Module 5 - Resampling - Recommended exercises

TMA4268 Statistical Learning V2019

Martin Kvisvik Larsen

07 februar, 2019

Problem 1: Explain how *k*-fold cross-validation is implemented

Problem 3: Selection bias and the "wrong way to do CV"

```
library(boot)
# GENERATE DATA
# reproducible
set.seed(4268)
n=50 #number of observations
p=5000 #number of predictors
d=25 #top correlated predictors chosen
kfold=10
#generating predictor data
xs=matrix(rnorm(n*p,0,4),ncol=p,nrow=n) #simple way to to uncorrelated predictors
dim(xs) # n times p
# generate class labels independent of predictors - so if all classifies as class 1 we expect 50% error
ys=c(rep(0,n/2),rep(1,n/2)) #now really 50% of each
table(ys)

# WRONG CV - using cv.glm
# here select the most correlated predictors outside the CV
corrs=apply(xs,2,cor,y=ys)
hist(corrs)
selected=order(corrs^2,decreasing = TRUE)[1:d] #top d correlated selected
data=data.frame(ys,xs[,selected])
#apply(xs[,selected],2,cor,y=ys) yes, ave the most correlated
# then cv around the fitting of the classifier - use logistic regression and built in cv.glm function
logfit=glm(ys~.,family="binomial",data=data)
cost <- function(r, pi = 0) mean(abs(r-pi) > 0.5)
cvres=cv.glm(data=data,cost=cost,glmfit=logfit,K=kfold)
cvres$delta
# observe - near 0 misclassification rate

# WRONG without using cv.glm - should be similar (just added to see the similarity to the RIGHT version
reorder=sample(1:n,replace=FALSE)
validclass=NULL
for (i in 1:kfold)
{
  neach=n/kfold
  trainids=setdiff(1:n,(((i-1)*neach+1):(i*neach)))
  traindata=data.frame(xs[reorder[trainids],],ys[reorder[trainids]])
```

```
  validdata=data.frame(xs[reorder[-trainids],],ys[reorder[-trainids]])
  colnames(traindata)=colnames(validdata)=c(paste("X",1:p),"y")
  data=traindata[,c(selected,p+1)]
  trainlogfit=glm(y~.,family="binomial",data=data)
  pred=plogis(predict.glm(trainlogfit,newdata=validdata[,selected]))
  print(pred)
  validclass=c(validclass,ifelse(pred > 0.5, 1, 0))
}
table(ys[reorder],validclass)
1-sum(diag(table(ys[reorder],validclass)))/n

# CORRECT CV
reorder=sample(1:n,replace=FALSE)
validclass=NULL
for (i in 1:kfold)
{
  neach=n/kfold
  trainids=setdiff(1:n,(((i-1)*neach+1):(i*neach)))
  traindata=data.frame(xs[reorder[trainids],],ys[reorder[trainids]])
  validdata=data.frame(xs[reorder[-trainids],],ys[reorder[-trainids]])
  colnames(traindata)=colnames(validdata)=c(paste("X",1:p),"y")
  foldcorrs= apply(traindata[,1:p],2,cor,y=traindata[,p+1])
  selected=order(foldcorrs^2,decreasing = TRUE)[1:d] #top d correlated selected
  data=traindata[,c(selected,p+1)]
  trainlogfit=glm(y~.,family="binomial",data=data)
  pred=plogis(predict.glm(trainlogfit,newdata=validdata[,selected]))
  validclass=c(validclass,ifelse(pred > 0.5, 1, 0))
}
table(ys[reorder],validclass)
1-sum(diag(table(ys[reorder],validclass)))/n
```

Bootstrapping

Problem 1: Probability of being part of a bootstrap sample

- a) Probability of drawing x_i: $P(X = x_i) = \frac{1}{n}$, Probability of not drawing x_i: $P(X \neq x_i) = 1 - \frac{1}{n}$
- b) Probability of not drawing x_i: $P(X_x \neq x) = (1 - \frac{1}{n})^n$, $P(atleast\ one\ x_i) = 1 - (1 - \frac{1}{n})^n$
- c)