24 mai 2017

Caio VALENTE

Rochd MALIKI

Compte rendu TP2

MAJ INFO – UV2

413 : Algorithmique avancé

TELECOM Bretagne

Institut Mines-Télécom

Sommaire

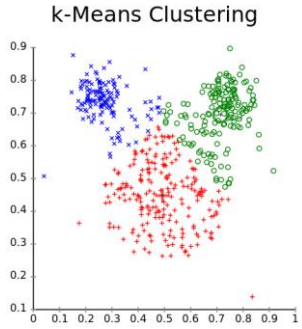
Table des matières

1. PRESENTATION DE L'ALGORITHME K-MEANS

1.1 HISTORIQUE :

Le terme « k-means » a été utilisé pour la première fois par James MacQueen en 19671, bien que l'idée originale ait été proposée par Hugo Steinhaus en 19572. L'Algorithme classiquea été proposé par Stuart Lloyd en 1957 à des fins de modulation d'impulsion codée, mais il n'a pas été publié en dehors des Bell Labs avant 19823. En 1965, E. W. Forgy publia une méthode essentiellement similaire, raison pour laquelle elle est parfois appelée « méthode de Lloyd-Forgy » 4. Une version plus efficace, codée en Fortran, a été publiée par Hartigan et Wong en 1975/1979.

1.2 PRINCIPES DE FONCTIONNEMENT :



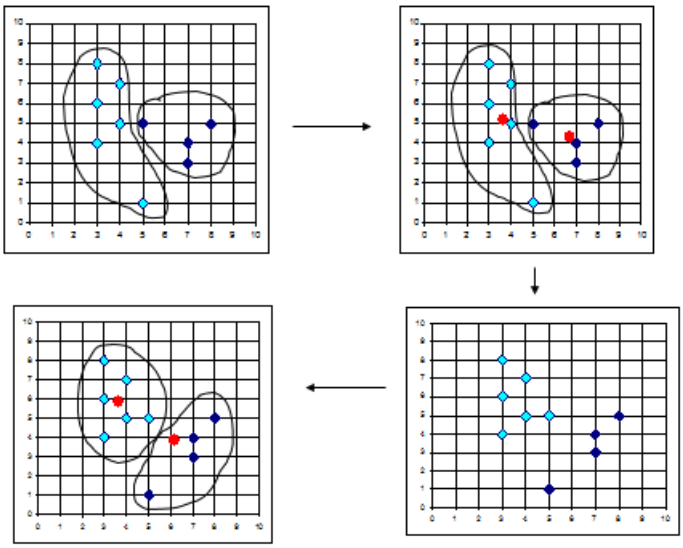
L'algorithme k-means permet de construire une collection d'objets similaires au sein d'un même groupe. Autrement dit, l'algorithme permet de classifier les objets tels que les objets de grande similarité sont dans un même groupe, et les objets de groupes différents sont de faible similarité.

- L'algorithme *k-means* est en 4 étapes :
  1. Choisir k objets formant ainsi k *clusters*(1)
  2. (Ré)affecter chaque objet O au cluster C_i de centre M_i tel que dist(O,M_i) est minimal
  3. Recalculer M_i de chaque cluster (le barycentre)
  4. Aller à l'étape 2 si on vient de faire une affectation

1.3 EXEMPLE :

On suppose qu'on a un ensemble de 10 données ayant deux attributs chacune. On représente ces attributs sur l'axe vertical et horizontal pour avoir une vue en deux dimension. On souhaite répartir ces données en deux groupes c'est à dire avoir deux classes de données en utilisant l'algorithme de k-means.

Pour cela on choisit deux centres (en rouge), alors on calcule la distance de chaque point aux deux centres et on affecte à chaque point la classe correspondante.



Lorsque nos points sont classés, on redéfinit les deux centres en calculant le barycentre des deux classes, et on refait l'opération avec les nouveaux centres.

2. PRESENTATION DU CODE

Notre algorithme doit être capable de :

- ✓ Générer aléatoirement les données
- ✓ Charger les données Iris
- ✓ Choisir des centres aléatoires à partir des données générées ou lies
- ✓ Sauvegarde des données
- ✓ Calcul de la distance entre deux objets
- ✓ Affectation des objets aux différents groupes
- ✓ Donner le bon choix du nombre de clusters
- ✓ Mettre à jour les centres à chaque fois afin de bien regrouper les objets
- ✓ Détecter le nombre d'erreurs fait au niveau des données iris

3. FONCTIONS ET COMPLEXITES

3.1 K-MEANS :

Fonction :

On fournit le nombre n de points, la dimension k de ces points et le nombre de clusters à générer. Les points seront gérés aléatoirement dans un intervalle prédéfini.

Puis, l'algorithme choisit aléatoirement les centres parmi les données.

Après cela, on fait une répétition. Dans cette répétition, on va classifier les points selon ses proximités aux centres. C'est-à-dire, chaque point sera placé dans le groupe qui a le centre le plus proche de ce point.

Puis, les barycentres de chaque groupe seront calculés et ensuite les centres des groupes seront redéfinis selon le critère de proximité du barycentre.

La répétition est arrêtée quand on fait une répétition sans changer aucun centre de groupe et sans changer le groupe d'aucune donnée ou quand on fait un total de 300 répétitions.

Ensuite, les données classifiés et les centres des groupes sont écrits dans ces respectifs fichiers.

La fonction retourne les données classifiées.

3.2 COMPLEXITE :

La complexité de l'algorithme est de l'ordre de O(i*n*d*k), où :

- i est le nombre d'itération de l'algorithme
- d est la complexité du calcul de la distance : elle dépend de la distance choisie, pour la distance euclidienne elle est de l'ordre de la dimension d'une donnée parmi les données
- k est le nombre de centres (clusters)
- n est le nombre de données

3.3 LES AUTRES FONCTIONS :

- **read_iris_data(filename)**

Charge les données d'un fichier csv (en format iris) appelé filename et retourne la liste correspondante.

- **write_data(datas, filename)**

  Écrit des données dans un fichier csv.
- **write_centers(centers, filename)**

  Écrit des centres dans un fichier csv.
- **generatepoints(n, dimension, min_ = 0, max_ = 1000)**

  Génère n points aléatoires dans un intervalle de valeurs spécifique
- **choseRandomicCenters(n, points)**

  Choisit n centres aléatoires à partir de points disponibles
- **euclideanDistance(point_1,point_2)**

Calculer la distance euclidienne entre deux points. Si les points ont des dimensions différentes, la plus petite dimension sera considérée dans le calcul

- **distance_iris(point_1,point_2)**

Calcule la distance euclidienne en prenant en compte les paramètres iris normalisés Euclidienne distance

- **nearestNeighbour(point, neighbours,iris=False)**

Trouver le voisin le plus proche d'un point

- **classificatePoints(points, centers,iris=False)**
  Regroupe les points selon leurs centres les plus proches
- **barycenter(points, groupNum)**
  Calcule le point barycentrique d'un groupe
- **pointsOfGroup(classifiedPoints, groupNumber)**
  Filtre la matrice de points par un groupe spécifique
- **calculateBaryCenters(points, numberOfCenters)**
  Calcule et retourne les barycentres de tous les groupes
- **updateCenters(points, centers,iris=False)**
  Recalcule les centres des groupes basé sur les barycentres de chaque groupe
- **iris_nbr_errors(points)**
  Calcule le nombre d'erreurs sur un appel d'algorithme k-means sur les données iris

4. VERSIONS ;

On peut avoir plusieurs version de l'algorithme en l'améliorant, en jouant sur ces variantes en particuliers :

4.1 LE CHOIX DES DISTANCES :

Nous avons constaté que la distance euclidienne n'est pas toujours une bonne distance, si nous manipulons des données de grandes dimensions en l'occurrence. Il est donc impératif de faire un choix de la distance qui va être exploité pour faire le clustering, ce choix dépend de la nature des données bien entendu. Pour ce faire il est recommandé de voir la bibliothèque *scipy* de python qui propose plusieurs type de distance, faire les test sur celle-ci et choisir celle qui permet un clustering efficace et rapide.

4.2 LA CONDITION D'ARRET :

L'arrêt de l'algorithme de la méthode des "k-means" se fait

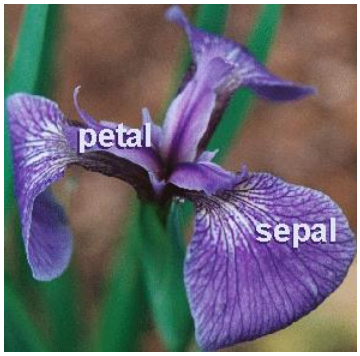
· Lorsque deux itérations successives conduisent à une même partition.

· Lorsqu'on fixe un critère d'arrêt tel que le nombre maximal d'itérations.

5. IRIS DATA :

Le jeu de données comprend 50 échantillons de chacune des trois espèces d'iris « Iris setosa, Iris virginica et Iris versicolor ». Quatre caractéristiques ont été mesurées à partir de chaque échantillon : la longueur et la largeur des sépales et des pétales, en centimètres. Sur la base de la combinaison de ces quatre variables, Fisher a élaboré un modèle d'analyse discriminante linéaire permettant de distinguer les espèces les unes des autres. Donc un échantillon du tableau de données iris, contient

[Longueur sépale, largeur sépale, longueur pétale, largeur pétale]



6. TESTS SUR LES DONNEES IRIS :

6.1 CHOIX DU NOMBRE DE CLUSTER :

Afin de choisir le nombre de classe permettant de donner la meilleure répartition des données, on trouve trois méthodes connues :

-**Elbow method**[2] : Rappelons que l'idée de base des méthodes de partitionnement, telle que k-means clusters, est de définir des groupes tels que la variance entre les données de chaque groupes et leurs centres est minimisée.

-**Average silhouette method :** calcule la moyenne des observations pour différentes valeurs de k. Le nombre optimal de clusters k est celui qui maximise la moyenne sur une plage de valeurs possibles pour k.

-Gap statistic method : compare la variation pour différentes valeurs de k avec leurs valeurs attendues sous la distribution de référence nulle des données, c'est-à-dire une distribution sans groupement évident.

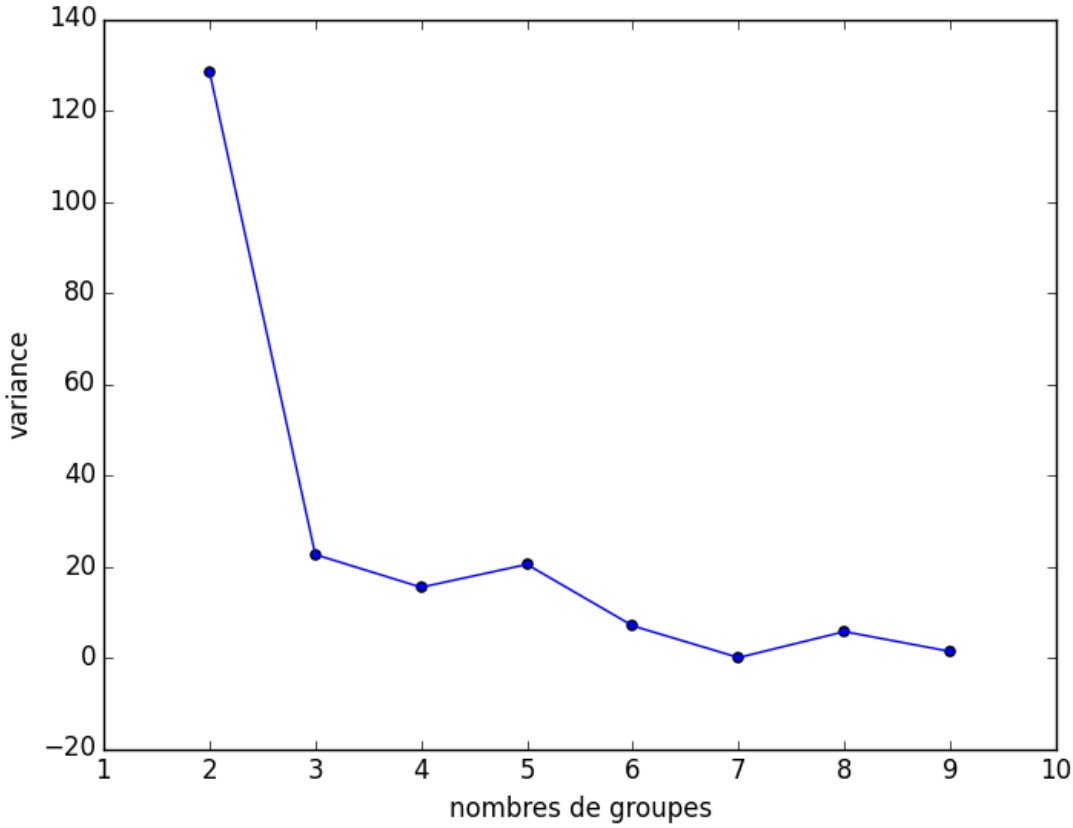
Pour ce faire nous avons choisi de tester l'heuristique **Elbow Method** sur notre algorithme, et en particulier sur les données Iris, pour appliquer cette méthode il faut calculer la variance suivante

$$V = \sum_{i=1}^{k} \sum_{x \in Ci} dist(Ci, x)$$

On doit choisir un certain nombre de centres de sorte que l'ajout d'un autre centre ne donne pas une meilleure modélisation des données. Plus précisément, si l'on trace la variance en fonction du nombre de groupes, le premier nombre groupe qui

ajouterait beaucoup d'informations (réduit largement de variance) est le nombre de groupe qui correspond à un meilleur *clustering*[3].

Sachant que le nombre de cluster idéal pour celui-ci est connu, nous avons obtenu le graph suivant :



Nous remarquons que nous avons bien « le coude » pour un nombre de groupes égale à 3.

6.2 DISTANCE APPLIQUEE SUR LES DONNEES IRIS

Après plusieurs essaies sur les données iris nous avons constaté que la distance euclidienne n'est pas un bon choix pour faire le clustering, nous avons choisi d'utiliser une version pondérée de la distance euclidienne pour qu'elle soit adaptée à la nature de celles-ci. Cela donne un résultat plus rapide et efficace que lorsqu'on utilise une distance euclidienne. Une amélioration au niveau des pondérations est envisageable.

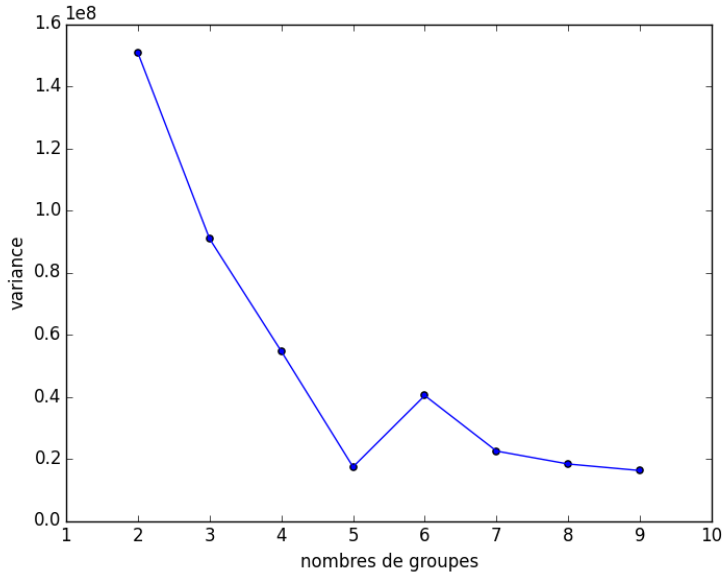
6.3 NOMBRES D'ERREURS :

Notre algorithme permet aussi de calculer le nombre d'erreurs faites sur le clustering des données iris en particulier puisque on connait à priori le groupement idéal, « Iris-setosa, Iris-versicolor, Iris-virginica », le nombre d'erreurs varie en fonction de la distance choisie et du choix du point d'arrêt. En utilisant la distance euclidienne pondérée, Il s'est avéré que la condition d'arrêt que nous avions choisi n'est pas très satisfaisante, puisque nous avons fait tourner notre programme 300 itérations et le résultat obtenue a donné largement moins d'erreurs qu'avec la condition d'arrêt, nous proposons ainsi une condition d'arrêt plus robuste qui prend en considération aussi les changements lors de la mise à jour des centres.

7. INTERET DE L'ELBOW METHOD :

Comme nous l'avons vu le test de l'Elbow method sur les données Iris correspond bien à la valeur attendue, or ce n'est pas le cas toujours, c'est-à-dire le nombre de cluster n'est pas toujours connu. Autrement dit, cette méthode va permettre de déterminer le bon cluster si nous hésitons sur combien de cluster on veut grouper nos données.

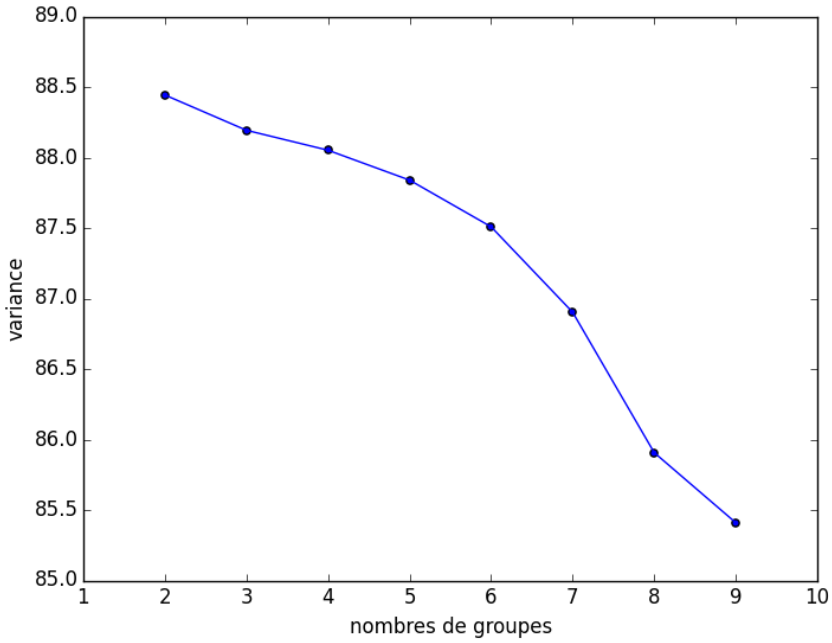
Pour ce faire, nous allons générer 1000 données aléatoires de dimension 4, avec des valeurs d'attribut entre 0.0 et 1000.0 et on applique l'elbow method sur ces données :



On remarque que le bon choix de cluster pour ces points en particulier est 5.

Donc pour bien classer ces données il nous faut 5 groupes. L'elbow method dans ce cas nous a aidés à choisir le meilleur nombre de groupes.

On applique de nouveau l'elbow method mais cette fois-ci sur 1000 données de 1 seule dimension et des valeurs d'attributs entre 0.0 et 1.0



Nous constatons que si la densité de données est très grande sur un intervalle (1000 points sur [0,1] en l'occurrence) la method d'elbow ne permet pas de déduire le bon nombre de clusters.

Comme toute autre méthode, cette méthode a des limitations, reste discutable et ne marche pas à tous les coups.

8. CONCLUSION :

Ce TP était une occasion de découvrir l'algorithme K-means et ses domaines d'application. En essayant plusieurs méthodes, nous avons pu remarquer la différence des résultats en fonction des critères choisis.

.

Pour ce qui est des choix de calcul de distance, nos études ne mènent pas à des résultats qui nous permettraient d’éliminer une méthode par rapport à une autre.

Les conditions d’arrêt sont à leurs tours importantes. En effet, pour un critère basé sur le nombre d'itérations défini, nous risquerons d'avoir un mauvais résultat si l'on était devant des données complexes. L'arrêt sur une tolérance donnée s'avère efficace pour limiter le nombre d'itérations, toutefois, il faut bien prendre le soin de choisir le bon nombres de groupes.

Références bibliographique :

https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set

www.math-info.univ-paris5.fr/~lomn/Cours/DM/Material/.../clustering.PPT

https://fr.wikipedia.org/wiki/K-moyennes

https://fr.wikipedia.org/wiki/Iris_(jeu_de_donn%C3%A9es)

http://www.memoireonline.com/10/08/1603/m_classification-population-categories-socio-economiques-methodologie-application17.html

http://www.sthda.com/english/wiki/determining-the-optimal-number-of-clusters-3-must-known-methods-unsupervised-machine-learning

Glossaire :

(1)Cluster : groupe
(2) Elbow method : La méthode du coude
(3)Clustering : groupement

Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
+33 (0)2 29 00 11 11
www.telecom-bretagne.eu