Université Paris Descartes
UFR de Mathématiques et Informatique

Classification
Examen du 03/01/2017 - Correction

Exercice 1

1. — pour $k = 1$, on classe chaque individu dans la classe de son plus proche voisin, qui est lui-même, donc chaque individu est classé dans sa propre classe : e_1 et e_4 sont classés dans le groupe 1, e_2, e_3 et e_5 dans le groupe 0. L'erreur d'apprentissage est nulle (c'est toujours le cas pour $k = 1$).

   — pour $k = 3$, on regarde les 3 plus proches voisins :

   — pour e_1 : $X = 5$, donc les trois plus proches sont $X = 5$, $X = 3$, $X = 2$, c'est-à-dire e_1, e_4 et e_5 qui ont pour classes $Y = 1$, $Y = 1$, $Y = 0$ respectivement. La classe 1 est majoritaire, donc on classe dans le groupe 1.

   — pour e_2, les trois plus proches sont e_2, e_3, e_5, de classes $Y = 0$ pour les 3. Donc on classe dans 0.

   — pour e_3, les trois plus proches sont e_3, e_2, e_5, de classes $Y = 0$ pour les 3. Donc on classe dans 0.

   — pour e_4, les trois plus proches sont e_4, e_5, e_1, de classes $Y = 1$, $Y = 0$, $Y = 1$. Donc on classe dans 1.

   — pour e_5, les trois plus proches sont e_5, e_4, e_3, de classes $Y = 0$, $Y = 1$, $Y = 0$. Donc on classe dans 0.

   On voit que le classement est le même, et l'erreur d'apprentissage est nulle (mais ce n'est pas forcément le cas en général pour $k = 3$).

   — pour $k = 5$, on regarde les 5 plus proches voisins. Comme il n'y a que 5 individus en tout, ces 5 voisins seront toujours e_1, e_2, e_3, e_4 et e_5. Donc chaque e_i est classé dans le même groupe, qui est le groupe majoritaire, c'est-à-dire le groupe 0. On fait donc 2 erreurs de classement ; le taux d'erreur d'apprentissage est de 40%.

2. La validation "leave-one-out" consiste à classer chaque individu en l'excluant de la base d'apprentissage. Pour la méthode des plus proches voisins, ça revient donc à considérer les plus proches voisins d'un individu parmi tous les autres.

   — pour $k = 1$: on classe chaque individu dans la classe de son plus proche voisin parmi les autres, donc :

   — e_1 est classé dans la classe de e_4, c'est-à-dire en 1,

   — e_2 est classé dans la classe de e_3, c'est-à-dire en 0,

   — e_3 est classé dans la classe de e_2, c'est-à-dire en 0,

   — e_4 est classé dans la classe de e_5, c'est-à-dire en 0,

   — e_5 est classé dans la classe de e_4, c'est-à-dire en 1,

   Le taux d'erreurs est ici de 40%.

   — pour $k = 3$:

— les 3 plus proches voisins de e_1 sont e_4, e_5, e_3, de classes 1, 0, 0, donc on classe en 0,

— les 3 plus proches voisins de e_2 sont e_2, e_3, e_5, de classes 0, 0, 0, donc on classe en 0,

— les 3 plus proches voisins de e_3 sont e_2, e_5, e_4, de classes 0, 0, 1, donc on classe en 0,

— les 3 plus proches voisins de e_4 sont e_5, e_1, e_3, de classes 0, 1, 0, donc on classe en 0,

— les 3 plus proches voisins de e_5 sont e_4, e_3, et e_1 ou e_2. On peut choisir par convention de prendre e_1 (indice le plus petit), ce qui donne les classes 1, 0 et 1. Donc on classe en 1.

On a ici un taux d'erreurs de 60% (ou 40% si on choisit la convention inverse pour classer e_5).

— pour $k = 5$: en excluant l'individu considéré de la base d'apprentissage, on n'a plus que 4 individus, donc on doit se restreindre aux 4 plus proches voisins, qui sont à chaque fois tous les autres.

— pour e_1, les voisins sont donc e_2, e_3, e_4, e_5, de groupes 0, 0, 1, 0, donc on classe dans 0.

— pour e_2, les voisins sont e_1, e_3, e_4, e_5, de groupes 1, 0, 1, 0, donc on a une indétermination. On peut décider de classer dans le groupe d'indice minimal, c'est-à-dire 0.

— pour e_3 et e_5, on a aussi une indétermination ; avec la même convention on les classera dans 0.

— enfin e_4 est classé dans 0

Au final avec notre convention on classe ici tout le monde dans le même groupe 0, don on a un taux d'erreurs de 40%. Ceci dit avec la convention inverse on se retrouverait avec un taux d'erreurs de 100%.

La taille des données est ici beaucoup trop petite pour qu'on puisse faire de vrais commentaires. On peut malgré tout remarquer que les taux d'erreurs par validation croisée sont plus importants que les taux d'erreurs d'apprentissage, ce qui est classique (la validation croisée permet d'avoir une meilleure estimation de l'erreur réelle de la méthode, tandis que l'erreur d'apprentissage est trop optimiste).

Exercice 2

1. La méthode CART consiste à construire récursivement un arbre binaire tel que :
   — la branche initiale correspond à l'ensemble de tous les individus à classer,
   — à chaque étape, on sépare les individus en 2 sous-ensembles suivant la valeur d'une seule variable observée,
   — le choix de cette variable doit maximiser l'hétérogénéité des classes $Y = 0$ ou $Y = 1$ dans les 2 sous ensembles obtenus. Pour cela on peut utiliser l'indice de Gini.
2. — première étape : comparons les indices de Gini pour les 4 variables :

— si on choisit X_1, on aura pour le groupe X_1 = Normale : 2 individus dans la classe Y = Normale, et 1 dans Y = Elevée. L'indice de Gini est donc $I(X_1 = \text{Normale}) = 1 - (2/3)^2 - (1/3)^2 = 0.44$ pour ce groupe. Pour le groupe X_1 = Elevée : 2 individus dans la classe Y = Normale, et 3 dans Y = Elevée. L'indice de Gini est donc $I(X_1 = \text{Elevée}) = 1 - (2/5)^2 - (3/5)^2 = 0.48$. L'indice de Gini moyenné est donc de $(3/8) * 0.44 + (5/8) * 0.48 = 0.465$.

— si on choisit X_2, on aura $I(X_2 = \text{Non}) = 1 - (1/4)^2 - (3/4)^2 = 0.375$ et $I(X_2 = \text{Oui}) = 1 - (3/4)^2 - (1/4)^2 = 0.375$. L'indice moyenné est de 0.375.

— si on choisit X_3, on aura $I(X_3 = \text{Faible}) = 1 - (1/3)^2 - (2/3)^2 = 0.44$ et $I(X_3 = \text{Fort}) = 1 - (3/5)^2 - (2/5)^2 = 0.48$. L'indice moyenné est de 0.465.

— si on choisit X_4, on aura $I(X_4 = \text{Faible}) = 1 - (2/4)^2 - (2/4)^2 = 0.5$ et $I(X_4 = \text{Forte}) = 1 - (2/4)^2 - (2/4)^2 = 0.5$. L'indice moyenné est de 0.5.

On voit que le meilleur choix est X_2 (indice moyenné minimal).

— deuxième étape : pour la branche X_2 = Non, on se retrouve avec les données suivantes :

X_1	X_3	X_4	Y
Élevée	Faible	Forte	Élevée
Élevée	Fort	Forte	Élevée
Élevée	Faible	Faible	Élevée
Normale	Fort	Faible	Normale

On voit tout de suite qu'en choisissant la variable X_1, on se retrouvera avec un groupe de 3 individus de classe Y = Normale et un groupe d'un seul individu de classe Y = Élevée. Par conséquent ce choix est forcément optimal (les indices de Gini valent 0), et il est inutile de tester les autres possibilités.

Pour la branche X_2 = Oui, on se retrouve avec les données suivantes :

X_1	X_3	X_4	Y
Normale	Fort	Faible	Elevée
Normale	Fort	Forte	Normale
Elévée	Faible	Faible	Normale
Elévée	Fort	Forte	Normale

Comparons les indices de Gini pour les 3 variables :

— si on choisit X_1, on aura pour le groupe X_1 = Normale : 1 individu dans la classe Y = Normale, et 1 dans Y = Elevée. L'indice de Gini est donc $I(X_1 = \text{Normale}) = 1 - (1/2)^2 - (1/2)^2 = 0.5$ pour ce groupe. Pour le groupe X_1 = Elevée : 2 individus dans la classe Y = Normale, l'indice de Gini est donc $I(X_1 = \text{Elevée}) = 0$. L'indice de Gini moyenné est donc de $(2/4) * 0.5 + (2/4) * 0 = 0.25$.

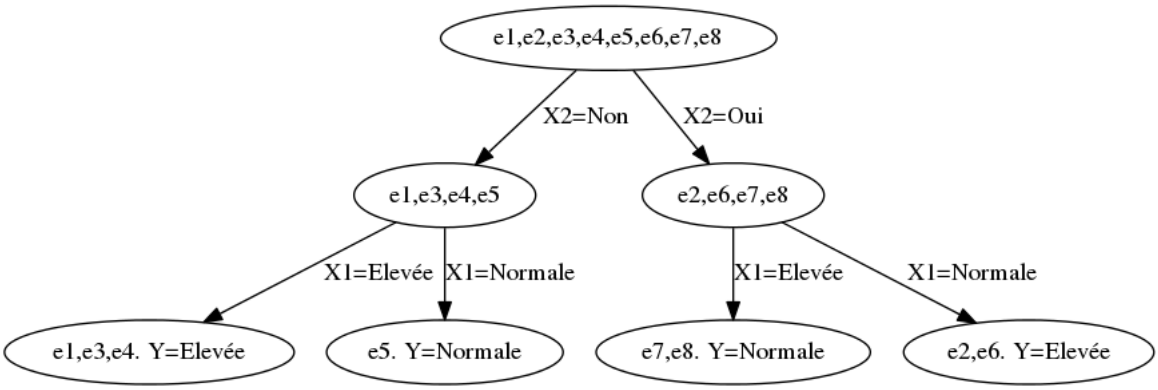
— si on choisit X_3, on aura $I(X_3 = \text{Faible}) = 0$ et $I(X_3 = \text{Fort}) = 1 - (1/3)^2 - (2/3)^2 = 0.44$. L'indice moyenné est de $(1/4) * 0 + (3/4) * 0.44 = 0.33$.

— si on choisit X_4, on aura $I(X_4 = \text{Faible}) = 1 - (1/2)^2 - (1/2)^2 = 0.5$ et $I(X_4 = \text{Forte}) = 0$. L'indice moyenné est de 0.25.

On peut donc choisir X_1 ou X_4. Choisissons X_1.

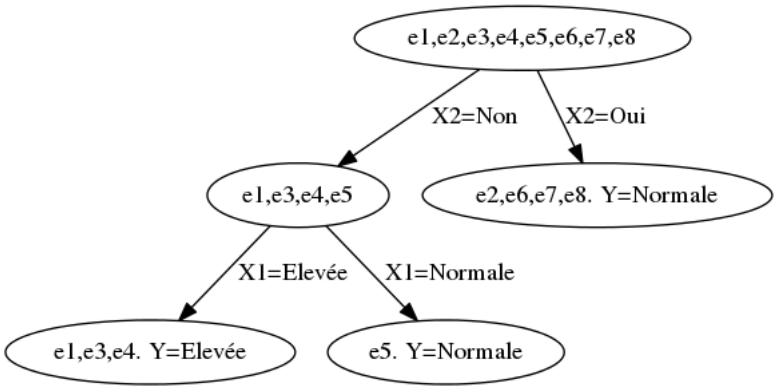
On arrête le développement de l'arbre ici car toutes les branches ont à présent moins de 3 individus.

3. L'arbre final obtenu est :



Cet arbre comporte 4 feuilles.

Remarque : pour la dernière feuille, contenant les individus e_2 et e_6, on a un individu de chaque classe, donc l'attribution de la classe est arbitraire. On a choisi ici d'attribuer la classe Y = Elevée, mais si on avait fait le choix inverse, alors on se serait retrouvé avec deux feuilles aboutissant à la même classe à droite, et du coup on aurait pu directement simplifier l'arbre en enlevant la deuxième étape à droite :



Cet arbre comporte 3 feuilles.

4. Pour l'arbre à 4 feuilles ci-dessus la règle de classification $\hat{t}$ est :

$$\hat{t}(X) = \begin{cases} \text{Élevée si } (X_2 = \text{Non et } X_1 = \text{Élevée}) \text{ ou si } (X_2 = \text{Oui et } X_1 = \text{Normale}) \\ \text{Normale sinon.} \end{cases}$$

5. Pour l'arbre à 4 feuilles ci-dessus, seul e_6 est mal classé. Le taux d'erreur d'apprentissage est donc de $1/8 = 12.5\%$.
6. On a utilisé uniquement les variables X_1 et X_2, donc ces deux variables semblent déterminantes pour la concentration d'ozone. Cependant on avait fait un choix arbitraire lors de la construction de l'arbre ; on aurait pu aussi utiliser X_4.
7. D'après la règle de classification obtenue, on a X_2 = Non et X_1 = Normale, donc on classe dans Y = Normale. On conclue qu'il n'y a pas de risque que la concentration d'ozone soit élevée.

Exercice 3

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 ?