir compte des données supervisées. En effet, par rapport à l'initialisation de l'algorithme des k-means++, la première étape où l'on choisit le premier centre de manière uniforme est celle qui change. Elle est remplacée par le choix de G premiers centres, où G représente le nombre de labels distincts étiquetant les données supervisées ($G \leq k$). Ces G centrés sont définis comme isobarycentres de points portants leurs labels respectifs. Les autres centres restants ($k - G$) sont choisis avec une probabilité proportionnelle au poids D^2 exactement comme dans l'algorithme d'initialisation des k-means++. La suite de l'algorithme est la même en recourant à l'algorithme de Lloyd. Voici alors l'algorithme d'initialisation pour ce cas :

Algorithme d'initialisation des centres pour les k-means++

Entrées : $X^{(u)}$ (données non labellisées), $X^{(s)}$ (données labellisées) et k (le nombre de centres)
Sorties : C (l'ensemble des centres initiaux)
1 poser n_ℓ le nombre d'observations labellisées avec le label ℓ
2 initialiser $C = \emptyset$
3 Pour $\ell = 1, 2, ..., k$ **faire**
4 **Si** $n_\ell > 0$ **alors**
5 poser c_ℓ égal au point moyen des données ayant ℓ pour label
6 actualiser $C = C \cup \{c_\ell\}$
7 **Fin Si**
8 Fin Pour
9 Tant que $card(C) \leq k$ **faire**
10 choisir un $x \in X^{(u)}$ avec probabilité proportionnelle à $D^2(x)$
11 actualiser $C = C \cup \{x\}$
12 Fin Tant que
13 sortir C

Critère de qualité

En suivant le même cheminement que pour les k-means++, nous démontrons que cette nouvelle manière d'initialiser les centres initiaux lorsqu'il existe des données supervisées permet d'améliorer la borne obtenue pour les k-means++. Ce résultat est énoncé dans le théorème suivant :

Théorème 2 : Supposons que nous avons des données supervisées et posons $C = \emptyset$. Pour chaque label ℓ pour lequel il y a des observations supervisées, nous ajoutons à C le point moyen c_ℓ de toutes ces observations. A la fin de ces ajouts, supposons que $card(C) = G$. Notons n_{ℓ_j} le nombre d'observations affectées du label ℓ_j et n_j le cardinal du cluster associé au label ℓ_j dans C_{OPT} pour $j = 1, 2, ..., G$. Il reste donc $u = k - G$ clusters non couverts par C dans C_{OPT}. Ajoutons alors $t = u$ nouveaux centres à C pour former C' en utilisant le poids D^2 et en ne prenant pas aucune observation labellisée.

La fonction potentielle ϕ' associée au clustering C' ainsi obtenu vérifie alors

$$\mathbb{E}[\phi'] \leq 8 \cdot (2 + \ln(k - G)) \cdot \phi_{OPT}$$

Les lemmes présentés dans cette section ont pour but de démontrer ledit théorème.

Lemme 5. *Soient $d \geq 1$ un entier, $A = \{x_i \in \mathbb{R}^d, i = 1, 2, ..., n\}$ un ensemble de points et $S \subset A$ un sous ensemble de A de cardinal g choisi aléatoirement de manière uniforme dans l'ensemble des parties de A à g éléments.*
Alors, on a :

$$\mathbb{E}\left[\sum_{x_i \in S} x_i\right] = \frac{g}{n} \sum_{x_i \in A} x_i$$

Démonstration.
Posons $X_i = \mathbb{1}_{\{x_i \in S\}}$. C'est une variable aléatoire définie sur A et suivant une loi de Bernoulli de paramètre $\frac{g}{n}$ car le tirage est uniforme. S étant de cardinal g, nous pouvons alors écrire :

$$\begin{aligned}
\mathbb{E}\left[\sum_{x_i \in S} x_i\right] &= \mathbb{E}\left[\sum_{i=1}^{n} x_i \cdot X_i \mid \sum_{i=1}^{n} X_i = g\right] \\
&= \sum_{i=1}^{n} x_i \mathbb{E}\left[X_i \mid \sum_{i=1}^{n} X_i = g\right] \text{ (par linéarité de l'espérance)}
\end{aligned}$$

Pour tout $1 \leq i \leq n$, la variable $X_i \mid \sum_{i=1}^{n} X_i = g$ est une variable de Bernoulli également. Calculer son espérance revient donc à déterminer la probabilité qu'elle prenne la valeur 1. Ainsi, nous avons :

$$
\begin{aligned}
\mathbb{E}\left[X_i \mid \sum_{i=1}^{n} X_i = g\right] &= \mathbb{P}\left(X_i = 1 \mid \sum_{i=1}^{n} X_i = g\right) \\
&= \frac{\mathbb{P}\left(X_i = 1\ ,\ \sum_{i=1}^{n} X_i = g\right)}{\mathbb{P}\left(\sum_{i=1}^{n} X_i = g\right)} \\
&= \frac{\mathbb{P}\left(X_i = 1\ ,\ \sum_{j=1, j\neq i}^{n} X_j = g-1\right)}{\mathbb{P}\left(\sum_{i=1}^{n} X_i = g\right)} \\
&= \frac{\mathbb{P}\left(X_i = 1\right) \cdot \mathbb{P}\left(\sum_{j=1, j\neq i}^{n} X_j = g-1\right)}{\mathbb{P}\left(\sum_{i=1}^{n} X_i = g\right)} \quad \text{(par indépendance)} \\
&= \frac{\frac{g}{n} \cdot \binom{n-1}{g-1} \cdot \left(\frac{g}{n}\right)^{g-1} \cdot \left(1-\frac{g}{n}\right)^{n-g}}{\binom{n}{g} \cdot \left(\frac{g}{n}\right)^{g} \cdot \left(1-\frac{g}{n}\right)^{n-g}} \\
&= \frac{\binom{n-1}{g-1}}{\binom{n}{g}} \\
&= \frac{g}{n}
\end{aligned}
$$

Il en résulte alors

$$\mathbb{E}\left[\sum_{x_i \in S} x_i\right] = \sum_{i=1}^{n} x_i \cdot \frac{g}{n}$$

ce qui conclut la preuve. □

Lemme 6. *Soient $d \geq 1$ un entier, $A = \{x_i \in \mathbb{R}^d, i = 1, 2, ..., n\}$ un ensemble de points et $S \subset A$ un sous ensemble de A de cardinal g choisi aléatoirement de manière uniforme dans l'ensemble des parties de A à g éléments.*
Alors, on a :

$$\mathbb{E}\left[\left(\sum_{x_i \in S} x_i\right)^{\intercal} \left(\sum_{x_i \in S} x_i\right)\right] = \frac{g(g-1)}{n(n-1)} \sum_{i \neq j} x_i^{\intercal}\, x_j + \frac{g}{n} \sum_{i=1}^{n} x_i^{\intercal}\, x_i$$

Démonstration.
Les hypothèses étant les mêmes que celles du lemme 5, la démonstration est du même type et utilise les mêmes arguments.
Posons $X_i = \mathbb{1}_{\{x_i \in S\}}$. C'est une variable aléatoire définie sur A et suivant une loi de Bernoulli de paramètre $\frac{g}{n}$ car le tirage est uniforme. S étant de cardinal g, nous pouvons alors écrire :

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_{x_i \in S} x_i\right)^\intercal \left(\sum_{x_i \in S} x_i\right)\right] &= \mathbb{E}\left[\left(\sum_{i=1}^{n} x_i \cdot X_i\right)^\intercal \left(\sum_{j=1}^{n} x_j \cdot X_j\right) \mid \sum_{l=1}^{n} X_l = g\right] \\
&= \mathbb{E}\left[\sum_{1 \leq i,j \leq n} x_i^\intercal x_j X_i X_j \mid \sum_{i=l}^{n} X_l = g\right] \\
&= \sum_{1 \leq i,j \leq n} x_i^\intercal x_j \mathbb{E}\left[X_i X_j \mid \sum_{i=l}^{n} X_l = g\right]
\end{aligned}$$

Pour tous $1 \leq i, j \leq n$, la variable $X_i X_j \mid \sum_{l=1}^{n} X_l = g$ est une variable de Bernoulli également. Calculer son espérance revient donc à déterminer la probabilité qu'elle prenne la valeur 1. Nous distinguons deux cas : le cas où $i = j$ et le le cas où $i \neq j$.

Si $i = j$, comme $X_i^2 = X_i$, nous nous retrouvons avec la même espérance qui a été calculée dans la démonstration du lemme 5 et qui vaut donc $\frac{g}{n}$

Si $i \neq j$, nous avons :

$$
\begin{aligned}
\mathbb{E}\left[X_iX_j \mid \sum_{l=1}^{n} X_l = g\right] &= \mathbb{P}\left(X_iX_j = 1 \mid \sum_{l=1}^{n} X_l = g\right) \\
&= \frac{\mathbb{P}\left(X_i = 1,\ X_j = 1\ ,\ \sum_{l=1}^{n} X_l = g\right)}{\mathbb{P}\left(\sum_{l=1}^{n} X_l = g\right)} \\
&= \frac{\mathbb{P}\left(X_i = 1,\ X_j = 1,\ \sum_{l=1,l\neq i,j}^{n} X_l = g-2\right)}{\mathbb{P}\left(\sum_{l=1}^{n} X_l = g\right)} \\
&= \frac{\mathbb{P}\left(X_i = 1\right)\mathbb{P}\left(X_j = 1\right)\cdot\mathbb{P}\left(\sum_{l=1,l\neq i,j}^{n} X_l = g-2\right)}{\mathbb{P}\left(\sum_{l=1}^{n} X_l = g\right)} \\
&= \frac{\left(\frac{g}{n}\right)^2\cdot\binom{n-2}{g-2}\cdot\left(\frac{g}{n}\right)^{g-2}\cdot\left(1-\frac{g}{n}\right)^{n-g}}{\binom{n}{g}\cdot\left(\frac{g}{n}\right)^{g}\cdot\left(1-\frac{g}{n}\right)^{n-g}} \\
&= \frac{\binom{n-2}{g-2}}{\binom{n}{g}} \\
&= \frac{g(g-1)}{n(n-1)}
\end{aligned}
$$

En combinant les deux cas nous obtenons alors

$$
\begin{aligned}
\mathbb{E}\left[\left(\sum_{x_i\in S} x_i\right)^{\intercal}\left(\sum_{x_i\in S} x_i\right)\right] &= \sum_{i\neq j} x_i^{\intercal}x_j\mathbb{E}\left[X_iX_j \mid \sum_{i=l}^{n} X_l = g\right] + \sum_{i=1}^{n} x_i^{\intercal}x_i\mathbb{E}\left[X_iX_j \mid \sum_{i=l}^{n} X_l = g\right] \\
&= \sum_{i\neq j} x_i^{\intercal}x_j\frac{g(g-1)}{n(n-1)} + \sum_{i=1}^{n} x_i^{\intercal}x_i\frac{g}{n}
\end{aligned}
$$

ce qui donne le résultat attendu. □

Nous allons maintenant démontrer le lemme 7. Précisons une chose : en plus de toutes les hypothèses qui sont données dans ce lemme, il faut noter d'une part que S est un ensemble des points labellisés et ayant le même label et d'autre part que la fonction ϕ dont il est question est la fonction potentielle associée au clustering avec un seul centre,

l'isobarycentre $\overline{a}_S$ de S, obtenu lors de la première étape de l'algorithme d'initialisation des k-means++ semi supervisés. Ainsi pour le cluster A contenant S, $\phi(A)$ revient tout simplement à $\sum_{a\in A} \|a - \overline{a}_S\|^2$. C'est dans ce sens que nous noterons exceptionnellement $\phi(A)$ par $\phi\left(A; \overline{a}_S\right)$ dans l'énoncé de ce lemme.

Lemme 7. *Soient $d \geq 1$ un entier, $A = \{x_i \in \mathbb{R}^d, i = 1, 2, ..., n\}$ un cluster dans C_{OPT} et $S \subset A$ un sous ensemble de A de cardinal g choisi aléatoirement de manière uniforme dans l'ensemble des parties de A à g éléments.*
Alors, on a :

$$\mathbb{E}\left[\phi\left(A; \overline{a}_S\right)\right] = \left(1 + \frac{n-g}{g(n-1)}\right) \phi_{OPT}(A)$$

avec

$$\phi_{OPT}(A) = \sum_{a_i \in A} \|a_i - c(A)\|^2 \quad ,$$

$c(A)$ le point moyen de A et $\overline{a}_S$ le point moyen de S

Démonstration.
Nous avons :

$$\begin{aligned}
\mathbb{E}\left[\phi\left(A; \overline{a}_S\right)\right] &= \mathbb{E}\left[\sum_{a\in A} \|a - \overline{a}_S\|^2\right] \\
&= \mathbb{E}\left[\sum_{a\in A} \|a - c(A)\|^2 + n\|c(A) - \overline{a}_S\|^2\right] \qquad \text{(par le lemme 1)} \\
&= \sum_{a\in A} \|a - c(A)\|^2 + n\mathbb{E}\left[\|\overline{a}_S - c(A)\|^2\right] \\
&= \phi_{OPT}(A) + n\mathbb{E}\left[\|\overline{a}_S - c(A)\|^2\right]
\end{aligned}$$

Calculons donc $\mathbb{E}\left[\|\overline{a}_S - c(A)\|^2\right]$:
Nous avons

$$\begin{aligned}
\|\overline{a}_S - c(A)\|^2 &= \langle \overline{a}_S - c(A)\ ,\ \overline{a}_S - c(A)\rangle \\
&= \langle \overline{a}_S, \overline{a}_S\rangle - 2\langle c(A), \overline{a}_S\rangle + \langle c(A), c(A)\rangle \\
&= \overline{a}_S^{\intercal}\ \overline{a}_S - 2\langle c(A), \frac{1}{g}\sum_{a\in S} a\rangle + c(A)^{\intercal}\ c(A) \\
&= \overline{a}_S^{\intercal}\ \overline{a}_S - \frac{2}{g}c(A)^{\intercal}\ \sum_{a\in S} a + c(A)^{\intercal}\ c(A)
\end{aligned}$$

En passant à l'espérance, nous obtenons :

$$
\begin{aligned}
\mathbb{E}\left[\|\overline{a}_S - c(A)\|^2\right] &= \mathbb{E}\left[\overline{a}_S^\intercal\ \overline{a}_S\right] - \frac{2}{g}c(A)^\intercal\ \mathbb{E}\left[\sum_{a\in S} a\right] + c(A)^\intercal\ c(A) \\
&= \mathbb{E}\left[\overline{a}_S^\intercal\ \overline{a}_S\right] - \frac{2}{g}c(A)^\intercal\ \frac{g}{n}\sum_{a\in A} a + c(A)^\intercal\ c(A) \quad \text{(par le lemme 5)} \\
&= \mathbb{E}\left[\overline{a}_S^\intercal\ \overline{a}_S\right] - 2c(A)^\intercal\ \frac{1}{n}\sum_{a\in A} a + c(A)^\intercal\ c(A) \\
&= \mathbb{E}\left[\overline{a}_S^\intercal\ \overline{a}_S\right] - 2c(A)^\intercal\ c(A) + c(A)^\intercal\ c(A) \\
&= \mathbb{E}\left[\frac{1}{g}\left(\sum_{a\in S} a\right)^\intercal\ \frac{1}{g}\left(\sum_{a\in S} a\right)\right] - c(A)^\intercal\ c(A) \\
&= \frac{1}{g^2}\left(\frac{g(g-1)}{n(n-1)}\sum_{i\neq j} a_i^\intercal\ a_j + \frac{g}{n}\sum_{i=1}^{n} a_i^\intercal\ a_i\right) - c(A)^\intercal\ c(A) \quad \text{(par le lemme 6)} \\
&= \frac{g-1}{gn(n-1)}\sum_{i\neq j} a_i^\intercal\ a_j + \frac{1}{gn}\sum_{i=1}^{n} a_i^\intercal\ a_i - c(A)^\intercal\ c(A) \\
&= \frac{g-1}{gn(n-1)}\sum_{i,j} a_i^\intercal\ a_j - \frac{g-1}{gn(n-1)}\sum_{i=j} a_i^\intercal\ a_j + \frac{(n-1)}{gn(n-1)}\sum_{i=1}^{n} a_i^\intercal\ a_i - c(A)^\intercal\ c(A) \\
&= \frac{g-1}{gn(n-1)}\left(\sum_{a_i\in A} a_i\right)^\intercal\left(\sum_{a_j\in A} a_j\right) + \frac{n-g}{gn(n-1)}\sum_{i=1}^{n} a_i^\intercal\ a_i - c(A)^\intercal\ c(A) \\
&= n^2\frac{g-1}{gn(n-1)}c(A)^\intercal\ c(A) + \frac{n-g}{gn(n-1)}\sum_{i=1}^{n} a_i^\intercal\ a_i - c(A)^\intercal\ c(A) \\
&= -\frac{n-g}{g(n-1)}c(A)^\intercal\ c(A) + \frac{n-g}{gn(n-1)}\sum_{i=1}^{n} a_i^\intercal\ a_i \\
&= \frac{n-g}{g(n-1)}\left(\frac{1}{n}\sum_{i=1}^{n} a_i^\intercal\ a_i - c(A)^\intercal\ c(A)\right) \\
&= \frac{n-g}{gn(n-1)}\left(\sum_{i=1}^{n}\|a_i\|^2 - n\|c(A)\|^2\right) \\
&= \frac{n-g}{gn(n-1)}\left(\sum_{i=1}^{n}\|a_i - c(A)\|^2\right) \quad \text{(en appliquant le lemme 1 avec z=0)} \\
&= \frac{n-g}{gn(n-1)}\phi_{OPT}(A)
\end{aligned}
$$

et le résultat est démontré !

□

Énonçons maintenant la version du lemme 4 pour le cas semi - supervisé et qui est utilisé dans la démonstration du théorème 2 sur la majoration de l'espérance de la fonction potentielle lorsqu' il y a des données labellisées :

Lemme 8. *Soient C un clustering arbitraire et ϕ sa fonction potentielle associée. Supposons qu'il y a u clusters non couverts par C dans C_{OPT} (avec $u \geq 0$ entier). Notons X_u l'ensemble des points de ces clusters et $X_c = X - X_u$ (son complémentaire) l'ensemble des points dans les clusters couverts. Supposons que nous ajoutons $t \leq u$ nouveaux centres à C (en excluant les données labellisées) choisis de manière aléatoire proportionnellement au poids D^2. Notons C' le nouveau clustering ainsi obtenu et ϕ' sa fonction potentielle associée. Alors, on a :*

$$\mathbb{E}\left[\phi'\right] \leq \left(\phi\left(X_c\right) + 8 \cdot \phi_{OPT}\left(X_u\right)\right) \cdot \left(1 + H_t\right) + \frac{u-t}{u} \cdot \phi\left(X_u\right);$$

avec $H_t = \sum_{i=1}^{t} \frac{1}{i}$ la $t^{ème}$ somme partielle de la série harmonique.

Démonstration. Elle est similaire à celle du lemme 4. Il suffit juste d'utiliser les probabilités tenant compte de nouvelles hypothèses ajoutées pour le cas semi-supervisé : étant donné un ensemble A, la probabilité de choisir un nouveau centre a dans cet ensemble proportionnellement au poids D^2 tout en veillant à ne pas choisir a parmi les données labellisées est alors donnée par :

$$\frac{\phi\left(A \cap X^{(u)}\right)}{\phi\left(X^{(u)}\right)}$$

où $X^{(u)}$ désigne l'ensemble des données non labellisées.

De plus, en remarquant que toute donnée labellisée est considérée comme faisant partie d'un cluster couvert, nous avons alors

$$X^{(s)} \cap X_u = \emptyset$$

où $X^{(s)}$ désigne l'ensemble des données labellisées (supervisées).

Ainsi, en tenant compte de ces remarques et en suivant le cheminement de la démonstration présentée pour le lemme 4, nous obtenons le résultat annoncé. □

Finalement, voici le résultat principal de cette partie :

Théorème 2. *Supposons que nous avons des données supervisées et posons $C = \emptyset$. Pour chaque label ℓ pour lequel il y a des observations supervisées, nous ajoutons à C le point moyen c_ℓ de toutes ces observations. A la fin de ces ajouts, supposons que $card(C) = G$. Notons n_{ℓ_j} le nombre d'observations affectées du label ℓ_j et n_j le cardinal du cluster associé au label ℓ_j dans C_{OPT} pour $j = 1, 2, ..., G$. Il reste donc $u = k - G$ clusters non couverts par C dans C_{OPT}. Ajoutons alors $t = u$ nouveaux centres à C pour former C' en utilisant le poids D^2 et en ne prenant pas aucune observation labellisée.*
La fonction potentielle ϕ' associée au clustering C' ainsi obtenu vérifie alors

$$\mathbb{E}[\phi'] \leq 8 \cdot (2 + \ln(k - G)) \cdot \phi_{OPT}$$

Démonstration.
Nous allons appliquer le lemme 8 avec $u = t = k - G$. Pour cela, remarquons d'abord que

$$X_c = \bigcup_{j=1}^{G} X_{\ell_j}$$

où X_{ℓ_j} désigne de le cluster de C_{OPT} correspondant au label ℓ_j. Ainsi,

$$\phi(X_c) = \sum_{j=1}^{G} \phi\left(X_{\ell_j}\right)$$

et

$$\begin{aligned}
\phi_{OPT}(X_u) &= \phi_{OPT}(X - X_c) \\
&= \phi_{OPT}(X) - \sum_{j=1}^{G} \phi_{OPT}\left(X_{\ell_j}\right) \\
&= \phi_{OPT} - \sum_{j=1}^{G} \phi_{OPT}\left(X_{\ell_j}\right)
\end{aligned}$$

Compte tenu de ces constats, le résultat du lemme 8 s'écrit alors :

$$\mathbb{E}[\phi'] \leq \left(\sum_{j=1}^{G} \left(\phi(X_{\ell_j}) - 8\phi_{OPT}(X_{\ell_j}) \right) + 8\phi_{OPT} \right) \cdot (1 + H_{k-G})$$

Et en passant à l'espérance dans le deuxième membre tout en gardant à l'esprit que tous les termes sont des constantes sauf $\phi(X_{\ell_j})$ et en appliquant le lemme 7 avec $A = X_{\ell_j}$ (S correspondant ici à l'ensemble des observations de X_{ℓ_j} qui sont labellisées avec $\overline{a}_S = c_{\ell_j}$) pour tout X_{ℓ_j}, nous obtenons :

$$\begin{aligned}
\mathbb{E}[\phi'] &\leq \left(\sum_{i=1}^{G} \left(\left(1 + \frac{n_j - n_{\ell_j}}{n_{\ell_j}(n_j - 1)} \right) \phi_{OPT}(X_{\ell_j}) - 8\phi_{OPT}(X_{\ell_j}) \right) + 8\phi_{OPT} \right) \cdot (1 + H_{k-G}) \\
&\leq \left(\sum_{i=1}^{G} \left(\left(-7 + \frac{n_j - n_{\ell_j}}{n_{\ell_j}(n_j - 1)} \right) \phi_{OPT}(X_{\ell_j}) \right) + 8\phi_{OPT} \right) \cdot (1 + H_{k-G}) \\
&\leq 8\phi_{OPT} \cdot (1 + H_{k-G})
\end{aligned}$$

la dernière inégalité étant due au fait que pour tout $1 \leq j \leq G$, nous avons

$$n_j - n_{\ell_j} \leq n_j - 1$$

ce qui implique que la somme sur les j intervenant dans le second membre de l'avant dernière inégalité est toujours négative.
Pour conclure, nous utilisons l'inégalité [3] $H_{k-G} \leq 1 + \ln(k - G)$ et nous obtenons

$$\mathbb{E}[\phi'] \leq 8 \cdot \phi_{OPT} \cdot \left(2 + \ln(k - G) \right)$$

ce qui est le résultat escompté. □

3. Voir l'annexe 3 à la page 29 pour la démonstration de cette inégalité

L'amélioration de la borne ne semble pas très importante comparée à la borne obtenue pour les k-means++. Cependant, au vu de la fin de la preuve, plus il y a des données supervisées, plus cet algorithme sera plus performant que celui des k-means++.

I.4 Résultats annexes utilisés dans cette partie

Dans cette section, nous présentons (pour information) trois résultats qui ont été utilisés indirectement ou directement dans les démonstrations effectuées précédemment.

Inégalité de Cauchy-Schwarz

Annexe 1. ***(Inégalité de Cauchy-Schwarz dans*** $\mathbb{R}^n$***)***
On considère l'espace euclidien $\mathbb{R}^n$*. Alors, on a :*

$$\forall x \in \mathbb{R}^n, \forall y \in \mathbb{R}^n, \langle x, y\rangle \ \leq \|x\| \ \|y\|.$$

Démonstration. Soient $x, y \in \mathbb{R}^n$ et $t \in \mathbb{R}$. On a :

$$0 \leq \|x + ty\|^2 = \|y\|^2 t^2 + 2\langle x, y\rangle t + \|x\|^2$$

Le discriminant de ce polynôme du second degré est donné par :

$$\Delta = 4\langle x, y\rangle^2 - 4\|x\|^2 \ \|y\|^2$$

Comme ce polynôme est de signe constant, il admet au plus une racine réelle et donc son discriminant est négatif ou nul :

$$\begin{aligned} \Delta \leq 0 \ &\Leftrightarrow \ \langle x, y\rangle^2 - \|x\|^2 \ \|y\|^2 \leq 0 \\ &\Leftrightarrow \ |\langle x, y\rangle| \ \leq \|x\| \ \|y\| \ ; \end{aligned}$$

d'où le résultat escompté. □

Inégalité des moyennes

Annexe 2. ***(Inégalité des moyennes : arithmétique et quadratique)***
Soient $u \geq 1$ *un entier et* $a_1, a_2, ..., a_u$ *des réels. Alors :*

$$\frac{1}{u}\sum_{i=1}^{u} a_i \leq \sqrt{\frac{1}{u}\sum_{i=1}^{u} a_i^2}.$$

Démonstration. C'est une application immédiate de l'inégalité de Cauchy-Schwarz dans R^u (démontrée en annexe 1) en prenant $x = (a_1, a_2, ..., a_u)$ et $y = \left(\frac{1}{u}, \frac{1}{u}, ..., \frac{1}{u}\right)$. □

Majoration des sommes partielles de la série harmonique

Annexe 3. ***(Majoration de la série harmonique par le logarithme)***
Pour tout $k \geq 1$ *entier, on pose* $H_k = \sum_{i=1}^{k} \frac{1}{i}$ *la* $k^{ème}$ *somme partielle de la série harmonique . Alors, on a :*

$$\forall k \geq 1 \quad , H_k \leq 1 + \ln k$$

Démonstration.
Pour $k = 1$, l'inégalité est trivialement vérifiée.
Supposons alors $k \geq 2$. Pour tout $2 \leq i \leq k$, nous nous intéressons alors à la fonction $x \longmapsto \frac{1}{x}$ restreinte à l'intervalle $[i-1, i]$. Cette fonction est continue et strictement décroissante. Ainsi,

$$\forall x \in [i-1, i], \quad \frac{1}{i} \leq \frac{1}{x}$$

Et par positivité de l'intégrale,

$$\frac{1}{i} \leq \int_{i-1}^{i} \frac{1}{x} dx$$

Et en sommant sur toutes les valeurs de i et en utilisant la relation de Chasles, nous obtenons :

$$1 + \sum_{i=2}^{k} \frac{1}{i} \leq 1 + \int_{1}^{k} \frac{1}{x} dx$$

Soit

$$H_k \leq 1 + \ln k$$

Ainsi, l'inégalité est bien établie. □

II Partie appliquée

Dans cette section, nous allons tester et comparer les différents algorithmes de kmeans, kmeans++ et kmeans++ semi-supervisés sur un jeu de données clients. Nos mesures de performances sont le coût tel qu'estimé par la fonction potentielle, ainsi que la durée d'exécution de l'algorithme.

II.1 La Base de Données

La base de données contient 60366 observations (clients) et 39 variables quantitatives et qualitatives. Tous les clients sont anonymisés, ce qui limitera l'interprétation de l'analyse. Nous commençons par analyser la base de données. Dans la Figure 1 nous affichons les statistiques d'une partie de nos données.

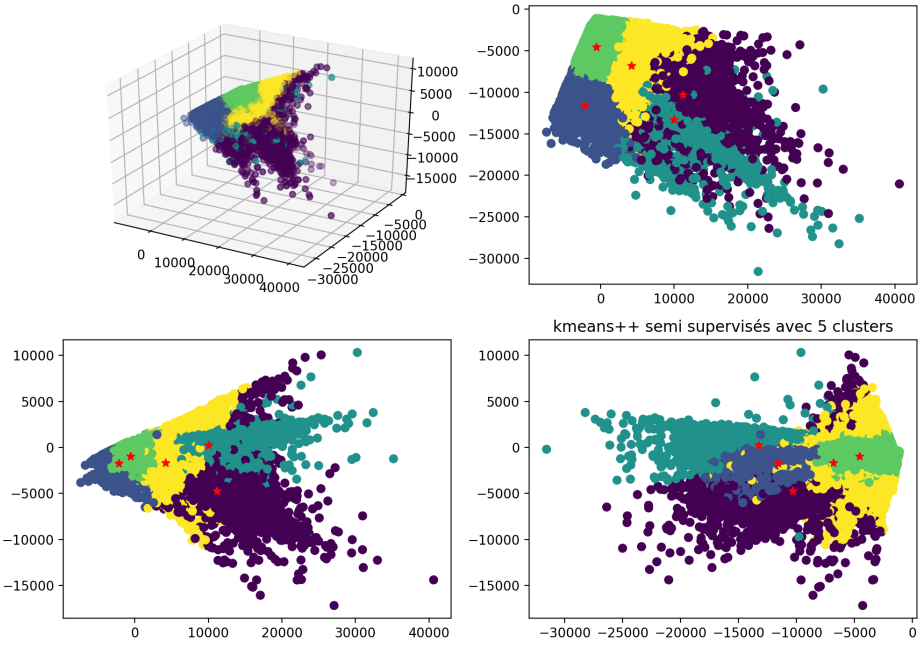
FIGURE 1 – Sommaire des statistiques d'une partie des données

Index	g_distance_to_sho	r_distance_to_sho	x_distance_to_sho	roducts_purchase	e_products_purch	mount_purchase(	avg_purchase	avg_price	shops_used	distance_shop_1	distance_shop_2	distance_shop_3	distance_shop_4	distance_shop_5	ucts_purchased_sl
count	60366	60366	60366	60366	60366	60366	60366	60366	60366	60366	60366	60366	60366	60366	60366
mean	2030.23	1396.43	2942.67	1778.7	330.668	4235.46	8.53208	3.66651	2.38005	2496.63	2488.24	1924.97	2882.8	2020.9	887.813
std	1119.9	1048.41	1327.52	2185.04	236.002	5006.74	10.3154	9.13317	1.01218	1281.53	1417.36	1157.75	1730.09	1260.85	1438.48
min	6.64247	6.64247	6.64247	1	1	0.212	0.212	0.212	1	93.2834	11.1904	17.8443	6.64247	25.4607	0
25%	1250.24	573.611	2117.22	227	127	653.118	4.53291	2.53375	2	1609.58	1501.9	1253.84	1532.68	1173.41	49
50%	1936.88	1184.94	2869.2	925	304	2355.14	6.86278	2.94426	2	2288.67	2355.4	1746.17	2704.32	1852.93	292
75%	2569.71	1962.38	3580.67	2551.75	500	6054.46	10.4295	3.55811	3	3144.06	3370.72	2323.56	4056	2466.56	1072
max	9004.16	9004.16	9267.7	22131	1465	51588.7	787.569	787.569	5	8019.92	9004.16	7395.25	9273.69	7465.81	17016

Nous voyons qu'il existe une seule variable qualitative, la variable 'shops used', les autres étant toutes des variables quantitatives continues. Nous remarquons aussi que toutes les variables quantitatives affichent de grandes dispersions. Par exemple, en regardant le nombre de produits achetés, nous voyons que certains clients achètent très peu, soit 1 seul article, tandis que d'autres achètent jusqu'à 22 131 articles, la moyenne étant 2 552.75. Ce type d'observation nous permettra de mieux interpréter les clusters proposés par les différentes versions des k-means.

Une Analyse en Composantes Principales (ACP) est effectuée afin de mieux regrouper et comprendre les variables quantitatives. Les 3 premières composantes principales expliquent moins de la moitié de l'inertie, soit 41.58% de la variance expliquée. Nous n'étudierons pas les résultats de l'ACP dans notre situation si ce n'est pour nous aider à mieux distinguer les clusters proposés par la méthode de kmeans++ semi supervisé, et mieux les associer aux variables du jeu de données.

FIGURE 2 – ACP en 3 dimensions



II.2 Choix du nombre de clusters

La méthode kmeans est une méthode d'apprentissage non-supervisée. Lors de son application les données sont séparées en plusieurs classes dont le nombre est prédéfini de façon à regrouper les individus ayant le plus de similarité. C'est ainsi qu'une des tâches clefs est de trouver le nombre approprié de classes, k. Il existe plusieurs techniques pour déterminer le nombre de classes. Nous en évoquons quelques unes parmi les plus connus :

1. Méthode du pouce :

Cette méthode est une méthode approximative où le nombre de classes, k est déterminé par : $k \approx \sqrt{n/2}$

2. L'indice de qualité :

Afin d'évaluer la qualité de la classification, les indices inertiels (l'inertie intra-classes et l'inertie inter-classes) sont souvent utilisés. L'inertie intra-classes 'mesure le degré d'homogénéité entre les objets appartenant à la même classe' tandis que l'inertie inter-classes 'mesure le degré d'hétérogénéité entre les classes.' Il existe plusieurs indices de qualité, par exemple l'indice de Dunn, l'indice de Calinski et Harabasz (CH) ou encore l'indice de Silhouette. Le premier calcule la distance minimale inter-classes et ainsi plus cette distance est grande, meilleur sera la classification. Le nombre de clusters qui maximise l'indice de Dunn sera pris comme le nombre optimal de clusters k. Toutefois, nous notons que cette méthode a un coût qui augmente proportionnellement à l'augmentation du nombre de clusters et du nombre de dimensions. Ainsi elle ne sera pas utilisée dans notre étude.

Introduit par Kauffman et Rousseew, l'indice de Silhouette nous donne une représentation visuelle de la distance entre un point d'une classe avec les points des classes voisines. Plus le coefficient ainsi calculé est proche de 1, plus la distance avec les classes voisines (inertie inter) est grande. Ceci s'observe notamment pour le nombre de classes optimal. À l'inverse, un coefficient proche de -1 nous indique une mauvaise classification de l'observation. Il faut déterminer les valeurs de silhouette moyenne pour différents nombres de clusters. Le nombre optimal de clusters sera celui qui affichera la valeur moyenne la plus élevée. Quand cette valeur sera la plus proche de 1, cela suggérera que les observations sont mieux classifiées.

3. Méthode du coude :

La méthode du coude est une technique visuelle très connue. L'idée derrière cette technique est d'utiliser la méthode k-means en parcourant k valeurs. À chacune des k valeurs, l'inertie intra-classe (c'est à dire, la distance moyenne entre les observations d'un cluster) est calculée et est affichée sur un graphique, nous permettant de mieux visualiser les résultats. L'objectif est de choisir la valeur k (qui sera le nombre de classes) créant un effet de 'coude', c'est-à-dire provoquant une baisse plus conséquente, plus soudaine l'inertie intra-classe. Nous disons ceci en gardant

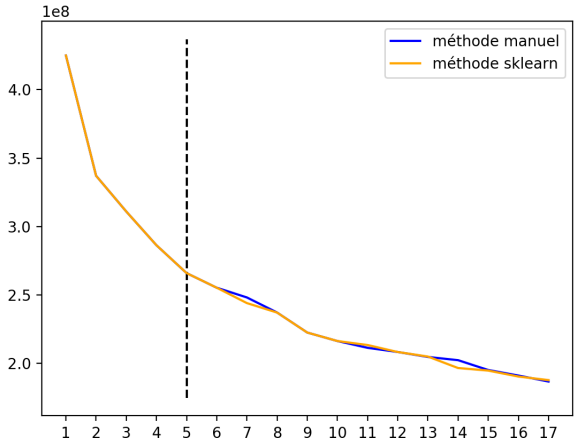
en tête que la somme des distances intra-classes aura toujours tendance à baisser, plus la valeur de k est grande.

4. La validation croisée :

La validation croisée regarde la stabilité des classes. C'est une méthode d'estimation de fiabilité d'un modèle basé sur un système d'échantillonnage. La validation croisée k-fold est utilisée pour déterminer le nombre optimal de clusters, k. Ce processus consiste en premier lieu à diviser l'échantillon en k sous-ensembles. Un de ces sous-ensembles est ensuite sélectionné comme ensemble de validation tandis que les k-1 autres sont utilisés comme ensemble d'apprentissage (dit données test). L'opération se répète k fois pour chacun des k sous-ensemble, et à chaque fois les données sont modélisées grâce à une régression, et la performance du modèle est prise en note. Ainsi, cette procédure est effectuée en utilisant différentes valeurs de k. La valeur optimale de k celle qui afficherait la plus petite erreur de validation croisée. Dans [5], une approche différente est mentionnée pour les données non-supervisées. Toutefois nous ne la présentons pas car nous n'allons pas l'utiliser dans cette étude (focalisée plus sur les résultats du semi-supervisé).

Dans le cadre de notre étude, nous choisissons de travailler avec la méthode du coude, une méthode que nous appliquons sur la méthode de kmeans++ semi-supervisé avec 60% de données labellisées. Cette méthode est d'ailleurs comparée avec celle de scikit-learn (ou encore sklearn) afin de déterminer l'exactitude de l'algorithme utilisé. Scikit-learn est une bibliothèque python souvent utilisée afin d'appliquer des méthodes de machine learning sur python. Elle contient des fonctions déjà créées, telles que les fonctions de kmeans et de kmeans++. Elle contient également des fonctions nous aidant à déterminer le nombre de clusters optimal. Ainsi nous l'utilisons à quelques reprises lors de cette étude afin de comparer nos résultats et afin de nous aider à faire la pseudo-labellisation utilisée par la suite par la méthode des kmeans++ semi-supervisés. Figure 3 affiche la répartition des sommes des erreurs au carré à la fois pour notre méthode, dite la méthode manuelle, ainsi que la méthode proposée par sklearn. Nous voyons une baisse plus soudaine lorsque nous avons 5 clusters. Ce sera ainsi le choix du nombre de clusters utilisé lors de l'application des algorithmes de clustering.

FIGURE 3 – Méthode du coude, comparaison sklearn et méthode manuelle



II.3 Résultats des expérimentations

Dans cette section nous cherchons à présenter les résultats obtenus par les algorithmes kmeans, kmeans++ et kmeans++ semi supervisés. De plus ce dernier a été analysé plus en détails afin d'afficher l'impact des différents pourcentages de données labellisées. Comme pour l'ACP la variable qualitative a été écartée de notre analyse. 6 essais ont été effectués afin d'évaluer le temps d'exécution des algorithmes (en secondes). La moyenne des distances intra-classes a été obtenue en se basant sur 100 simulations. Le nombre de clusters, déterminé en utilisant la méthode du coude, est 5. Nous baserons le reste de notre étude sur ces 5 clusters.
Nous cherchons en premier lieu à comparer les résultats de différents algorithmes. Les tableaux ci-dessous les affichent.

Tel qu'attendu la durée d'exécution moyenne de l'algorithme de kmeans est plus longue que celle du kmeans++ et du kmeans++ semi-supervisée (avec 60% de données labellisées). Cette dernière est d'ailleurs en moyenne 3 fois plus rapide que le kmeans ou le kmeans++. Au niveau du coût de la fonction nous voyons que le kmeans ++ semi supervisée affiche à nouveau le meilleur résultat (avec la plus petite distance moyenne intra-classes sur 100 simulations).

Nous notons toutefois que la distance intra-classe moyenne lorsque kmeans est utilisé est plus petite que celle du kmeans++. Bien qu'on se serait attendu à obtenir le résultat inverse, cette observation peut être due au fait que le jeu de données en question soit bien adapté aux kmeans. Ainsi, la qualité de classification obtenue par les kmeans est aussi

TABLE 1 – **Résultats en utilisant kmeans**

Temps d'exécution Minimum	Temps d'exécution Maximum	Temps d'exécution Moyen	Moyenne des distances intra-classe
0.867459	4.278867	1.867284	267024131.97

TABLE 2 – **Résultats en utilisant kmeans++**

Temps d'exécution Minimum	Temps d'exécution Maximum	Temps d'exécution Moyen	Moyenne des distances intra-classe
0.952380	1.963426	1.603216	269311160.94

TABLE 3 – **Résultats en utilisant kmeans++ semi-supervisées avec 60% de données labellisées**

Temps d'exécution Minimum	Temps d'exécution Maximum	Temps d'exécution Moyen	Moyenne des distances intra-classe
0.386259	0.616933	0.481936	265971783.28

bonne que celle des kmeans++. Cependant, les kmeans++ restent bien meilleurs dans le sens où l'algorithme est plus rapide pour des classifications considérées équivalentes.

TABLE 4 – **Résultats en utilisant kmeans++ semi-supervisées avec 25% de données labellisées**

Temps d'exécution Minimum	Temps d'exécution Maximum	Temps d'exécution Moyen	Moyenne des distances intra-classe
0.534357	0.759242	0.629477	265972507.4

TABLE 5 – **Résultats en utilisant kmeans++ semi-supervisées avec 99% de données labellisées**

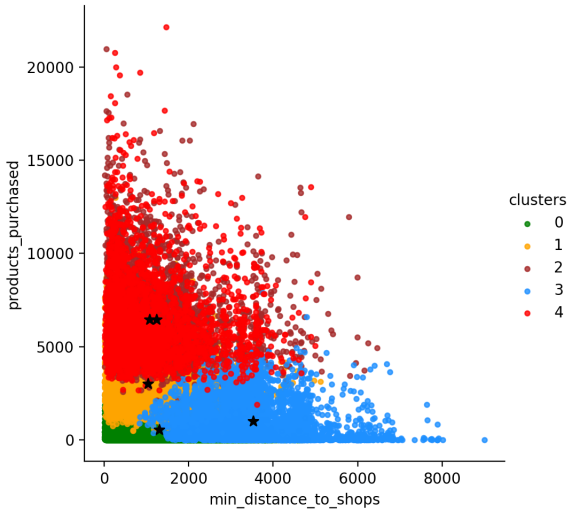
Temps d'exécution Minimum	Temps d'exécution Maximum	Temps d'exécution Moyen	Moyenne des distances intra-classe
0.115119	0.288531	0.232994	265976831.1

À partir des tableaux 4 et 5, nous voyons que plus le pourcentage de données est labellisé, plus le temps d'exécution est rapide. Toutefois, nous ne pouvons pas dire autant des distances intra-classe moyenne. En effet lorsque les données sont labellisées à 99% nous avons une inertie plus élevée que lors que les données sont moins labellisées. Une nouvelle fois la problématique pourrait venir du jeu de données, ou encore du fait que la supervision est en fait une pseudo-supervision car les vrais labels n'étaient pas disponibles. Ainsi cela expliquerait la variation des inerties moyennes. C'est pour cela que nous nous baserons davantage sur la durée d'exécution comme mesure de performance.

Interprétation

Nous cherchons désormais à interpréter les résultats obtenus sur le jeu de données utilisé dans cette étude. Figure 4 affiche les 5 clusters ainsi que leurs centres, tels que proposés par l'algorithme des kmeans++ semi-supervisés avec 60% de données labellisées. Nous rappelons que le nombre de clusters a préalablement été choisi par la méthode du coude.

FIGURE 4 – Affichage 5 clusters et 5 centres, determiné par kmeans++ semi-supervisé



Nous pouvons clairement voir que les clusters 1 et 3 (orange et rouge) sont moins distincts entre eux contrairement aux autres clusters. Effectivement, les centres de ces deux clusters sont très proches suggérant que nous aurions pu regrouper ces deux clusters. Nous avons choisi ici de représenter le nuage de points associé aux variables `products_purchased` et `min_distance_to_shops`. Ces variables ont été choisies car les clusters sont bien discriminés.
Ce que nous pouvons déduire du graphique est que les clients appartenant au cluster 4 sont ceux parcourant les plus petites distances par rapport aux magasins à l'inverse de ceux du cluster 0. D'ailleurs nous voyons que ces derniers achètent le moins de produits, tout comme les clients du cluster 2. Ainsi, nous pouvons établir un lien entre la distance parcourue pour aller au magasin et le nombre de produits achetés. En général, plus un client devra parcourir une plus longue distance, moins il ira au magasin, et le moins de produits il achètera. Sur un plan commercial et marketing, ce seront ainsi les clients des clusters 1,3 et 4 qui devraient être ciblés comme étant des acheteurs potentiels.

Nous cherchons désormais à établir d'autres liens avec les variables utilisées dans notre jeu de données. Nous commençons par analyser la variable qualitative `shops_used`. Figure 5 nous affiche la répartition des magasins ciblés dans l'étude. Les magasins 4 et 5 semblent

être les plus proches des clients. Pour le magasin 3 et le 4 (bien que la répartition du nuage de celui-ci est moins distinct que celui du magasin 3 sur le graphique), il est clair que moins le client voyage, plus il achète des produits. Toutefois vu la grande quantité de données, il est difficile de bien différencier les clusters, ainsi il faudra d'autres graphiques pour confirmer nos résultats.

FIGURE 5 – Variable qualitative

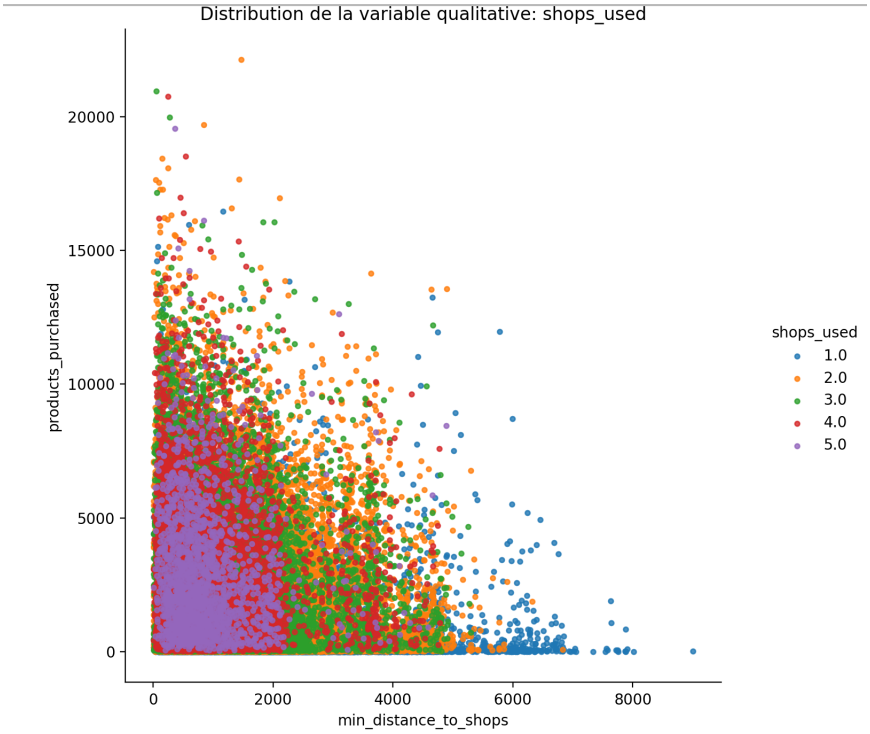
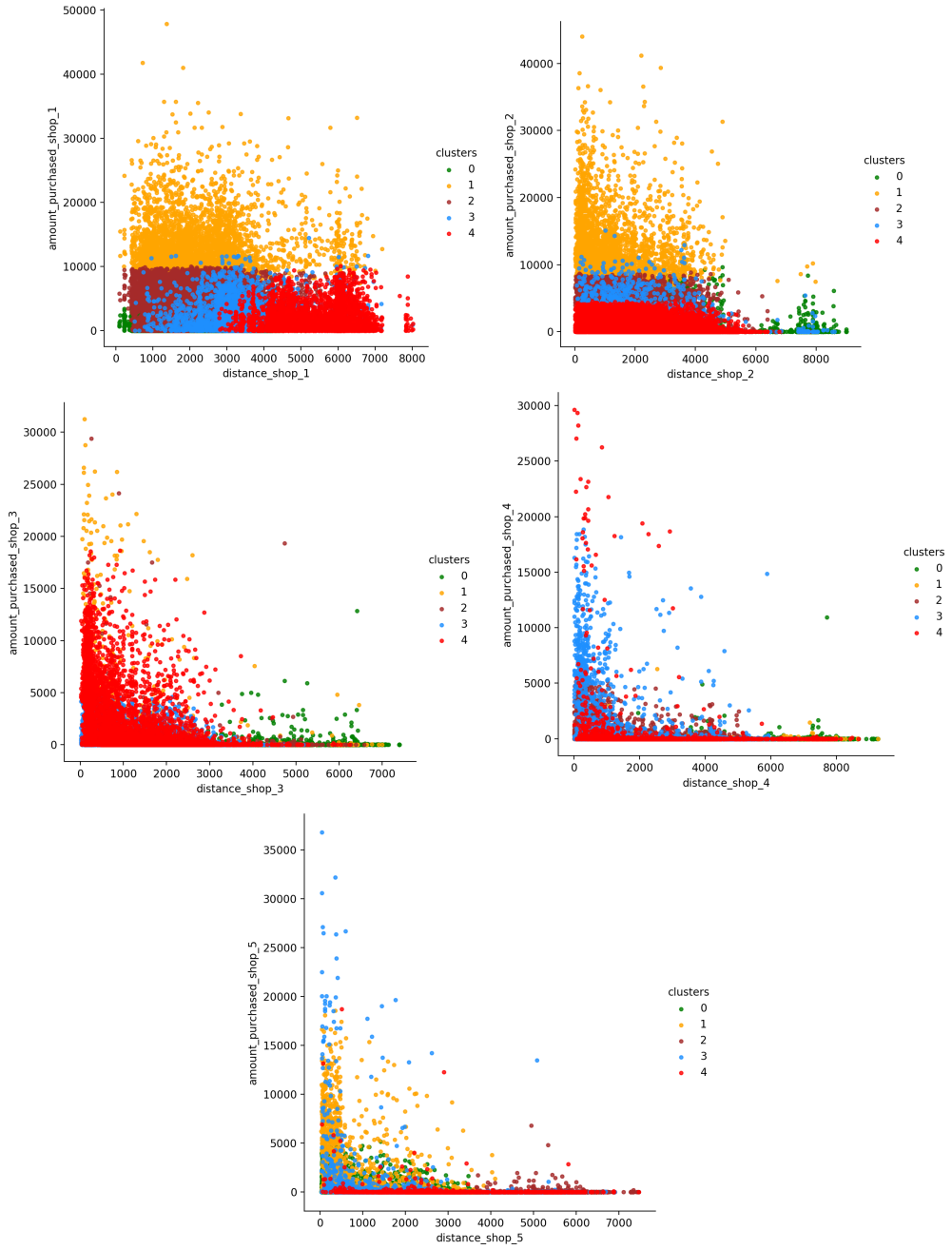


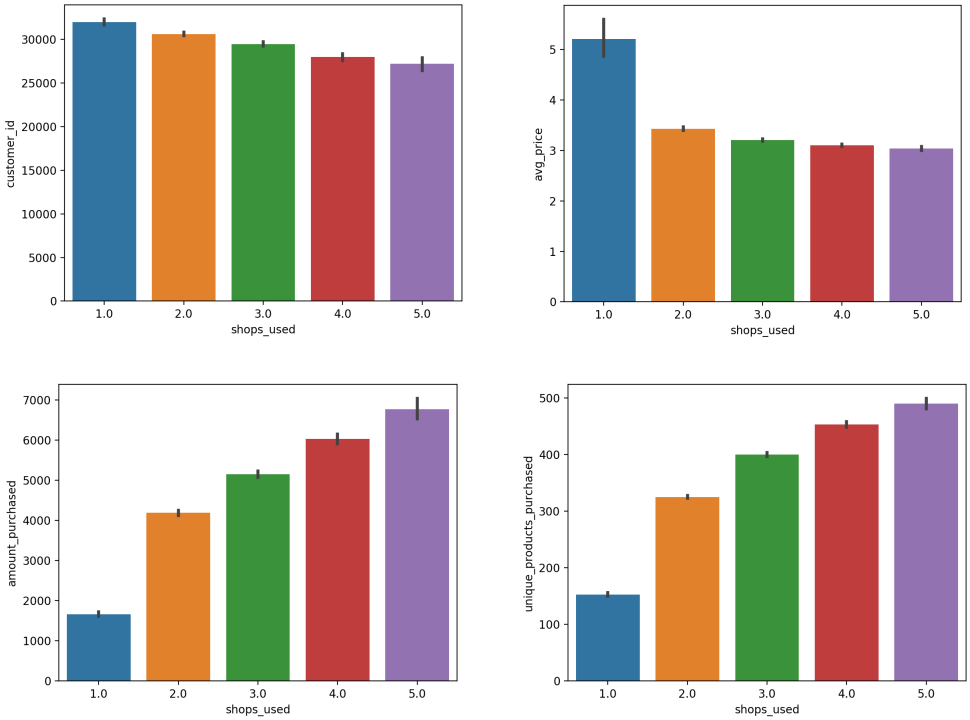
Figure 6 ci-dessous affichent la relation entre la distance pour aller à chaque magasin et le nombre de produits achetés dans ces magasins. Nous voyons dans le premier graphique que les clients des clusters 1, 2 et 3 semblent avoir une préférence pour le magasin 1. Les clients du cluster 5 sont ceux qui semblent habiter le plus loin du magasin 1. En revanche bien qu'ils en soient plus éloignés, leur consommation semble être un peu plus grande qu'au magasin 2, suggérant qu'ils préfèrent acheter les produits du magasin 1 au 2. Pour les 3 derniers graphiques, il est difficile d'établir un lien évident car les observations sont mélangées et peu distinguables. Nous nous baserons sur d'autres graphiques afin de tirer des conclusions.

FIGURE 6 – Relation nombre de produits acheté et distance pour chaque magasin



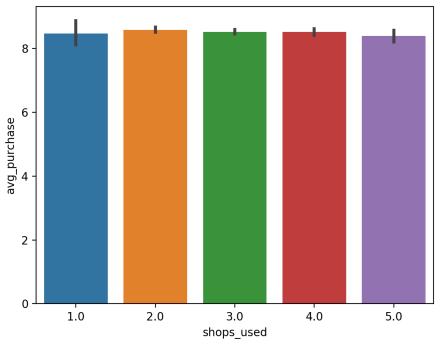
Nous utilisons des diagrammes en barre (voir ci-dessous) afin de nous aider à mieux comprendre le nombre de clients allant dans les magasins ainsi que les quantités exactes de produits vendus. Nous voyons que le magasin 5 est celui qui vend le plus de produits (unique et globalement) et c'est celui qui affiche le plus petit prix moyen des produits. Ainsi, les magasins 5 et 4 sont les magasins les plus vendeurs. Le magasin 1 est celui qui vend les produits les plus chers. Il n'est donc pas étonnant de voir qu'il vend une plus petite quantité de produits. Finalement, nous voyons quand même que plus de clients vont aux magasins 1 et 2, le magasin 5 étant celui qui a le moins de clients. De ces observations nous comprenons que les magasins 1 et 2 sont des magasins qui vendent des produits plus chers, avec pour conséquence le fait que même si les clients y vont plus souvent, ils ne vont pas nécessairement acheter les produits de ces magasins.

FIGURE 7 – Diagrammes de barre affichant le prix moyen, nombre de produits vendus et nombre clients allant dans chaque magasin



Toutes ces observations nous poussent à dire que la consommation diminue lorsque les prix augmentent, et ainsi les magasins les moins chers, soit 4 et 5, sont ceux vendant le plus de produits. Ils ont également plus de produits uniques que les magasins plus chers comme les 1 et 2. Ces derniers attirent beaucoup de clients, mais ceux-ci n'achèteront pas nécessairement les produits, d'où le fait que ces types de magasins vendent moins de produits. Au niveau des marges d'affaires de ces magasins, nous voyons dans le diagramme ci-dessous qu'il n'y a pas de grande différence entre ces boutiques. Effectivement, le quantité de consommation est compensée par le prix, ce qui fait que la vente moyenne reste plus ou moins égale.

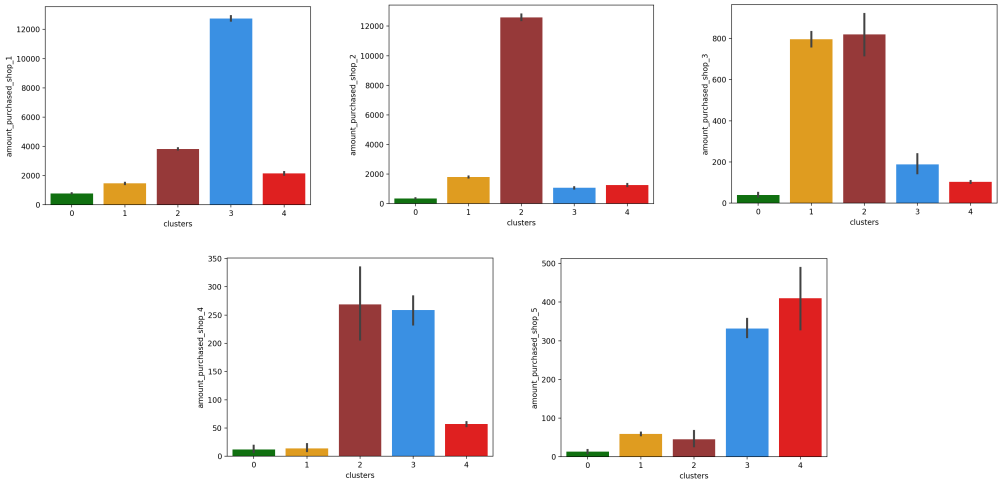
FIGURE 8 – Nombre moyen de produits vendus par chaque magasin



Nous cherchons maintenant à lier ces observations aux clusters obtenus par les kmeans++ semi-supervisés. Pour cela nous utilisons à nouveau des diagrammes en barre qui affichent le nombre d'individus de clusters qui achètent dans les différents magasins. À l'aide de ces graphiques (Figure 9), nous pouvons établir que certains magasins attirent davantage certains clients.

Le magasin 1 attire les individus du cluster 3, tandis que le magasin 2 attire plus le cluster 2, le magasin 3 attire les individus des clusters 1 et 2, le magasin 4 quant à lui attire les clients des clusters 2 et 3, et finalement le magasin 5 attire principalement les individus des clusters 4 et 3. Dans tous les cas le cluster 0 est celui qui est le moins représenté, confirmant le fait que ce cluster pourrait facilement être regroupé avec un autre cluster.

FIGURE 9 – Nombre de produits achetés par chaque cluster, pour chaque magasin



III Conclusion

Notre étude visait en premier lieu à évaluer la performance des k-means++ semi-supervisées par rapport aux k-means standards, et aux k-means++. Pour ce faire, nous avons fait une étude théorique de ces différents algorithmes avant de les appliquer sur un jeu de données clients.

Dans la partie théorique, nous avons vu que l'algorithme des k-means standards n'offre aucune garantie sur la qualité du clustering qu'il fournit. Ensuite, avons montré qu'en changeant la manière d'initialiser les centres dans les k-means++, nous obtenons un critère de qualité qui permet d'avoir une idée de la précision de cet algorithme en fonction du clustering optimal. Plus précisement, ce critère montre que le coût d'exécution de l'algorithme mesuré par la fonction potentielle, fonction obtenue comme somme des carrés

de distances intra clusters, est un $O(\ln k)$ par rapport au clustering optimal. Nous avons finalement vu qu'introduire des données supervisées dans le processus d'apprentissage pour contrôler l'initialisation des centres permettait d'améliorer davantage ce critère et d'obtenir un coût qui est un $O(\ln(k - G))$, où G désigne le nombre de clusters concernés par la supervision.

Dans la partie appliquée, les performances de ces trois algorithmes ont été comparées à l'aide des mesures comme le temps d'exécution ainsi que le coût estimé par la fonction potentielle. Nous avons vu que le coût, soit l'inertie intra classes moyenne, était généralement meilleur pour les kmeans++ semi supervisés comme attendu. Mais il est arrivé qu'il soit légèrement supérieur en augmentant le pourcentage de supervision alors que nous nous attendions à ce qu'il puisse baisser. Cela serait peut-être dû au fait que notre jeu de données ne contient pas de vrais labels. Mais en nous basant sur le temps d'exécution, nous avons vu clairement que les kmeans++ semi-supervisés restent la méthode la plus efficace en temps, en étant presque 3 fois plus rapide que les deux autres méthodes de classification non-supervisée abordées dans cette étude.

Enfin, nous avons utilisé la méthode semi-supervisée pour effecteur une segmentation des clients. L'état des données ne permettant pas de faire une analyse approfondie (données anonymisées notamment), nous avons obtenu des conclusions limitées. En décidant de labelliser les données à 60% et en choisissant d'analyser 5 clusters, comme déterminé par la méthode du coude, nous avons quand même trouvé quelques informations, certes limitées, mais pertinentes. Nous avons principalement noté que la segmentation des clients peut être faite en se basant sur les magasins. Par exemple, la boutique 1 vend plus aux clients du cluster 3, la boutique 5 vend plus aux clients du cluster 4, ou encore la boutique 2 attire principalement le cluster 2. De plus à l'aide de diagrammes supplémentaires, nous avons constaté que certains magasins vendent des produits plus chers, comme les magasins 1 et 2, suggérant que ceux qui y achètent, les individus de clusters 3 et 2 principalement, seraient plus aisés. De plus, nous avons remarqué que le cluster 0 est de petite taille et regroupe les individus qui vivent plus près des magasins mais qui achètent le moins de produits.

Pour finir, nous signalons que même avec des kmeans++ semi-supervisés, nous n'étions pas capables de caractériser davantage les clusters obtenus pour notre jeu de données. Dans un cas pratique en entreprise par exemple, une étude plus approfondie peut être menée au sein de chaque magasin en ciblant les individus des clusters identifiés dans cette étude qui y viennent afin de cerner davantage leurs attentes.

Références

[1] Oumaima Alaoui Ismailli, Vincent Lemaire et Antoine Cornuèjols : Une méthode supervisée pour initialiser les centres des k-moyennes. *Conférence Internationale Francophone sur l' Extraction et Gestion des Connaissances*, 2016.

[2] David Arthur et Sergei Vassilvitskii : K-means++ : The advantages of careful seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics*, 2007.

[3] Fabien Panloup : Apprentissage statistique en grande dimension. *Notes de cours du Master 2 Data Science 2017 - 2018.*

[4] Xin Wang, Chaofei Wang et Junyi Shen : Semi-supervised k-means clustering by optimisation initial cluster centers. *Web Information Systems and Mining*, 2011.

[5] Fu Wei et Carey E. Priebe : Estimating the number of clusters using cross-validation. *Stern School of Business, New York University*, 2017.

[6] Jordan Yoder et Carey E. Priebe : Semi-supervised k-means++. *Journal of Statistical Computation and Simulation*, 2017.