POLYTECH
LYON

[K-Means (Centres mobiles)]

[UE:Analyse et fouilles de données]

Fait par :

Malasri Janumporn

Rayene Hakoume

Table des matières

I. Introduction

Notre Shiny application est pour classer les données du dataset Iris, qui est intégré dans le package "datasets", en moyen du K-Means. La structure du dataset "Iris" est :

'data.frame': 150 obs. of 5 variables:
$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
$ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
$ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
$ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
$ Species : Factor w/ 3 levels "setosa","versicolor","virginica": 1 1 1 1 1 1 1 1 1 1 ...

K-Means

Le clustering K-Means (MacQueen 1967) est l'un des algorithmes d'apprentissage automatique non supervisés les plus couramment utilisés pour partitionner un ensemble de données donné en un ensemble de k groupes (c'est-à-dire k clusters), où k représente le nombre de groupes prédéfinis par l'utilisateur. Il classe les objets dans plusieurs groupes, de sorte que les objets au sein d'un même groupe soient aussi similaires que possible (c'est-à-dire une similitude intra-classe élevée), tandis que les objets de différents groupes sont aussi dissemblables que possible (c'est-à-dire similitude inter-classe faible). Dans le clustering k-means, chaque cluster est représenté par son centre (c'est-à-dire centroïde) qui correspond à la moyenne des points affectés au cluster.

Algorithme K-Means en général

Les étapes sont résumés comme suit:

1. Spécifier le nombre de clusters (K) à créer (par l'utilisateur)

2. Sélectionnez de manière aléatoire k objets du jeu de données comme centres ou moyens de cluster initiaux.
3. Assigne chaque observation à son centroïde le plus proche, en fonction de la distance euclidienne entre l'objet et le centroïde.
4. Pour chacun des k clusters, mettez à jour le centroïde du cluster en calculant les nouvelles valeurs moyennes de tous les points de données du cluster. Le centoïde d'un K_ième cluster est un vecteur de longueur p contenant les moyennes de toutes les variables pour les observations dans le K_ième cluster; p est le nombre de variables.
5. Réduisez de manière itérative le total dans la somme du carré. Autrement dit, itérez les étapes 3 et 4 jusqu'à ce que les affectations de cluster cessent de changer ou que le nombre maximal d'itérations soit atteint. Par défaut, le logiciel R utilise 10 comme valeur par défaut pour le nombre maximal d'itérations.

***Remarque* :**

La fonction kmeans (package "stats") intrégré dans R. L'algorithme de Hartigan et Wong (Hartigan 1979) est utilisé par défaut. Note que certains auteurs utilisent-signifie faire référence à un algorithme spécifique plutôt que la méthode générale: le plus souvent l'algorithme donné par MacQueen mais parfois celui donné par Lloyd (Lloyd 1957) et Forgy (Forgy 1965). L'algorithme de Hartigan--Wong fait généralement un meilleur travail que l'un ou l'autre, mais en essayant plusieurs démarrages aléatoires (nstart > 1) est souvent recommandé. Dans de rares cas, lorsque certains des points sont extrêmement proches, l'algorithme peut ne pas converger à l'étape « Quick-Transfer », signalant un avertissement (et retour 'ifault = 4'). L'arrondissement léger des données peut être souhaitable dans ce cas.

Pour faciliter l'exploration programmatique,k = 1 est autorisé, notamment retour du centre et withinss (Vecteur de la somme intra-cluster des carrés, un composant par cluster).

À l'exception de la méthode Lloyd--Forgy,les clusters seront toujours renvoyé si un nombre est spécifié. Si une matrice initiale de centres est fournie, il est possible que Aucun point ne sera le plus proche d'un ou de plusieurs centres, ce qui est actuellement le cas une erreur pour la méthode Hartigan--Wong.

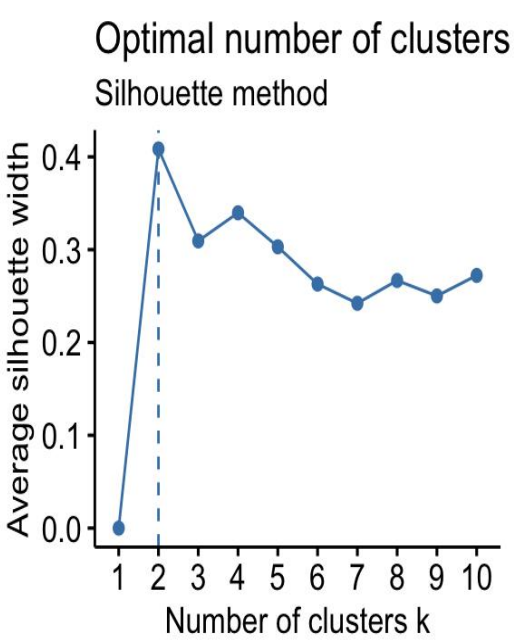
Silhouette Algorithme pour déterminer la valeur optimale de k

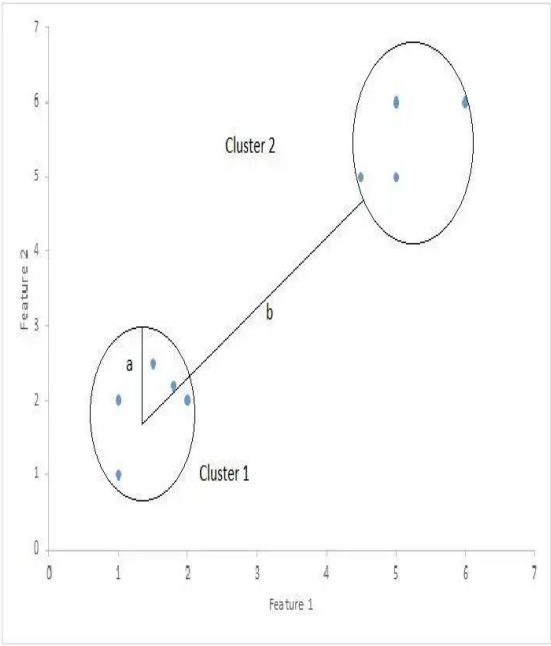
Pour chaque observation i, la largeur de la silhouette est défini comme suit:
Mettre a(i) = dissimilarité moyenne entre i et tous les autres points du groupe auquel i appartient (si i est la seule observation dans son cluster,s(i) = 0, sans autres calculs).
Pour tous les autres clusters C mettre :
d(i,C)= moyenne dissimilarité de i avec toutes les observations de C.
Le plus petit d'entre eux est:

b(i) = minC d(i,c), et peut être vu comme la dissemblance entre i et son « voisin » cluster, c'est-à-dire le plus proche auquel il n'appartient pas. Finalement:

$$\mathbf{s(i)} = \frac{\mathbf{b(i)-a(i)}}{\mathbf{max(a(i),b(i))}}$$

Observations avec une grande (presque 1) sont très bien groupé, un petit (autour de 0) signifie que l'observation se situe entre deux clusters et les observations avec un s(i) négatif sont probablement placés dans le mauvais cluster.





Elbow méthode

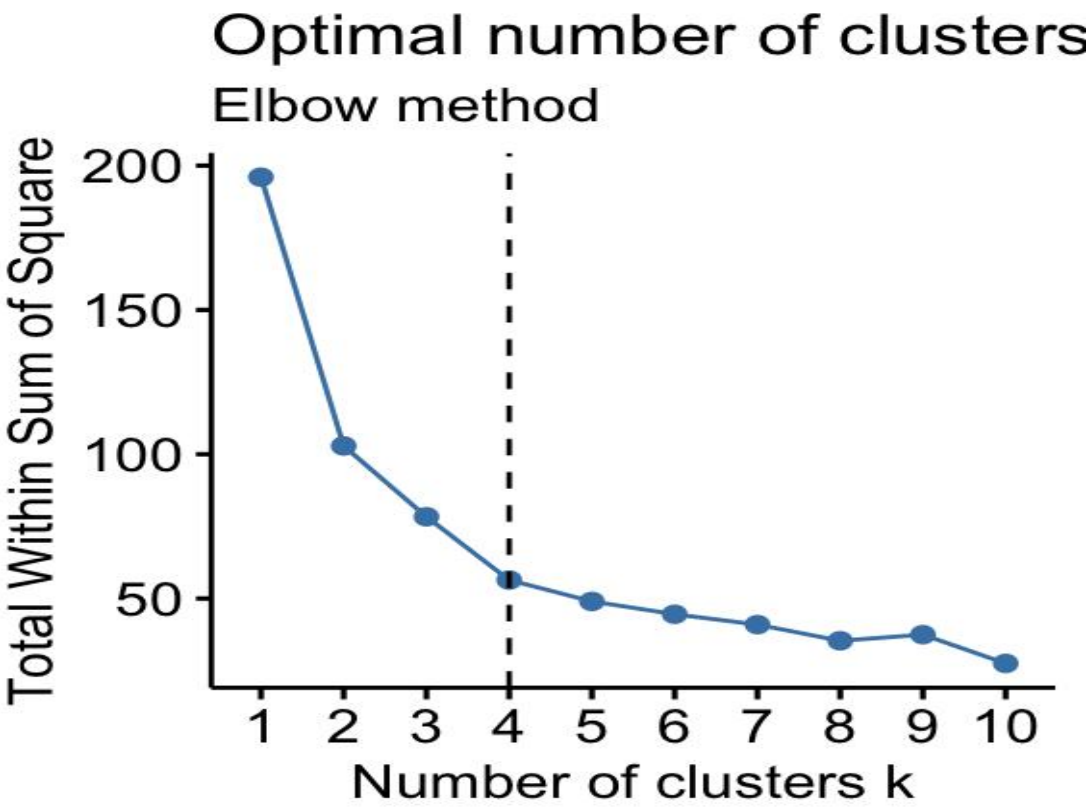
Le clustering K-Means est de définir des clusters de manière à minimiser la variation totale intra-cluster (ou la somme totale du carré au sein du cluster (WSS)) soit minimisée. Le WSS total mesure la compacité du clustering et nous voulons qu'il soit aussi petit que possible.

La méthode Elbow considère le WSS total en fonction du nombre de clusters: Il faut choisir un certain nombre de clusters afin que l'ajout d'un autre cluster n'améliore pas beaucoup mieux le WSS total.

Le nombre optimal de clusters peut être défini comme suit :

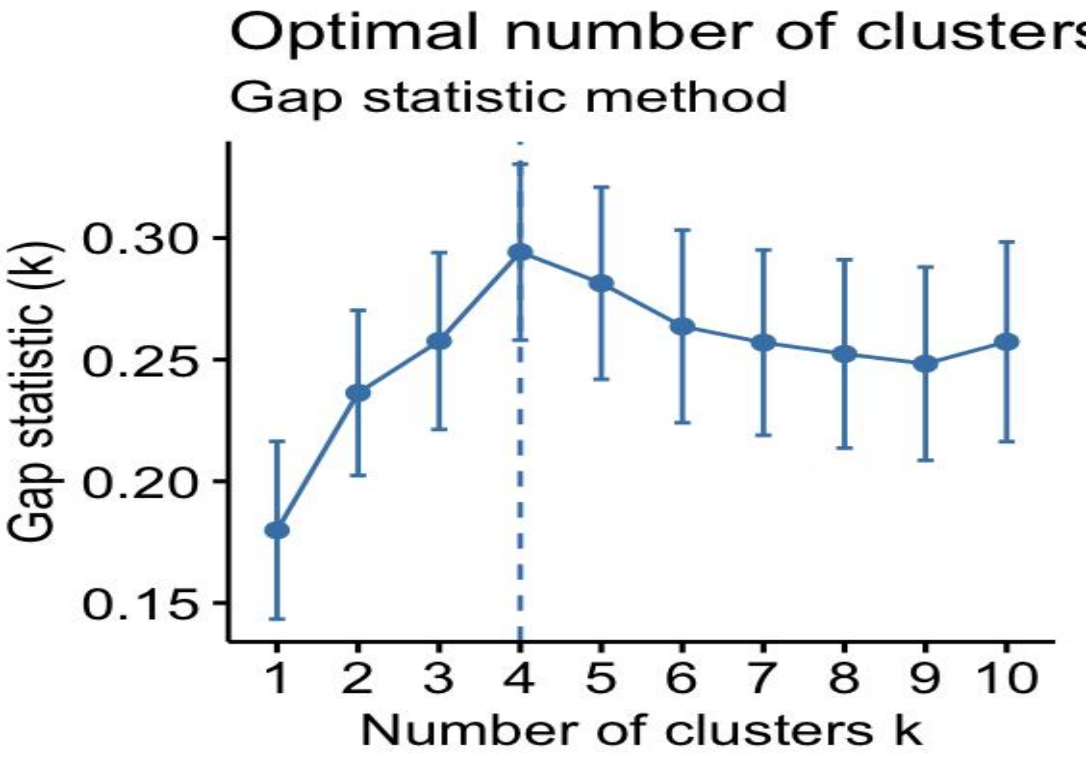
1. Algorithme de clustering de calcul (par exemple, clustering k-means) pour différentes valeurs de k. Par exemple, en faisant varier k de 1 à 10 clusters.
2. Pour chaque k, calculez la somme totale du carré (wss) à l'intérieur du cluster.
3. Tracer la courbe de wss en fonction du nombre de clusters k.

4. L’emplacement d’une courbure (genou) dans la placette est généralement considéré comme un indicateur du nombre approprié de clusters.



Méthode de statistique des écarts (gap statistique)

La statistique de l’écarta été publiée par R. Tibshirani, G. Walther et T. Hastie (Université de Standford 2001). L’approche peut être appliquée à n’importe quelle méthode de clustering. La statistique de l’écart compare le total au sein de la variation intra-cluster pour différentes valeurs de k avec leurs valeurs attendues sous la distribution de référence nulle des données. L’estimation des clusters optimales sera la valeur qui maximise la statistique de l’écart (c'est-à-d. qui donne la statistique de l’écart le plus important). Cela signifie que la structure de regroupement est loin de la distribution uniforme aléatoire des points.



Installation

Cette application a besoin de packages installés suivants :

- **FactoMineR** est pour calculer ACP en 2 dimensions (2 components) afin de visualiser des données en graphique

- **factoextra** est pour visualiser des données en graphique de 2D, ici on utilise pour afficher les classes obtenues de l'algorithme K-means et afficher un nombre de classes optimal par raaport les trois méthodes i.e. silhouette, within sum of squares(wss), gap statique(gap_stat).

- **shiny** est pour créer une interface graphique dynamique pour l'application. Si vous n'avez pas encore ces packages, tapez une commandes suivant avant de démarrer l'application :

install.packages("nom de package")

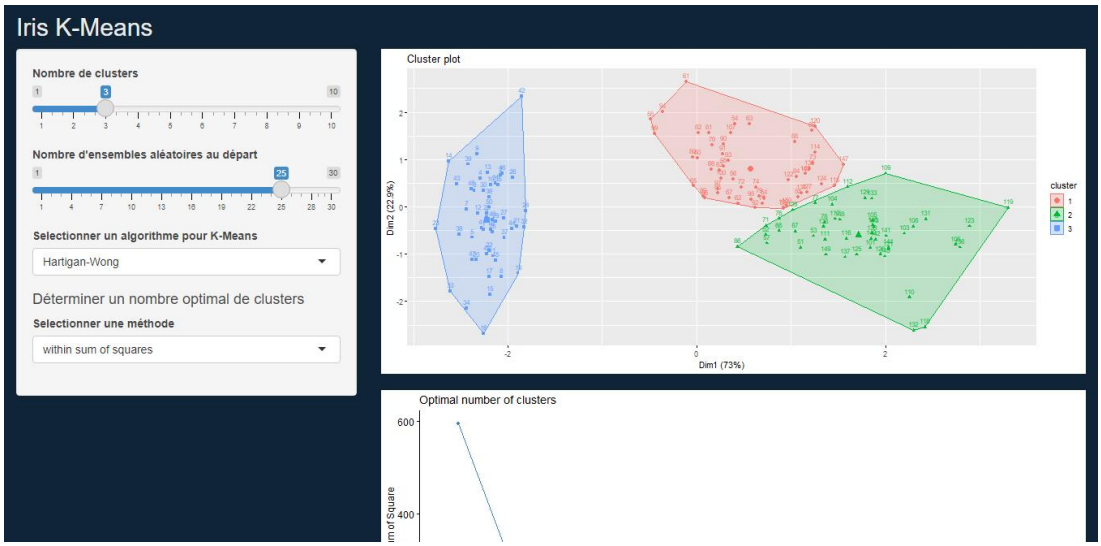
R version 4.2.2

Implémentation

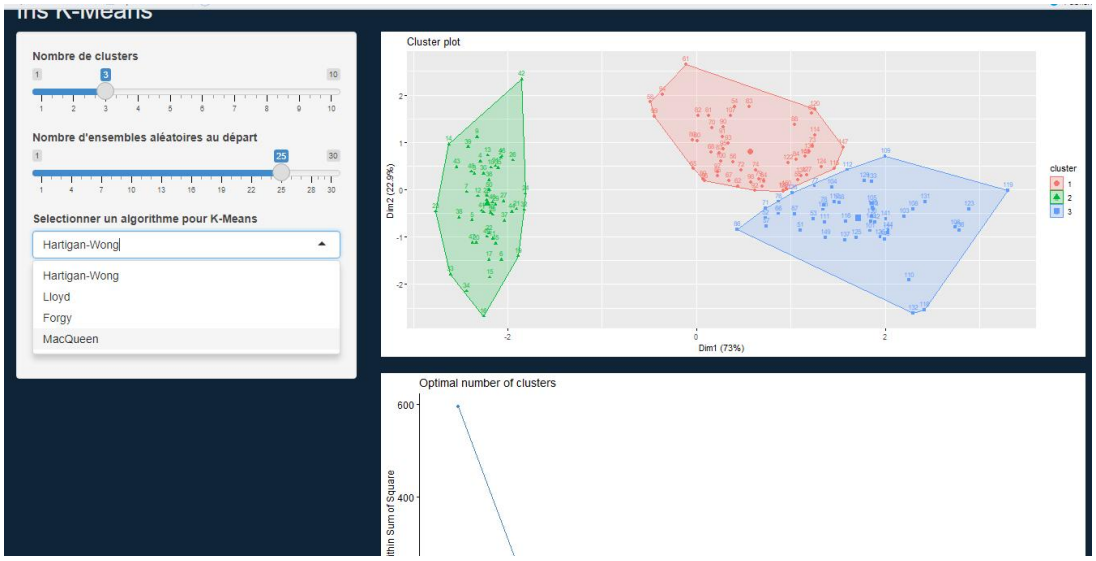
Au lancement de l'application, un utilisateur peut paramétrer :

- Un nombre de clusters (3 par défaut),les choix sont entre 1 et 10 inclusif.

- Un nombre d'ensembles aléatoires au départ (25 par défaut), les choix sont entre 1 et 30 inclusif.



- Un alogorithme pour K-Means ("Hartigan-Wong" par défaut), les choix sont "Hartigan-Wong", "Lloyd", "Forgy" et "MacQueen".



• Une méthode pour déterminer un nombre optimal de clusters ("within sum of squares" ou wss par défaut). les choix sont "silhouette", "withun sum of squares" et "gap statique".

La fonction kmeans :

kmeans(x, centers, iter.max = 10, nstart, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"), trace=FALSE)

Arguments :

• x : Matrice numérique de données, ou un objet qui peut être contraint à une telle matrice (telle qu'un vecteur numérique ou une trame de données avec tous colonnes numériques).

• centers : Soit le nombre de clusters, disons k, ou un ensemble de centres de clusters initiaux (distincts). Si un nombre, un ensemble aléatoire des lignes (distinctes) sont choisies comme centres initiaux.Ici, on met 3 par défaut.

• iter.max : Nombre maximal d'itérations autorisées, ici on prends 10.

• nstart : Si c'est un nombre, combien d'ensembles aléatoires devrait être choisi?, ici on prends 25 par défaut.

• algorithm : Caractère: peut être abrégé. il y a "Hartigan-Wong", "Lloyd", "Forgy", "MacQueen". Ici om met "Hartigan-Wong" par défaut.

• trace : Nombre logique ou entier, actuellement utilisé uniquement dans le méthode par défaut ("Hartigan-Wong"): si positive (ou true), le traçage des informations sur la progression de l'algorithme est produit. Des valeurs plus élevées peuvent produire plus d'informations de traçage, ici on prends FALSE.

Visualisation

Les résultats de K-Means sont interprétés en graphique par des fonctions dans le package "factoextra" comme suivant :

1- Un graphique de 2D montre la distribution des données dans chaque cluster défini en utilisant la fonction "fviz_cluster()".

fviz_cluster() utilise la fonction PCA() si un nombre de variables > 2 pour visualiser les observations en points de 2D. Une ellipse est dessinée autour de chaque cluster.

fviz_cluster(object, data,geom = c("point","text"), labelsize=8, ellipse.type = "convex", ggtheme = theme_gray())

Arguments

- object : Un objet de classe « partition » créé par les fonctions pam(), clara() ou fanny() dans le package cluster; « kmeans » [dans le paquet de statistiques]; « dbscan » [dans le paquet FPC]; « Mclust » [dans mclust]; « HKMums », « eclust » [en factoextra]. Les valeurs possibles sont également n'importe quel objet de liste avec des données et des composants de cluster (ex. : object = list(data = mydata, cluster = myclust)).
- data : Données qui ont été utilisées pour le clustering. Requis uniquement lorsque object est une classe de kmeans ou dbscan.
- geom : Texte spécifiant la géométrie à utiliser pour le graphique. Les valeurs autorisées sont la combinaison de c("point", "texte"). Utilisez "point" (pour afficher uniquement points); "texte" pour n'afficher que les étiquettes; c(" point ", " texte ") pour afficher les deux types.
- labelsize : Taille de police pour les étiquettes
- ellipse.type : Caractère spécifiant le type d'image. Les valeurs possibles sont « convexe », « confiance » ou types pris en charge par l'inclusion de l'un des types c("t", "norm", "Euclide")
- ggtheme : Fonction, nom du thème ggplot2. La valeur par défaut est theme_pubr(). Les valeurs autorisées incluent les thèmes officiels de ggplot2 : theme_gray(), theme_bw(), theme_minimal(), theme_classic(), theme_void(), ...

2- Un graphique de 2D montre un nombre optimal de clusters en utilisant la fonction "fviz_nbclust()".
fviz_nbclust() détermines et visualise le nombre optimal de clusters utilisant différentes méthodes : silhouette, within sum of squares, gap statistique. Les méthodes de partitionnement, telles que le clustering k-means, nécessitent le utilisateurs pour spécifier le nombre de clusters à générer.

fviz_nbclust(x,FUNcluster,method = c("silhouette", "wss", "gap_stat"),k.max=10)

Arguments

- x : Matrice numérique ou trame de données
- FUNcluster : Une fonction de partitionnement. Les valeurs autorisées incluent kmeans, pam, clara et hcut (pour le clustering hiérarchique).
- method : Méthode à utiliser pour déterminer le nombre optimal de clusters.
- k.max : Le nombre maximum de clusters à prendre en compte doit être d'au moins deux. Ici on prends 10.

Réferences

- Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency vs interpretability of classifications. Biometrics, 21, 768--769.

- Hartigan, J. A. and Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. Applied Statistics, 28, 100--108. 10.2307/2346830.

- Lloyd, S. P. (1957, 1982). Least squares quantization in PCM. Technical Note, Bell Laboratories. Published in 1982 in IEEE Transactions on Information Theory, 28, 128--137.

- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, eds L. M. Le Cam & J. Neyman, 1, pp.281--297. Berkeley, CA: University of California Press.

- R. Tibshirani, G. Walther, and T. Hastie (2001). Estimating the number of clusters in a data set via gap statistic.Journal Statist Soc.B,63 Part 2,pp 411-423, Standford University, USA.