Université Paris Descartes
UFR de Mathématiques et Informatique

Classification Master 1 Ingénierie Mathématique
Examen partiel du 12 mars 2019 - durée 1h30

On écrira toutes les commandes demandées dans l'éditeur en prenant soin d'enregistrer le fichier à la fin de la séance (peu importe le nom et l'emplacement du fichier). On pourra aussi enregistrer les résultats graphiques dans des fichiers pdf (Menu "Export" puis "Save as pdf").

Exercice 1

Soit le tableau de données suivant (trois observations et trois variables binaires)

	X^1	X^2	X^3
e_1	1	0	1
e_2	1	1	1
e_3	0	1	0

1. Rappeler la définition de la dissimilarité de Dice telle que vue en cours.
2. Calculer à la main les dissimilarité de Dice entre les trois individus.
3. Entrer le tableau de données en R et appliquer la fonction dist.binary de la librairie `ade4` pour le même critère de dissimilarité. Pourquoi ne retrouve-t-on pas les mêmes valeurs ? (bien regarder l'aide de la fonction dist.binary). Appliquer la modification nécessaire aux valeurs obtenues avec R afin de vérifier le calcul fait à la question 2.

Exercice 2

Le jeu de données `elec88` de la librairie `ade4` concerne les résultats des élections présidentielles de 1988 en France, suivant la localisation géographique. Le tableau sur lequel on va travailler est `elec88$tab` qui donne les proportions de vote par candidat dans chaque département.

1. Quel choix de dissimilarité, parmi celles vues en cours vous semble la plus adaptée pour ces données ? Calculer avec R la matrice de dissimilarité correspondante à l'aide de la fonction `dist.quant`.
2. Réaliser une classification ascendante hiérarchique sur ces données avec la dissimilarité précédente et la stratégie d'agrégation de Ward, puis afficher le dendrogramme.
3. Quel nombre de classes correspond à un saut maximal du critère d'agrégation ? Calculer la partition pour ce nombre de classes.
4. Réaliser à présent une classification avec la méthode des k-moyennes, pour le nombre de classes choisi précédemment, et comparer la partition obtenue avec la partition obtenue par la CAH.
5. Dans les données étudiées, les observations correspondent à des départements, qui ont des nombres d'habitants très variables. On pourrait chercher à prendre en compte cet aspect pour améliorer la classification de ces données, en cherchant à pondérer les observations par le nombre d'habitants du département. Sans chercher à le réaliser en pratique, comment selon vous pourrait-on modifier la méthode de classification hiérarchique pour faire jouer cette pondération ? Même question pour la méthode des k-moyennes.

Dans la suite de cet exercice, on va chercher à déterminer un nombre de classes optimal pour la méthode des k-moyennes, en utilisant une méthode similaire à celle utilisée pour la CAH.

6. Que représentent les valeurs `totss, withinss, tot.withinss, betweenss` contenues dans la sortie de la méthode `kmeans` ? Quel rapport y a-t-il entre ces valeurs et la stratégie d'agrégation de Ward ? Ceci donne en fait un critère de comparaison de la qualité des deux partitions obtenues (par CAH et par k-moyennes). Laquelle est la meilleure selon ce critère ?
7. La méthode `kmeans` peut donner des résultats différents à chaque test. Pourquoi ? Vérifier sur quelques essais que les partitionnements obtenus peuvent être différents, de même que les valeurs de la variable `tot.withinss` correspondantes.
8. Pour éviter cet aléa, on peut jouer sur le paramètre `nstart` de `kmeans`. Que réalise précisément la fonction `kmeans` lorsque ce paramètre est fixé à une valeur plus grande que 1 ? Pourquoi ceci permet-il de réduire l'aléa ?
9. Vérifier expérimentalement que la valeur par défaut de ce paramètre `nstart=10` permet d'obtenir une valeur stable de `tot.withinss`, alors que c'est beaucoup moins vrai si on choisit `nstart=1`. Pour cela, on comparera l'écart-type et la moyenne empiriques obtenues pour cette valeur sur 100 essais.
10. A présent on va faire varier le nombre de classes k : pour toutes les valeurs possibles de k, appliquer la méthode `kmeans` sur les données et enregistrer les valeurs de la variable `tot.withinss` obtenues dans un vecteur.
11. Trouver ensuite le nombre de classes k optimal, correspondant à un saut maximal de la valeur de cette variable, de façon similaire à la procédure utilisée pour la CAH. Retrouve-t-on le même nombre de classes que précédemment ?