TP: Bootstrap et modèle linéaire

Résumé

Exemple d'utilisation de l'échantillonnage bootstrap pour le modèle linéaire.

1 Échantillonnage aléatoire simple

L'étude de grandes bases de données nécessite d'extraire un sous-ensemble par un échantillonnage alétoire simple dans la base afin de réaliser des analyses ou mettre au point des programmes sur un échantillon représentatif de taille raisonnable. Ce sondage est très simple à réaliser en R ; il est utilisé ici pour diviser l'échantillon en deux parts aléatoires : échantillon d'apprentissage et échantillon test. Le générateur est initialisé avec une semence spécifique à chaque (binôme) étudiant pour construire des échantillons différents.

```
# Utiliser trois  chiffre au hasard (xxx)
# comme initialisation du générateur
set.seed(xxx)
npop=1000
# tirage de 200 indices sans remise
testi=sample(1:npop,200)
# Liste des indices restant qui n'ont pas été tirés
appri=setdiff(1:npop,testi)
```

Ces listes d'indices seront utilisées dans les prochains TPs pour extraire des échantillons d'apprentissage et de test afin de comparer entre elles les performances des différentes méthodes dans leur capacité de prédiction.

2 Bootstrap

Principe

Le *bootstrap* est une technique de rééchantillonnage permettant de simuler la distribution d'un estimateur quelconque pour en apprécier le biais et la variance, donc le risque quadratique, ou encore pour en estimer un intervalle de confiance même si la loi théorique est inconnue. Le principe consiste à estimer itérativement ce même estimateur sur des réalisations différentes de l'échantillon obtenus aléatoirement par n tirages avec remise dans l'échantillon initial.

Un échantillon bootstrap est facilement obtenu avec l'option `replace=TRUE` de la commande `sample`. Comparer :

```
sample(1:20,20)
sample(1:20, 20, replace=TRUE)
```

Régression simple

L'exemple élémentaire ci-dessous génère une enveloppe pouvant s'interpréter comme une région de confiance de la droite de régression du revenu en fonction du nombre d'appartements.

```
#Lire les données si nécessaire :
suit=read.table("suitincom.dat")
names(suit)=c("revenu","nbappt")
# Tracé du nuage de points
plot(suit$nbappt,suit$revenu)
# Itération du tirage de l'échantillon bootstrap
# et des estimations des régressions
for (i in 1:100){
suit.b=suit[sample(47,47,replace=TRUE),]
reg=lm(revenu~nbappt,data=suit.b)
abline(reg)}
```

La même chose est obtenue pour l'autre modèle :

```
lsuit=data.frame(log(suit$nbappt),log(suit$revenu))
names(lsuit)=c("Lrevenu","Lnbappt")
plot(lsuit$Lnbappt,lsuit$Lrevenu)
for (i in 1:100){
suit.b=lsuit[sample(47,47,replace=TRUE),]
reg=lm(Lrevenu~Lnbappt,data=suit.b)
abline(reg)}
```

Ou encore pour des régressions non-paramétriques :

```
plot(lsuit$Lnbappt,lsuit$Lrevenu)
for (i in 1:100) {
suit.b=lsuit[sample(47,47,replace=TRUE),]
lsuit.spl=smooth.spline(suit.b$Lnbappt,
    suit.b$Lrevenu,df=4)
lines(lsuit.spl, col = "blue")}
```

Régression linéaire multiple

Cet outil permet d'obtenir des informations sur les distributions des paramètres.

```
# Lecture des données
ukcomp1=read.table('ukcomp1_r.dat', header=TRUE)
```

Le bootstrap construit "à la main" :

```
# Initialisation de la matrice qui contiendra
# les différentes estimations
stock=data.frame(matrix(0,100,5))
names(stock)=c("CST","CFTDT","LOGSALE",
    "NFATAST","CURRAT")
# Itération des tirages des échantillons
# et des estimations
for (i in 1:100) {
Ib=sample(40,40,replace=TRUE)
stock[i,]=coef(lm(RETCAP~WCFTDT+LOGSALE+
    NFATAST+CURRAT,data=ukcomp1[Ib,])) }
# Dispersion des estimations des paramètres}
boxplot(stock,horizontal=TRUE)
```

Même chose pour le modèle optimal au sens du R^2 ajusté.

```
stock=data.frame(matrix(0,100,9))
names(stock)=c("CST","WCFTDT","LOGSALE","LOGASST",
  "NFATAST","FATTOT","INVTAST","QUIKRAT","CURRAT")
for (i in 1:100) {
Ib=sample(40,40,replace=TRUE)
stock[i,]=coef(lm(RETCAP~WCFTDT+LOGSALE+LOGASST+
   NFATAST+FATTOT+INVTAST+QUIKRAT+CURRAT,
   data=ukcomp1[Ib,]))}
boxplot(stock,horizontal=TRUE)
```

Comparer les dispersions des paramètres.

2.1 Librairie spécifique

Des fonctions sont prévues dans le package `boot` pour simplifier le travail et qui s'appliquent à toute statistique. Ainsi, pour estimer les biais et écarts-types "bootstrap" des paramètres en régression :

```
# Chargement de la librairie
library(boot)
# Définition de la fonction qui calcule
# la statistique d'intérêt à bootstraper
# Celle-ci comporte deux paramètres :
# les données et les indices d'un échantillon
uk1.fun=function(d,i) coef(lm(RETCAP~
   WCFTDT+LOGSALE+NFATAST+CURRAT,data=d[i,]))
# lancement du bootstrap
uk1.b=boot(ukcomp1,uk1.fun,R=999)
# Paramètres, biais et écart-type bootstrap
# des paramètres
uk1.b
boxplot(data.frame(uk1.b$t),horizontal=TRUE)
```

Même chose pour l'autre modèle.

```
uk2.fun=function(d,i) coef(lm(RETCAP~WCFTDT+
  LOGSALE+LOGASST+NFATAST+FATTOT+INVTAST+
  QUIKRAT+CURRAT,data=d[i,]))
uk2.b=boot(ukcomp1,uk2.fun,R=999)
uk2.b
boxplot(data.frame(uk2.b$t),horizontal=TRUE)
```