Université Paris Descartes
UFR de Mathématiques et Informatique

Classification
Examen du 03/01/2017

Préalables : Ouvrir Rstudio, puis entrer les commandes suivantes :

```
library(class)
library(rpart)
install.packages("mlbench")
library(mlbench)
install.packages("randomForest")
library(randomForest)
```

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).

Exercice 1

Soit le tableau de données suivant (5 individus, 1 variable quantitative X et 2 groupes $Y = 0$, $Y = 1$) :

	e_1	e_2	e_3	e_4	e_5
X	5	-1	0	3	2
Y	1	0	0	1	0

1. Appliquer à la main l'algorithme des k-plus proches voisins sur les données, successivement pour $k = 1$, $k = 3$ puis $k = 5$. Calculer le taux d'erreurs d'apprentissage à chaque fois.
2. Calculer dans les 3 cas précédents le taux d'erreur obtenu par validation croisée "leave-one-out". On pourra utiliser R ou le faire directement à la main. Comparer et commenter ces taux d'erreurs avec ceux obtenus précédemment.

Exercice 2

Le tableau suivant récapitule les conditions dans lesquelles une concentration d'ozone élevée ou faible a été observée. L'objectif de l'exercice est de prédire la concentration d'ozone pour un jour donné en fonction de diverses conditions climatiques. On souhaite donc construire une règle de classification permettant de prédire le label Y="Concentration d'ozone" à partir des variables X_1="Température", X_2="Vent", X_3="Ensoleillement" et X_4="Humidité".

1. Rappeler le principe de la méthode CART pour construire des arbres de classification.
2. À partir de ces données, construire à la main selon la méthode CART l'arbre de classification T. On ne segmentera pas les nœuds composés de 3 ou de moins de 3 individus. Justifier soigneusement chacune des étapes. On pourra transformer le tableau en un tableau binaire.
3. Tracer l'arbre final T obtenu. Combien de feuilles contient T ? Attacher à chaque feuille un label.
4. Donner la règle de classification $\hat{t}$ induite par l'arbre T.
5. Calculer le taux d'erreur d'apprentissage de $\hat{t}$.

X_1	X_2	X_3	X_4	Y
Elévée	Non	Faible	Forte	Elevée
Normale	Oui	Fort	Faible	Elevée
Elévée	Non	Fort	Forte	Elevée
Elévée	Non	Faible	Faible	Elevée
Normale	Non	Fort	Faible	Normale
Normale	Oui	Fort	Forte	Normale
Elévée	Oui	Faible	Faible	Normale
Elévée	Oui	Fort	Forte	Normale

6. Quelles sont les variables qui semblent déterminer une concentration élevée d'ozone ?
7. On récolte les informations suivantes (X_1 = Normale, X_2 = Non, X_3 = Fort, X_4 = Forte), y a-t-il un risque que la concentration d'ozone soit élevée ?

Exercice 3

Les données Glass contiennent différentes analyses chimiques relevées sur des morceaux de verre de différents types. Le but est de parvenir à prédire le type de verre en fonction de ses caractéristiques chimiques.

1. Charger les données avec la commande data(Glass), puis les observer pour comprendre comment elles sont organisées.
2. Séparer les données en données d'apprentissage et données de test, en choisissant 150 individus tirés au hasard pour constituer les données d'apprentissage.
3. Construire un arbre de classification avec la méthode CART sur les données d'apprentissage, en développant l'arbre jusqu'au bout. Afficher l'arbre obtenu, puis utiliser cet arbre pour classer les données test, et afficher la table de contingence entre les vraies classes et les classes obtenues par classification. Enfin, calculer le taux d'erreurs par cette méthode.
4. Reprendre la question précédente en choisissant cette fois les options par défaut de l'algorithme CART dans R (ce qui ne développe pas l'arbre jusqu'au bout). La méthode est-elle plus performante ? Pourquoi ?
5. Pour évaluer plus précisément la performance, on va procéder à une validation croisée par blocs : choisir un ordre aléatoire des individus, puis classer en formant 21 blocs de 10 individus plus un bloc de 4 individus. Les individus de chaque bloc sont classés en choisissant tous les autres comme données d'apprentissage. Déterminer le taux d'erreurs pour la méthode CART (avec options par défaut).

Enfin on va comparer les performances de la méthode CART sur ces données avec celles de l'algorithme des forêts aléatoires. L'algorithme des forêts aléatoires est implémenté dans la fonction randomForest de la librairie randomForest. Elle s'utilise avec la même syntaxe que la fonction rpart de R.

6. Rappeler le principe de l'algorithme des forêts aléatoires.
7. Reprendre les questions 4 et 5 en utilisant la méthode des forêts aléatoires. Obtient-on des meilleurs résultats qu'avec la méthode CART ? Pourquoi ? Quelle étape de la méthode CART a été oubliée ici et pourrait améliorer les performances de classification ?