Notes Clustering et k-means

B Deporte

29 février 2024

Résumé

Notes de cours Introduction au Machine Learning pour L3 Maths

1 Introduction

On rappelle qu'il existe trois grandes familles d'algorithmes en Machine Learning : apprentissage supervisé, non-supervisé, par renforcement.

— **Apprentissage supervisé** :
L'apprentissage supervisé (**supervised learning** en Anglais) est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés. On distingue les problèmes de **régression** (prédiction d'une ou plusieurs variable(s) continue(s)), des problèmes de **classification** (prédiction d'une classe ou variable(s) discrète(s)).

La base d'apprentissage est la base d'entraînement, labelisée, et on fait l'hypothèse que la base d'entraînement est représentative de la distribution de probabilité des points de l'espace de départ. La performance de l'algorithme est déterminée par sa capacité à généraliser à des données non présentes dans la base d'apprentissage.

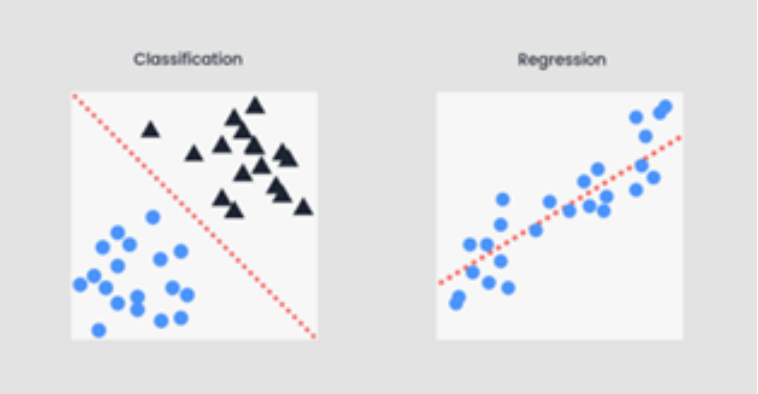

FIGURE 1 – Apprentissage supervisé

— **Apprentissage non-supervisé** :
L'apprentissage non supervisé est l'apprentissage automatique de structures sous-jacentes à des données non labelisées. Puisque les données ne sont pas étiquetées, il est impossible à l'algorithme de calculer de façon certaine un score de réussite. On distingue généralement les méthodes d'estimation de densités de probabilité et les méthodes discriminatives qui calculent des frontières de décision pour affecter une catégorie à un échantillon.

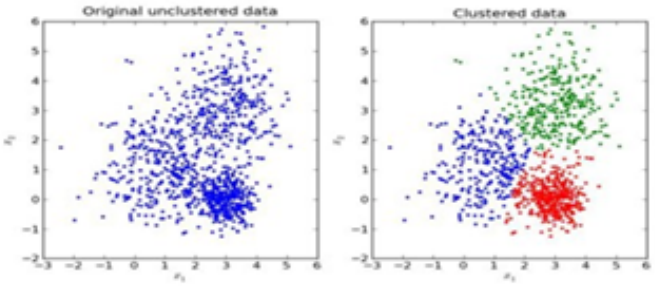

FIGURE 2 – Apprentissage non supervisé

— **Apprentissage par renforcement** :
L'apprentissage par renforcement consiste, pour un agent autonome (ex. : robot, agent conversationnel, personnage dans un jeu vidéo, etc.), à apprendre les actions à prendre (**policy**), à partir d'expériences (**actions space**), de façon à optimiser une récompense quantitative au cours du temps (**rewards**). L'agent est plongé au sein d'un environnement (**states space**) et prend ses décisions en fonction de son état courant. En retour, l'environnement procure à l'agent une récompense, qui peut être positive ou négative. L'agent cherche, au travers d'expériences itérées, un comportement décisionnel (**policy**) optimal, en ce sens qu'il maximise la somme des récompenses au cours du temps.

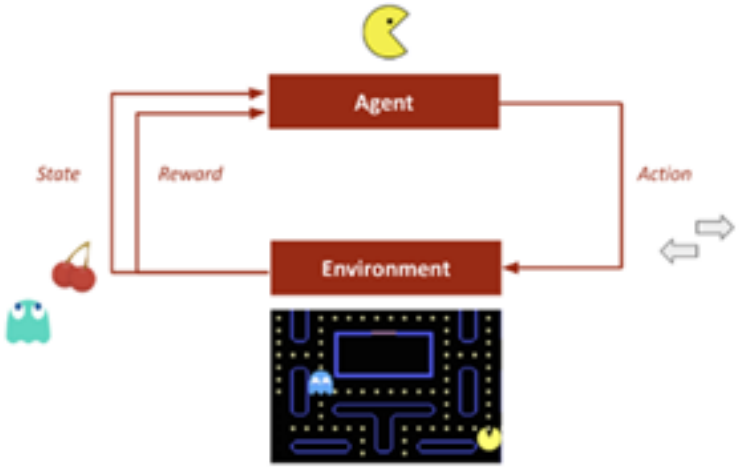

FIGURE 3 – Apprentissage par renforcement

Le sujet du cours est un type particulier d'apprentissage non-supervisé, appelé "**k-means**".

2 Intuition et metrics

On dispose d'un dataset de n points $\mathscr{X} = (x_i)_{1 \leq i \leq n} \subset \mathbb{R}^d$. L'idée est de chercher à grouper les x_i de façon aussi pertinente que possible en k groupes ou "clusters". Les points ne sont pas labellisés, c'est donc un problème d'apprentissage non-supervisé.

2.1 Exemple en 2D

Dataset initial (ici formé de cinq distributions Gaussiennes de $\mathbb{R}^2$)

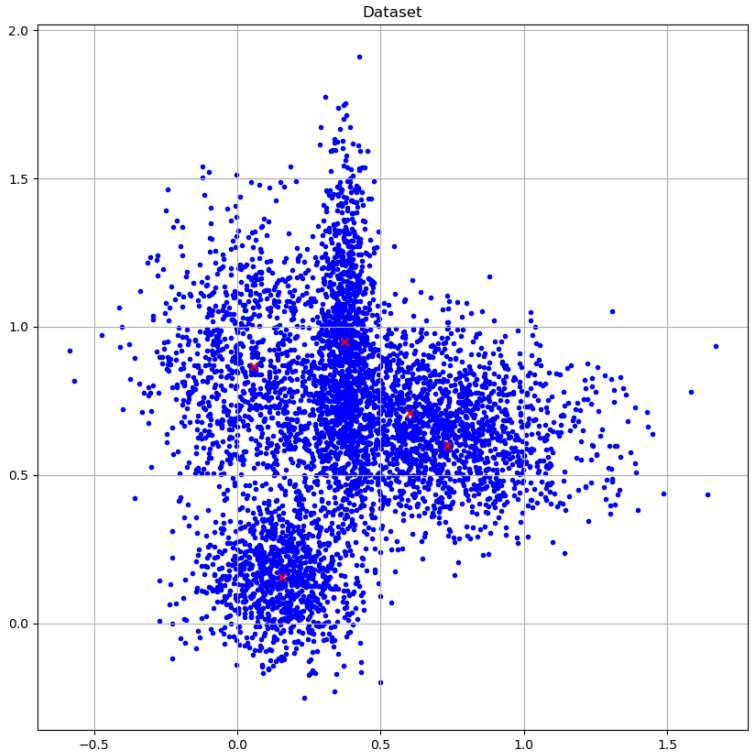

FIGURE 4 – Dataset initial

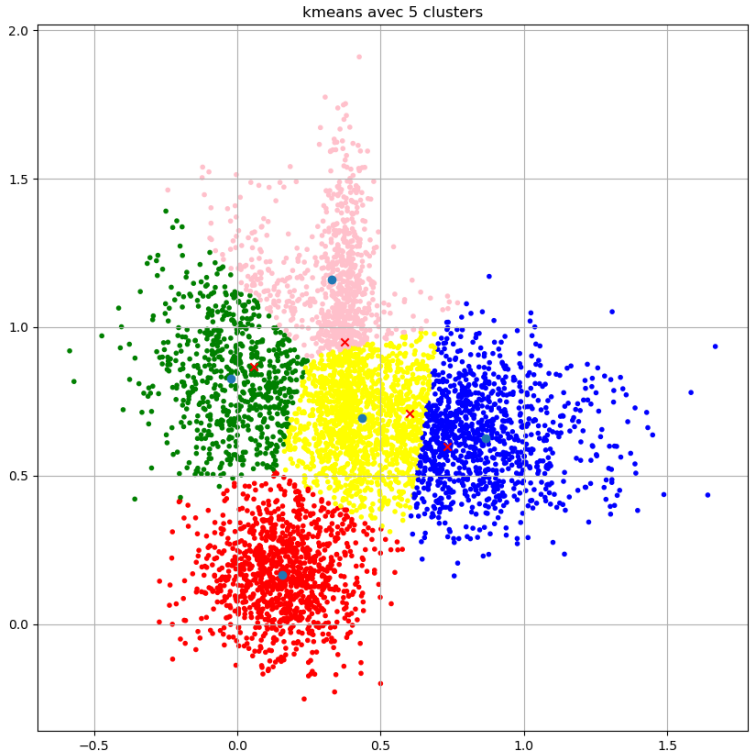

FIGURE 5 – Dataset avec 5 clusters

2.2 Formalisation

Mathématiquement, on cherche :

— un **quantifieur** d'ordre k, ou **mapping**, défini comme une fonction $q : \mathbb{R}^d \longrightarrow \mathscr{C}$ avec $\mathscr{C} = \{c_1, ..., c_k\}$ ($\mathscr{C}$ est appelé "alphabet"). où chaque point c_i est "représentatif" de C_i.

— une **partition** $\mathscr{P} = \{C_1, ..., C_k\}$ de $\mathbb{R}^d \cap \mathscr{X}$:

$$\bigcup_{1\leq i\leq n} C_i = \mathbb{R}^d \cap \mathscr{X}$$

$$\forall i,j, i\neq j \implies C_i \cap C_j = \emptyset$$

avec la numérotation imposée par :

$$q(x) = c_l \iff x \in C_l$$

Chaque point x_i est donc "assigné" à un unique cluster C_j par q. Pour k donné, la question est de déterminer une fonction de mapping q "performante", une partition $C_1, ..., C_k$ associée, et l'ensemble des points $c_1, ..., c_k$ (l'alphabet).

2.3 Le critère k-means

L'idée de "clustering" implique que l'on sait distinguer des points proches ou lointains, donc implique la connaissance d'une distance. On utilisera la plupart du temps la norme 2 (Euclidienne) dans $\mathbb{R}^d$:

$$\forall x \in \mathbb{R}^d, \|x\|_2^2 = \sum_i x_i^2$$

Le critère **k-means** consiste à chercher à minimiser la valeur moyenne des distances intra-clusters.

Formellement, l'objectif est :

$$\min_{C_1,...,C_k} \sum_{l=1}^{k} \frac{1}{|C_l|} \sum_{x_i,x_j\in C_l} \|x_i - x_j\|^2 \tag{1}$$

Pour chaque cluster C_l, on calcule la variance moyenne intra-cluster. Puis on cherche l'ensemble de clusters qui va minimiser la somme de ces variances moyennes.

Il faut ré-écrire différemment cette fonction objectif pour pouvoir avoir une forme 'tractable'.

On introduit alors, pour chaque cluster C_l, le centre de masse ou **centroïd** :

$$c_l = \frac{1}{|C_l|} \sum_{x_i\in C_l} x_i$$

Puis on calcule un peu :

$$\begin{aligned}
\sum_{x_i,x_j\in C_l} \|x_i - x_j\|^2 &= \sum_{x_i,x_j\in C_l} \left(\|x_i\|^2 + \|x_j\|^2 - 2\langle x_i\ ,\ x_j\rangle\right) \\
&= \sum_{x_i\in C_l} \left(|C_l|\|x_i\|^2 + \sum_{x_j\in C_l} \|x_j\|^2 - 2|C_l|\langle x_i\ ,\ c_l\rangle \right) \\
&= 2|C_l| \sum_{x_i\in C_l} \|x_i\|^2 - 2|C_l|^2\|c_l\|^2
\end{aligned}$$

où $\langle \cdot\ ,\ \cdot \rangle$ est le produit scalaire de $\mathbb{R}^d$.

De plus :

$$\begin{aligned}
\sum_{i\in C_l} \|x_i - c_l\|^2 &= \sum_{x_i\in C_l} \left(\|x_i\|^2 + \|c_l\|^2 - 2\langle x_i\ ,\ c_l\rangle\right) \\
&= \sum_{x_i\in C_l} \|x_i\|^2 + |C_l|\|c_l\|^2 - 2\langle \sum_{i\in C_l} x_i\ ,\ c_l\rangle \\
&= \sum_{x_i\in C_l} \|x_i\|^2 - |C_l|\|c_l\|^2
\end{aligned}$$

Donc :

$$\frac{1}{2} \sum_{x_i,x_j\in C_l} \|x_i - x_j\|^2 = |C_l| \sum_{x_i\in C_l} \|x_i - c_l\|^2$$

Chercher :

$$\min_{C_1,...,C_k} \sum_{l=1}^{k} \frac{1}{|C_l|} \sum_{x_i,x_j \in C_l} \|x_i - x_j\|^2$$

est donc équivalent à chercher :

$$\min_{\substack{C_1,...,C_k; \\ c_1,...,c_k}} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - c_j\|^2 \tag{2}$$

Le problème (2) est non-convexe, et est NP-complet. Il n'y a donc pas d'algorithme polynomial en temps permettant de trouver une solution.

3 Algorithme de Lloyd

On note que si l'on connaissait les centroïds c_j, alors on pourrait facilement assigner chacun des x_i au bon cluster en trouvant :

$$\min_j \|x_i - c_j\|^2$$

cad en assignant x_i au cluster dont le centre est le plus proche.

D'autre part, on note que, pour un ensemble $x_1, ..., x_n \in \mathbb{R}^d$, la fonction $\sum_{i=1}^{n} \|x_i - c\|^2$ est minimale pour $c = \frac{1}{n} \sum_i x_i$ (démo : écrire l'expression comme une fonction quadratique de c, calculer et annuler le gradient).

3.1 Description, limitations

Cela suggère un algorithme itératif, appelé algorithme de Lloyd ou algorithme k-means :

1. Choisir k centroïds initiaux $c_1, ..., c_k$
2. Assigner chaque point x_i au cluster "correct" C_j avec $j = argmin \|x_i - c_j^t\|^2$
3. Mettre à jour les centroïds : $c_j^{(t+1)} = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$
4. Répeter les steps 2 et 3 jusqu'à atteindre un critère de convergence.

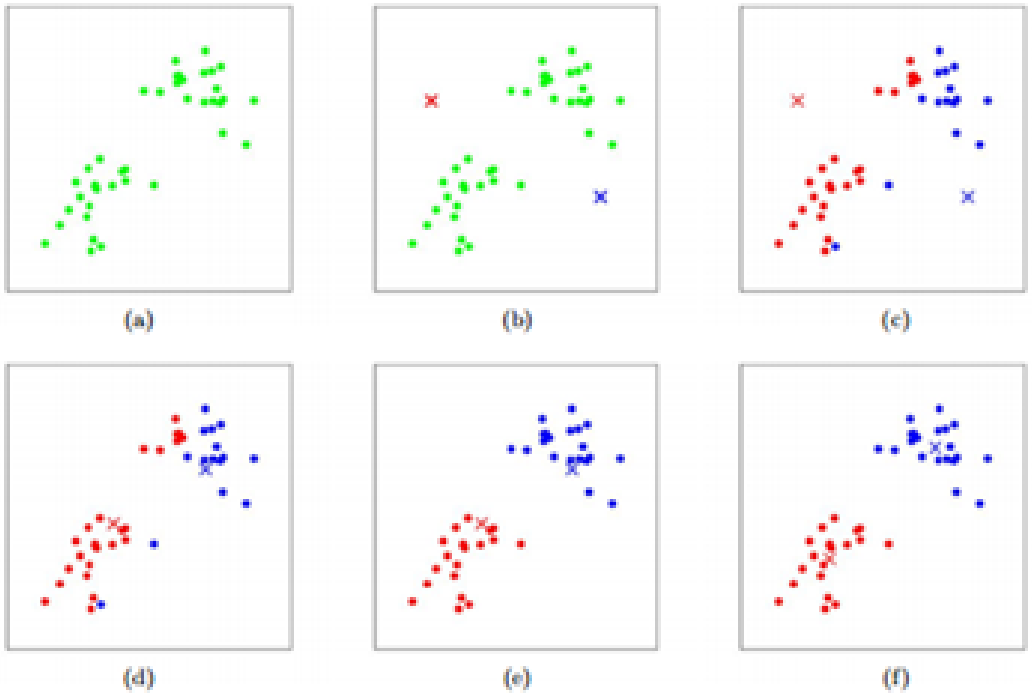

FIGURE 6 – Itérations k-means

L'algorithme de Lloyd est très populaire mais a des limitations à connaître absolument :

— Il est sensible à l'initialisation des centroïds. Des initialisations différentes produiront en général des clusters différents au final. (NB : l'initialisation par défaut dans l'implémentation scikit-learn résoud ce problème en grande partie).
— Rien ne garantit que la solution trouvée est la solution optimale. En d'autres termes, on peut trouver un optimum local.
— L'algorithme de Lloyd produit des clusters convexes, ce qui ne donne une bonne solution que si les clusters sont linéairement séparables.

3.2 Partitions de Voronoï, meilleur mapping d'ordre k

On va montrer ici que l'algorithme de Lloyd produit un mapping de type "plus proche voisin", dont la famille est celle qui contient le meilleur mapping possible à l'ordre k.

On considère dans la suite un alphabet $\mathscr{C} = \{c_1, ..., c_k\}$ de points de $\mathbb{R}^d$.

3.2.1 Quantifieurs

On a vu qu'un mapping, ou **quantifieur**, est défini comme une fonction :

$$q : x \in \mathbb{R}^d \longrightarrow q(x) \in \mathscr{C}$$

qui assigne un centre $c_i = q(x)$ à chaque x. On note $\mathscr{P} = \{A_1, ..., A_k\}$ la partition associée à q.

Un quantifieur est donc une fonction q qui "compresse" l'information contenue dans le dataset $(x_i)_{1 \leq i \leq k}$ en mappant chaque x_i à un centre c_j

Ce faisant, le quantifieur produit une erreur, que l'on définit comme la **distorsion** :

$$D(q, \mathscr{P}) = \sum_i \|x_i - q(x_i)\|^2$$

Certains quantifieurs sont plus intéressants que d'autres.

3.2.2 Partition de Voronoï et quantifieurs des plus proches voisins

On se place dans $\mathbb{R}^d$, que l'on munit d'une distance $d(x, y)$. NB : d est souvent la distance Euclidienne $\|.\|^2$, mais peut être une autre distance (norme 1, etc.)

Soit $(\mu_j)_{1 \leq j \leq k} \in \mathbb{R}^d$ un ensemble de *sites*, ou *générateurs*. La **cellule de Voronoï** ou **région de Voronoï** R_j associée au site μ_j est l'ensemble des points $x \in \mathbb{R}^d$ défini par :

$$R_j = \{x \in \mathbb{R}^d : d(x, \mu_j) \leq d(x, \mu_l) \; \forall \, l \neq j\}$$

Le **diagramme de Voronoï** ou **partition de Voronoï** est l'ensemble des R_j.

Propriété : dans le cas particulier de $\mathbb{R}^d$, les cellules de Voronoï sont convexes.

NB : la définition se généralise à un espace métrique quelconque (X, d)

On note $\mathscr{P}_V(\mathscr{C}) = \{R_1, ..., R_k\}$ la partition de Voronoï associée à $\mathscr{C}$.

Un **quantifieur des plus proches voisins** d'ordre k est un quantifieur si sa partition est une partition de Voronoï associée à son alphabet.

Distorsion d'un quantifieur des PPV - on a :

$$\begin{aligned} D(q_{PPV}, \mathscr{C}) &= \sum_{i=1}^{n} \|x_i - q_{PPV}(x_i)\|^2 \\ &= \sum_{i=1}^{n} \min_l \|x_i - c_l\|^2 \end{aligned}$$

par définition de la partition de Voronoï $\mathscr{P}_V$.

Proposition - De plus, pour tout quantifieur $q = (\mathscr{P}, \mathscr{C})$ d'ordre k, on a $D(q_{PPV}, \mathscr{P}_V) \leq D(q, \mathscr{P})$

Preuve :

$$\begin{aligned} D(q_{PPV}, \mathscr{P}_V) &= \sum_{i=1}^{n} \min_{1 \leq l \leq k} \|x_i - c_l\|^2 \\ &\leq \sum_{i=1}^{n} \|x_i - q(x_i)\|^2 \\ &= D(q, \mathscr{P}) \end{aligned}$$

Conclusion : à un ordre k donné, les meilleurs quantifieurs sont à chercher parmis les quantifieurs de type Plus Proches Voisins.

3.3 Convergence de l'algorithme de Lloyd

Le principe de la preuve est le suivant :
— si au moins un des centres est déplacé pendant le step 3 de l'algorithme de Lloyd, alors la distorsion décroît strictement.
— il y a un nombre fini d'itérations possibles.

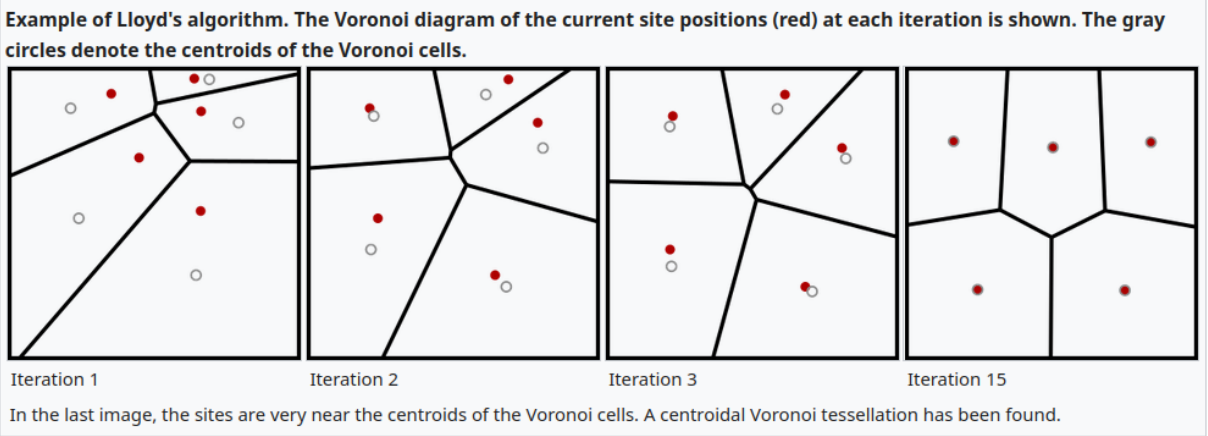

FIGURE 7 – Page Wiki

Notations
On note
— $\{c_1, ..., c_k\}$, les centres des clusters, et $\{C_1, ..., C_k\}$ les clusters au début du step 2.
— $\{c_1, ..., c_k\}$, les centres des clusters, et $\{C'_1, ..., C'_k\}$ les clusters à la fin du step 2 : les clusters ont été mis à jour.
— $\{c'_1, ..., c'_k\}$, les centres des clusters, et $\{C'_1, ..., C'_k\}$ les clusters à la fin du step 3 : les centres des clusters ont été mis à jour.
Et la fonction de coût :

$$\phi(c_1, ..., c_k, C_1, ..., C_k) = \sum_{l=1}^{k} \sum_{x \in C_l} \|x - c_l\|^2$$

On note d'abord que :

$$\phi(c_1, ..., c_k, C_1, ..., C_k) \geq \phi(c_1, ..., c_k, C'_1, ..., C'_k)$$

puisque, pour $c_1, ..., c_k$ fixés, la construction des $C'_1, ..., C'_k$ assigne chaque point x au centroïd le plus proche.

Ensuite, si $c'_i \neq c_i$ pour un i (cad si au moins un centre est déplacé lors du step 3), alors :

$$\phi(c_1, ..., c_k, C'_1, ..., C'_k) > \phi(c'_1, ..., c'_k, C'_1, ..., C'_k) \tag{3}$$

En effet, en notant que c'_i est le centre de masse de C'_i :

$$\begin{aligned} \sum_{x \in C'_i} \|x - c_i\|^2 &= \sum_{x \in C'_i} \|x - c'_i\|^2 + |C'_i|.\|c_i - c'_i\|^2 \\ &> \sum_{x \in C'_i} \|x - c'_i\|^2 \end{aligned}$$

Ce qui démontre (3) en sommant sur tous les clusters.

Enfin, on note que le nombre de $(C_1, ..., C_k)$ possible est k^n : chaque point x peut être assigné à un des k clusters possibles. Le nombre d'itérations de l'algorithme est majoré par k^n, car sinon, il devrait repasser deux fois sur le même clustering, ce qui est impossible puisque ϕ décroît strictement.

Conclusion : l'algorithme de Lloyd se termine obligatoirement.

4 Discussions

4.1 Notes sur implémentation dans scikit-learn

Lire absolument la doc scikitlearn : https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

4.2 Notes sur le choix du nombre de clusters

Une bonne discussion au chapitre 9 de https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/