Statistique en Grande Dimension et Apprentissage - TD/TP 5

Ce TD/TP a pour but d'illustrer le chapitre 5 sur les arbres binaires de décision, le bagging, le boosting et les forêts aléatoires.

Exercice 1. On souhaite comparer l'indice de Gini, l'erreur de classification et l'entropie.

1. Calculez ces quantités dans le cas binaire ($\{0,1\}$) en notant p la proportion de type 1.
2. Représentez les graphes associés (on pourra utiliser `R`).
3. Considérons maintenant le cas d'une variable Y à valeurs dans un ensemble à 3 éléments 0, 1 et 2. Calculez les quantités (Gini,…) lorsque $p_0 = 0.7, p_1 = 0.15, p_2 = 0.15$ et $p_0 = 0.7, p_1 = 0.25, p_2 = 0.05$. Que constatez-vous ? Comment interprétez-vous ce résultat ?
4. Pour une variable à m classes, calculez l'indice de Gini dans le cas équiprobable.
5. Montrez que l'indice de Gini au noeud t, noté $G(t)$ satisfait :

$$G(t) = 1 - \mathbb{P}(T_1 = T_2)$$

   où (T_1, T_2) désigne un couple de variables aléatoires indépendantes et de loi $\mu = (\hat{p}_{t,k})_{k=1}^{m}$.
6. L'entropie est la mesure "phare" de l'hétérogénéité. Elle représente aussi l'instabilité d'un système (en physique notamment). On peut montrer qu'elle majore la distance en *variation totale* de μ à la loi uniforme sur $\{1, \dots, m\}$. Il s'agit de l'inégalité de Pinsker.
   (a) Notons P et Q deux probabilités sur $\{1, \dots, m\}$. La distance en variation totale de P à Q est définie par :

$$d(P,Q)_{TV} = \frac{1}{2}\sum_{k=1}^{m} |P(k) - Q(k)|.$$

   La dissemblance de Kullback-Lleibler de P par rapport à Q est définie par :

$$D(P||Q) = \sum_{k=1}^{m} P(k) \log P(k)Q(k).$$

   Montrez que

$$D(P||Q) \geq \frac{1}{2} d(P,Q)_{TV}^2.$$

   (b) Appliquer ce résultat lorsque $P = \mu$ et Q est la loi uniforme sur $\{1, \dots, m\}$.

Exercice 2 (Bootstrap)**.** 1. Soit $Z_1, \dots, Z_q$ des variables aléatoires de même loi telles que $\mathrm{Var}(Z_1) = \sigma^2$ et telles que $\mathrm{Corr}(Z_i, Z_j) = \rho$ (indépendant de i et j). Montrez que

$$\mathrm{Var}\left(\frac{1}{q}\sum_{k=1}^{q} Z_i\right) = \rho\sigma^2 + \frac{1-\rho}{q}\sigma^2.$$

En appliquant ce résultat au bagging, discuter des conditions importantes pour que le bagging soit efficace.

2. Montrez que la probabilité que l'observation j n'appartienne pas à un échantillon bootstrap est égale à $(1 - \frac{1}{n})^n$. En déduire la proportion asymptotique d'observations n'appartenant pas à l'échantillon bootstrap.

Exercice 3 (Illustration méthode CART sur des données spam). Cet exemple est issu du mémoire de M1 de K. Eveilleau et A. Tanguy.

1. Charger la librairie `rpart`. Consulter l'aide de ce package.
2. Diviser l'échantillon en une partie `train` et une partie `test` (1/4 de la population) aléatoirement.
3. Construction de l'arbre de décision :

```
spam.rpart=rpart(formula = yesno .,method="class",control=rpart.control(cp=0.001), data=spam7)
```

4. A l'aide de la fonction `plotcp`, affichez l'évolution de l'erreur de prédiction (plus précisément, le pourcentage relatif à celle d'origine).
5. Quel rôle joue c_p ? Faites-le varier.
6. Affichez l'arbre maximal :
```
library(rpart.plot)
prp(spam.rpart,extra=1)
```
7. Pour obtenir l'arbre élagué, utilisez la fonction `prune`.
```
prune(spam.rpart,cp=??)
```
8. Evaluation des performances : sur l'échantillon d'entraînement, puis sur l'échantillon test sur l'arbre optimal, puis sur l'arbre maximal.

Exercice 4 (Illustration Bagging/Random Forests/Boosting sur des données spam). Diviser l'échantillon en une partie test (1/3) et une partie entraînement (2/3).

1. Bagging.
   (a) Utilisez la fonction `bagging` sur l'échantillon d'entraînement.
   ```
   bagging(yesno   .,coob = TRUE,data = spam.train,nbagg = 500).
   ```
   On pourra éventuellement faire varier le nombre d'arbres en jeu (attention au temps de calcul).
   (b) Affichez l'objet.
   (c) Estimer l'erreur de prédiction sur l'échantillon test.
   (d) Là encore, on pourra regarde l'évolution (pour $nbagg = 100, 200, 300, 400, 500$ par exemple).
2. Random Forests.
   (a) Charger la librairie Random Forests : `library(randomForest)`. Consulter l'aide.
   (b) Utilisez la fonction `randomForest` avec l'échantillon d'entraînement (pour $ntree = 100, 200, 300, 400, 500$ par exemple).
   (c) Affichez l'objet (ou faites un `summary`).
   (d) Estimer l'erreur de prédiction sur l'échantillon test. Conclusions
   (e) Tracez le graphe d'importance des variables à l'aide de la fonction `varImpPlot`.
3. Boosting.
   (a) Chargez la librairie `gbm`.
   (b) Utilisez la fonction `gbm`..
   (c) Mêmes questions que précédemment.
   (d) Affichage des variables d'importance.

Exercice 5. Testez les outils précédents sur une base de données "Optical Recognition of Handwritten Digits", `http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits`.