INF 301

TP 1 & 2 Étude et implementation de l'algorithme k-means

Louhaïdia Oussama

oussama.louhaidia@telecom-bretagne.eu

Octobre 2011

Table des matières

1 Introduction **2**

2 Description de L'algorithme **2**

3 Étude des points clés de l'Algorithme **4**

3.1 Indices de qualitée 4

3.2 Valeur de k 5

3.3 Initialisation 5

3.4 La distance 6

4 Complexité algorithmique **7**

5 Étude de la base I.R.I.S **7**

1 Introduction

Le k-means est un des algorithmes heuristique (résultat approximative) de partitionnement de données, appliquer à un ensemble de données, il regroupe se qui se rapproche (notion de distance & de proximité) dans une même classe. Ca a plusieurs applications, dans tous les domaines.
Parmi les autres algorithmes de partitionnement[10] :

- Le regroupement hiérarchique où selon un indice de similarité on regroupe les éléments de deux classes. Chaque éléments appartient à sa propre classe au début de l'algorithme.
- L'analyse en composantes principales est une approche géométrique, utilisé le plus souvent dans le traitement d'images.
- La méthode des nuées dynamiques qui est une généralisation du k-means.
- L'algorithme EM Gaussien dont le k-means n'est qu'une version simplifier.

Toutes ces méthodes sont NP-difficile dans leurs cas généraux. Des simplifications, ou des heuristiques sont misent en place pour réduire cette complexité. Plusieurs variantes de cette technique existent, notamment le k-means globale et le k-means séquentiel.

Néamoins le kmean reste l'algorithme le plus populaire dû à sa simplicité conceptuelle, sa rapidité et sa faible exigence en mémoire.

2 Description de L'algorithme

L'algorithme k-means définit par McQueen[3] est un des plus simples algorithmes de classification automatique de donnée. Il revient à minimiser le critère quadratique définie par :

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} \|x_i - \mu_j\|^2 \tag{1}$$

Avec μ les k centres des classes, et x les n éléments à classer. Chaque élément sera absorbé par le groupe qui lui est le plus proche, jusqu'à stabilisation de l'algorithme. Cette procédure vise à maximiser la ressemblance entre les éléments d'un même groupe.

Dans sa forme la plus simple, son idée principale et de choisir aléatoirement un ensemble de centres fixé a priori et de chercher itérativement la partition optimale de l'ensemble des points. Chaque individu est affecté au centre qui lui est le plus proche, après l'affectation de toutes les données le centre de gravité de chaque groupe est alors recalculé, ils constituent les

nouveaux représentants des groupes, lorsqu'ont abouti à un état stationnaire (aucune donnée ne change de groupe) l'algorithme est arrêté. Plusieurs logiciel implémentent le k-means notamment Matlab et VisuMap.

Algorithm 2.1: K-MEANS(x, k)

Inputs
$\mapsto$ Un fichier de N données, notées par x
$\mapsto$ Nombre de groupes souhaité, noté par k
Output
$\mapsto$ Un fichier des partitions des K groupes $C_1, C_2, ..$
Start
Initialisation aléatoire des centres C_k
repeat
-**Affectation :** générer une nouvelle partition en assignant chaque objet au groupe dont le centre est le plus proche
-**Représentation :** Calculer les centres associés à la nouvelle partition
until convergence de l'algorithme vers une partition stable

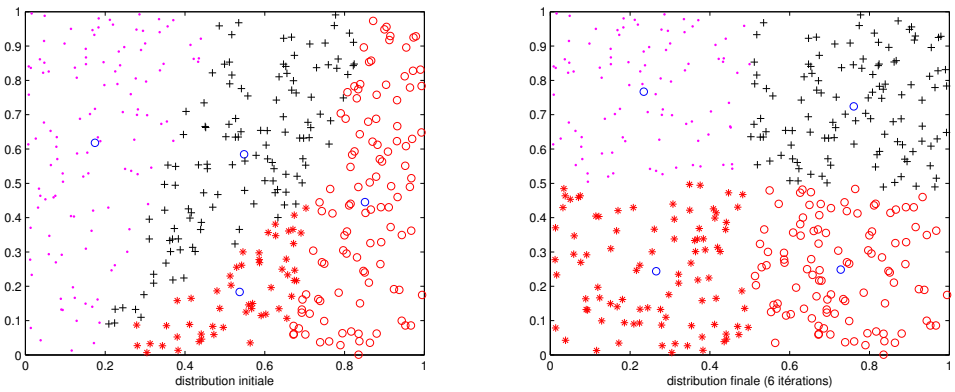

FIGURE 1 – Premier test sur 1000 données bidimentionnelle aléatoire

Pour faciliter les tests et les les observations des résultats obtenues, on utilise Matlab, qui reçoit à travers la procédure `plot.matlab` du module `inout.py` les m codes qui affiche l'ensemble des données sur le plan, ou dans l'espace. on se donne un ensemble de 1.000 point sur le plan. On se basera sur les critères de Dunn et de Davies-Bouldin pour analysé les performances des

différents algorithmes selon leurs différents paramètres. On affichera aussi la valeur du critère quadratique qui sert notamment à voir la différence dans le cas ou on compare deux expérience avec le même k.

3 Étude des points clés de l'Algorithme

Sauf dans les cas bidimensionelle, il est difficile de juger le résultat obtenue par l'algorithme, des indices numériques sont utilisés.

Une bonne

3.1 Indices de qualitée

Les indices utilisé dans ce TP seront :

- Indice de Dunn définit par :

$$D = \frac{S_{min}}{S_{max}} \tag{2}$$

  avec S_min la distance intracluster minimal, et S_max la distance iterclusster maximal. Cette indice doit être maximiser[1].
- Indice de Davies-Bouldin[2] :

$$D = \frac{1}{n} \sum_{i=1, i \neq j}^{n} max\{\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\} \tag{3}$$

  avec n nombre totale de clusters, σ_i est la moyenne des distances des éléments d'un clusters à leur centres c_i. $d(c_i, c_j)$ est la distance entre les centres des clusters i et j. Cette indexe doit être minimiser.
- l'indice RSQ[**?**] définit par :

$$D = \frac{D_n}{D_n + D_a} \tag{4}$$

  avec : D_n : inter-cluser distances & D_a : intra-cluster distances, est un indice universelles de qualité de partitionnement de donnée. Cet indice doit être proche de 1, sans pour autant le toucher (cas limite où chaque élément est centre de son cluster). L'idéale c'est d'approcher 1 avec un k minimale.

3.2 Choix de k

Trouver le k optimale n'est pas toujours simple, on remarque que si on augmente k, la qualité du résultat s'améliore, et est parfaite lorsque chaque point est centre de son cluster ($k = n$). Pour pallier à ce phénomène il est possible de rajouter un terme en fonction de k dans l'expression du critère (1) mais là encore son choix reste arbitraire[5].

En général, il n'existe pas d'indication sur le nombre le plus approprié de classes, et un "mauvais choix" pour la valeur de k conduira à un résultat sans rapport avec la réalité, et dont l'interprétation est difficile.

Une façon de choisir le k (qu'on appliquera ici au cas IRIS) est d'exécuter l'algorithme plusieurs fois pour chaque k sur un intervalle $[2, k_i]$, et de choisir le k qui donne les meilleurs indexes de qualité rechercher. On verra à travers le cas IRIS qu'un telle choix peut être abstrait et déconnecter par rapport à la réalité.

3.3 Initialisation des centres

Le k-means standard est très dépendant de la valeurs de départ des centres (Aléatoire dans notre algorithme), une mauvaise initialisation conduits à de mauvais résultats. En effet à chaque exécution correspond une solution différente de la première qui peuvent être très éloigner l'une de l'autre.

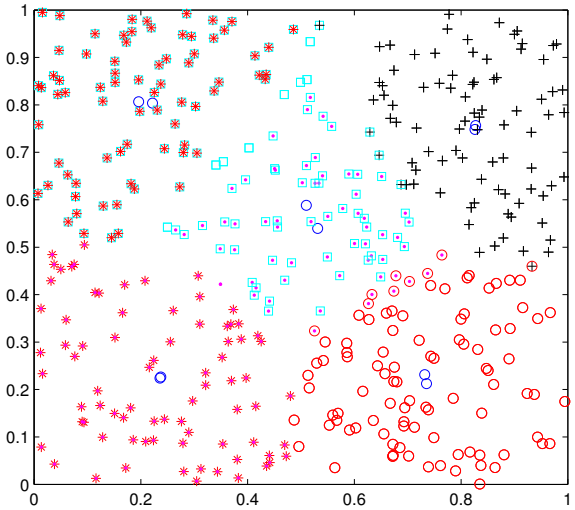

FIGURE 2 – Deux test sur la même base de donnée donne des centres (cercles en bleu) de clusters differents (proches néamoins). Certains éléments changent de cluster

Dans ce TP, les centres sont tous le temps initialisé aléatoirement en échantillonnant les données, on peut résumer 4 moyens d'initialisation :

- initialisations à partir des donnés : Choisir aléatoirement parmi les données un échantillon de centres pour initialiser l'algorithme.
- initialisations à partir de l'intervalle des données : Crées aléatoirement des vecteurs dans l'intervalle de variations des données.
- Pré-conditionnement des candidat : Enlevé les points les plus éloigner.
- Choix Algorithmique : Appliquer un algorithme de sélection sur les n échantillons et prendre le résultat comme centres de départ. Donne de bon résultats en générale

A travers l'exemple de la figure 2 on voit bien que, certains éléments change de groupe, cela confirme que l'algorithme n'est pas robuste vis à vis du choix initiales des centres. On note aussi que la distance entre les centres reste constante, leurs positions relatives aussi.

Une façon d'améliorer l'algorithme sur ce point et de répéter l'algorithme plusieurs fois et de mettre les éléments qui se retrouvent ensemble tous le temps dans le même cluster.

On obtiendra n ensemble de données chaque ensemble regroupe $Q(n)$ éléments. Les ensembles les plus représentatifs seront garder et leur centre de gravité deviennent les nouveaux centres. L'algorithme k-means sera alors exécuter à nouveau. On peut aussi exécuter le code plusieurs fois et de prendre la solution qui minimise le critère (1) (meilleur minima local).

3.4 Choix de la distance

J'ai tester sur deux différentes définition de distance entre u et v (deux vecteur d'un même espace E) :

- la distance euclidienne, définit par :

$$d(u,v) = \sqrt{\sum_{k=1}^{n}(u_k - v_k)^2} \tag{5}$$

- la distance de Manhattan, définit par :

$$d(u,v) = \sum_{k=1}^{n}|u_k - v_k| \tag{6}$$

Il existe d'autres notion de distance comme celle de Sebestyen ou de Mahalanobis, qui se basent sur la pondération de quelques un des constituants d'un vecteur par rapport aux autres.

Ce type de distance peuvent s'avérer utile dans les cas où certaines propriétés sont plus importantes que d'autres au sain d'une même classe.

4 Complexité algorithmique

Si on note t_e le temps de calcule de la distance entre deux points, et pour n échantillons de taille m, et pour k centres, la complexité en temps sera : $O(mknt_e)$, exécuter plusieurs fois la complexité devient $O(Imknt_e)$ pour I itérations. En effet la machine calcule n distances k fois pour un vecteur de taille m, et puis finalement elle recalcule les k centres (cela ce fait en une seul opération).

Pour la complexité en mémoire la machine doit garder les n échantillons en plus des k centres. La complexité sera donc $O(n + k)$.

La complexité augmente dans les cas où on refait le même algorithme plusieurs fois, pour dégager le meilleurs optima local possible.

5 Étude de la base I.R.I.S

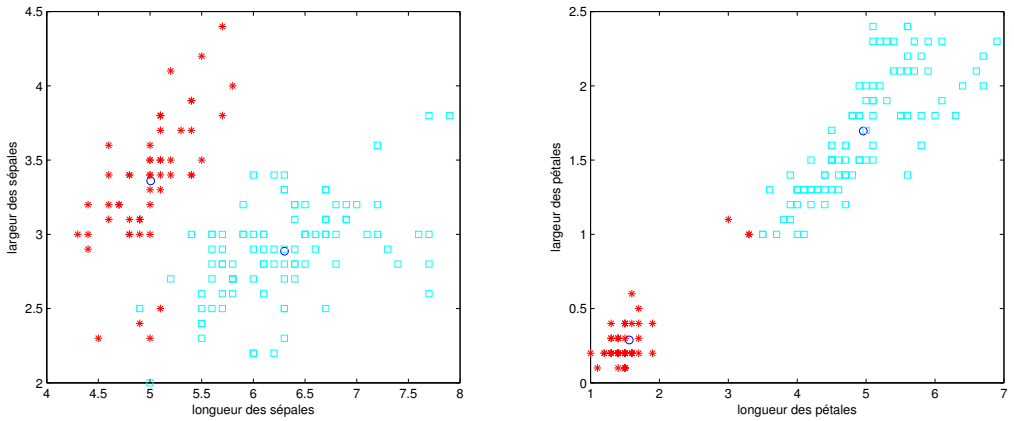

FIGURE 3 – Visualisation des données IRIS pour $k = 2$

Les données IRIS ont été introduit par Sir Ronald Fisher Aylmer (1936), qui a recueilli ses données pour distingué les fleurs d'Iris de la péninsule gaspésienne (Québec). Pour les trois types d'iris (Iris setosa, Iris virginica et Iris versicolor), et pour chaque fleur de son échantillon il a mesurer en cm quatre caractéristiques différentes, qui sont la longueur et la largeur des sépales et des pétales (Les colonnes du fichier iris.data sont ainsi ordonnés).

Dans le fichier `iris.data` fournie pour ce TP les 150 instances sont déjà classer en trois classes, chaque cinquantaine de données forme une même classe. Le but est d'arriver à cet même classification avec le k-means. Les résultat obtenue à travers mon algorithme avec la distance euclidienne et

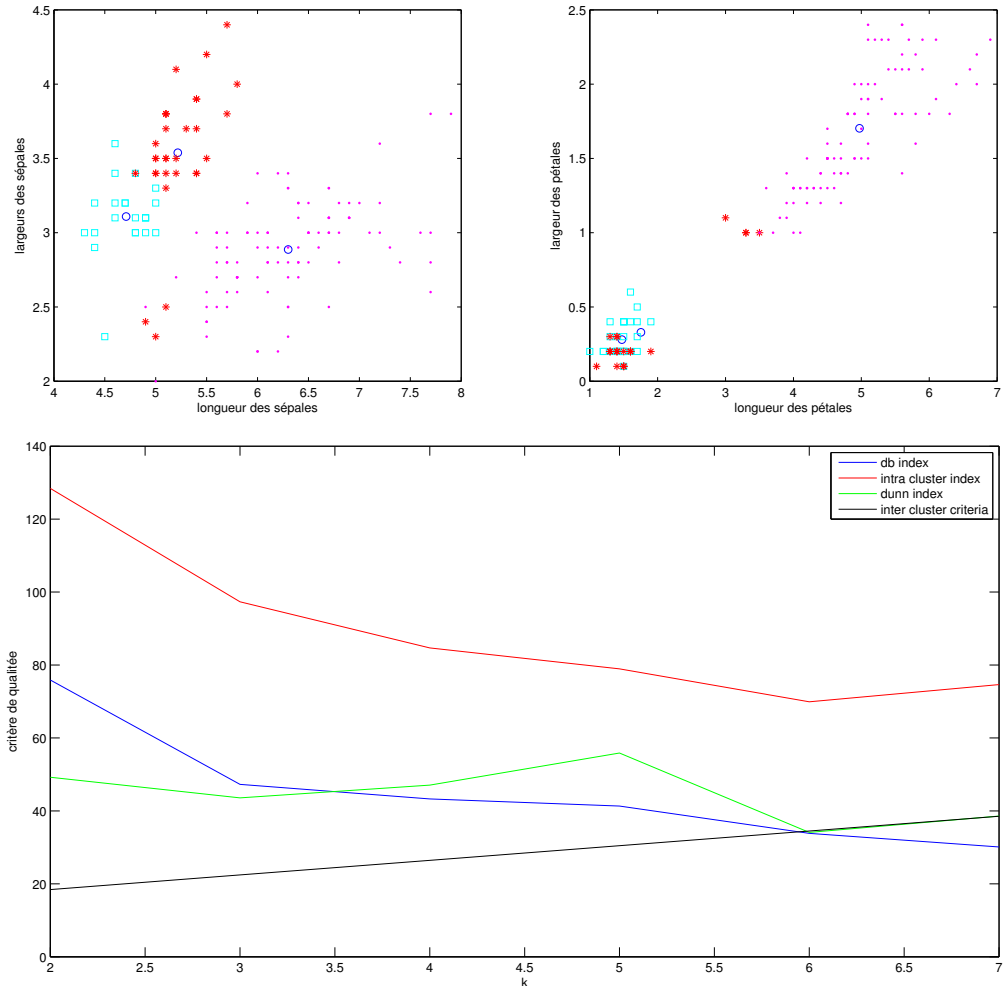

FIGURE 4 – (haut) Visualisation des données IRIS pour $k = 3$
(bas) Les indices obtenue pour différents k

un nombre de clusters égale à trois, montre que l'algorithme distingue la première classe et la deuxième (à 3 exceptions près), mais à du mal à distinguer la dernière classe de la deuxième (Figure 4).

Un k fixé à deux donne, à quelques exceptions près, le résultat attendu (Figure 3) en combinant les deux dernières classes en une seul, et ce quelque soit la distance utilisé. Un $k = 4$ donne des résultats instables à chaque exécution de l'algorithme.

L'illustration précédente montre les différents résultats obtenue pour différents k. L'indice de Dunn est maximisé pour : $k \in [2, 5]$, donc le meilleurs k dans ce cas serait 2 voir 3. L'indice de Dunn donne le résultat attendu par rapport

au choix de k. De même l'indice de Davies & Bouldin est minimiser pour $k \geq 3$, (on remarque la pente brusque qu'il y a entre $k = 2$ et $k = 3$).

La distance inter cluster, ne fait qu'augmenter, elle atteindra sa valeur finale pour $k = 150$, cas limite. La distance intra-cluster ne fait que diminuer elle atteindra 0 au même cas limite. Le meilleurs k étant celui qui rapproche l'indice RSQ définit précédemment de 1 sans y arriver.

Références

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

[2] Davies & Bouldin, 1979. Davies, D.L., Bouldin, D.W., (2000) A cluster separation measure. IEEE Trans. Pattern Anal. Machine Intell., 1(4), 224-227.

[3] MacQueen, J. B. (1967). "Some Methods for classification and Analysis of Multivariate Observations". 1. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281–297. MR0214227. Zbl 0214.46201. Retrieved 2009-04-07.

[4] Data Clustering : A Review, A.K. JAIN, Michigan State University, M.N. MURTY, Indian Institute of Science AND P.J. FLYNN, The Ohio State University. Full text here : tinyurl.com/6yjd37u

[5] Proposition d'une solution au problème d'initialisation cas du k-means by Z.Guellil et L.Zaoui. Université des sciences et de la technologie d'Oran MB, Université Mohamed Boudiaf USTO -BP 1505 El Mnaouer - ORAN - Algérie

[6] Edgar Anderson (1935). "The irises of the Gaspé Peninsula". Bulletin of the American Iris Society 59 : 2–5

[7] Data mining et statistique décisionnelle : l'intelligence des données, par Stéphane Tufféry

[8] Fisher, R.A. (1936). "The Use of Multiple Measurements in Taxonomic Problems". Annals of Eugenics. Full text : tinyurl.com/6kyk2ty

[9] Likas A. Vlassis M. & Verbeek J. The global k-means clustering algorithm, Pattern Recognition, 36, pp. 451- 461, 2003

[10] Jean-Pierre Nakache& Josiane Confais, Approche pragmatique de la classification