Université Paris Descartes
UFR de Mathématiques et Informatique

Classification Master 1 Ingénierie Mathématique
Examen final du 9 mai 2019

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

Partie A (à faire sur feuille)
Soit le tableau de données suivant (9 individus, deux variables quantitatives et une classe binaire)

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
X^1	1	1	2	3	4	5	5	6	7
X^2	3	7	6	4	2	3	6	5	3
Y	0	0	1	0	1	0	1	1	1

1. Représenter les observations sur un graphique avec X^1 et abscisse et X^2 en ordonnée.
2. Soient les deux nouvelles observations suivantes : $X_a = (4, 4)$ et $X_b = (3, 5)$. Placer ces points sur le graphique, puis pour chacune de ces deux observations, donner ses classes estimés par la méthode des k-plus proches voisins pour la distance euclidienne, avec $k = 1$ et avec $k = 3$.

Partie B (à faire avec R)
On cherche à évaluer expérimentalement le phénomène de sur-apprentissage pour la méthode des k-plus proches voisins, sur des données simulées simples. Les commandes R suivantes permettent de générer et de représenter graphiquement des données quantitatives avec deux variables et une classe binaire.

```
N = 200
X1 = runif(N)
X2 = runif(N)
Y = X2 + 0.1*rnorm(N) < 3*X1*(X1-.5)*(X1-1)+X1
plot(X1,X2,col=Y+1)
donnee = data.frame(X1,X2,Y)
```

3. Recopier et exécuter les commandes R, et expliquer comment sont construites ces données.
4. Séparer les données en données apprentissage datatrain et test datatest de même taille.
5. On considère la méthode des k-plus proches voisins avec la distance euclidienne et construite sur les données datatrain. Pour chacune des valeurs de k entre $k = 1$ et $k = 20$, calculer avec R les classements des données datatest et datatrain, et évaluer les taux d'erreurs correspondants. On enregistrera les valeurs obtenues dans deux vecteurs errTrain et errTest de taille 20.
6. Afficher sur un même graphique les deux courbes représentants les deux taux d'erreur en fonction de k. Pour quelles valeurs de k est-on en situation de sur-apprentissage ? Quelle valeur de k vous semble optimale ?

Exercice 2

Les données morphosport de la librairie ade4 contient des informations morphologiques sur 153 athlètes pratiquant différents sports. On va chercher à prédire le sport pratiqué en fonction des caractéristiques morphologiques.

1. Charger les données puis les observer pour comprendre comment elles sont organisées.
2. Former un jeu de données unique en concaténant les tableaux afin d'avoir la variable "sport" en dernière colonne.
3. Séparer les données en base d'apprentissage et en base de test, puis utiliser la méthode CART avec les paramètres par défaut. Afficher l'arbre obtenu, puis classer les données test et calculer le taux d'erreurs.
4. Répéter les commandes de la question précédente un grand nombre de fois afin d'obtenir une meilleure évaluation du taux d'erreur pour la méthode CART.
5. Rappeler le principe de la méthode des forêts aléatoires.
6. Utiliser la méthode des forêts aléatoires pour classer les données test, et calculer à nouveau le taux d'erreur. La méthode des forêts aléatoires est-elle plus performante sur ces données ?

Exercice 3

Les données cnc2003 de la librairie ade4 contiennent des informations sur la fréquentation des salles de cinéma pour chacun des départements français en 2003.

1. Charger les données puis les observer pour comprendre comment elles sont organisées. Supprimer les deux dernières colonnes (département et région) du jeu de données.
2. Normaliser les données avec la commande scale. Pourquoi cette normalisation est-elle justifiée ?
3. Utiliser la méthode des K-moyennes et avec les paramètres par défaut pour partitionner les données en $K = 6$ groupes.
4. Répéter les opérations précédente une deuxième fois, puis comparer les partitionnements obtenus. La méthode donne-t-elle des partitionnements stables ?

Afin de stabiliser et d'accélérer la méthode des K-moyennes, une méthode récente appelée *K-means++* a été proposée. Il s'agit de remplacer l'initialisation aléatoire des centres par une initialisation spécifique obtenue par l'algorithme suivant :

— Choix d'un premier centre C^1 aléatoire parmi les observations $X^1, \ldots, X^N$.

— Pour $k = 1$ à $K - 1$: on suppose que k centres ont été choisis, et on choisit le $(k+1)$-ième centre de la façon suivante :

  — On calcule les distances euclidiennes $d_{il} := \|X^i - C^l\|$ entre tous les individus X^i, $1 \leqslant i \leqslant N$ et tous les centres déjà choisis C^l, $1 \leqslant l \leqslant k$.

  — On calcule les distances minimales $\delta_i = \min\{d_{il}, 1 \leqslant l \leqslant k\}$ pour chaque individu i, puis les probabilités $p_i = \frac{\delta_i}{\sum_{i=1}^N \delta_i}$.

  — On choisit le nouveau centre C^{k+1} en effectuant un tirage aléatoire parmi les individus, avec des probabilités p_i.

La fonction suivante permet de réaliser une étape de l'algorithme : le calcul des distances euclidiennes entre les individus contenus dans le tableau X et des centres contenus dans le tableau C :

```
DistEucl = function(X,C)
{
  N = nrow(X)
  K = nrow(C)
  D2 = matrix(NA,N,K)
  for(i in 1:N)
  {
    D2[i,] = rowSums((X[i,]-C)**2)
  }
  sqrt(D2)
}
```

5. Recopier cette fonction et la commenter.
6. Écrire une fonction InitKmeanspp(X,K) qui à partir d'un tableau de données quantitatives X, et d'un nombre de classes K, effectue l'initialisation des centres par la méthode K-means++ et renvoie les centres C (sous forme de tableau $N \times K$) ainsi obtenus.
7. Appliquer la méthode des K-moyennes pour $K = 6$ sur les données en remplaçant l'initialisation aléatoire des centres par l'initialisation *K-means++*.
8. Évaluer expérimentalement l'amélioration apportée par *K-means++* en terme de stabilité (les partitions obtenues sont-elles plus stables) et en terme de rapidité (l'algorithme des k-moyennes converge-t-il en plus ou moins d'itérations ?)