Exercise 7

Empirical, functional and generalisation risk

G. Bontempi

Question

Consider an input/output regression task where $n = 1$, $E[\mathbf{y}|x] = \sin(\pi/2x)$, $p(y|x) = \mathcal{N}(\sin(\pi/2x), \sigma^2)$, $\sigma = 0.1$ and $\mathbf{x} \sim \mathcal{U}(-2, 2)$. Let N be the size of the training set and consider a quadratic loss function.

Let the class of hypothesis be $h_M(x) = \alpha_0 + \sum_{m=1}^{M} \alpha_m x^m$ with $\alpha_j \in [-2, 2], j = 0, \dots, M$.

For $N = 20$ generate $S = 50$ replicates of the training set. For each replicate, estimate the value of the parameters that minimise the empirical risk, compute the empirical risk and the functional risk.

The student should

- Plot the evolution of the distribution of the empirical risk for $M = 0, 1, 2$.
- Plot the evolution of the distribution of the functional risk for $M = 0, 1, 2$.
- Discuss the results.

Hints: to minimise the empirical risk, perform a grid search in the space of parameter values, i.e. by sweeping all the possible values of the parameters in the set $[-1, -0.9, -0.8, \dots, 0.8, 0.9, 1]$. To compute the functional risk generate a set of $N_{ts} = 10000$ i.i.d. input/output testing samples.

Regression function

Let us first define a function implementing the conditional expectation function, i.e. the regression function

```
rm(list=ls())
## This resets the memory space


regrF<-function(X){
  return(sin(pi/2*X))
}
```

Parametric identification function

This function implements the parametric identification by performing a grid search in the space of parameters. Note that for a degree equal to m, there are $m + 1$ parameters. If each parameter takes value in a set of values of size V, the number of configurations to be assessed by grid search amounts to V^{m+1}. Grid search is definitely a poor way of carrying out a parametric identification. Here it is used only to illustrate the notions of empirical risk.

```
parident<-function(X,Y,M=0){

  A=seq(-1,1,by=0.1)
  ## set of values that can be taken by the parameter

```

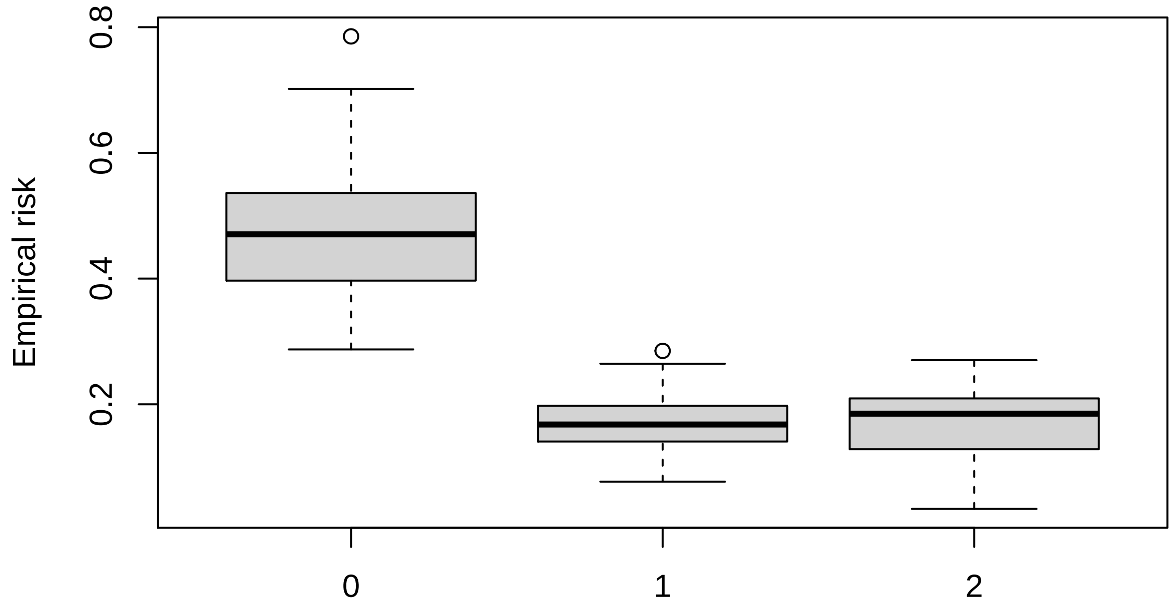
```
  N=NROW(X)
  Xtr=numeric(N)+1
  if (M>0)
    for (m in 1:M)
      Xtr=cbind(Xtr,X^m)

  l <- rep(list(A), M+1)
  cA=expand.grid(l)
  ## set of all possible combinations of values

  bestE=Inf

  ## Grid search
  for (i in 1:NROW(cA)){
    Yhat=Xtr%*%t(cA[i,])
    ehat=mean((Yhat-Y)^2)
    if (ehat<bestE){
      bestA=cA[i,]
      ## best set of parameters
      bestE=ehat
      ## empirical risk associated to the best set of parameters
    }
  }
  return(list(alpha=bestA,Remp=bestE))
}
```

Monte Carlo simulation

Here we generate a number S of training sets of size N. For each of them we perform the parametric identification, we select the set of parameters α_N and we compute the functional risk by means of a test set of size N_{ts}

```
sdw=0.1
S=50
N=20
M=2
aEmp=array(NA,c(S,M+1))
aFunct=array(NA,c(S,M+1))
Nts=10000

# test set generation
Xts<-runif(Nts,-2,2)
Yts=regrF(Xts)+rnorm(Nts,0,sdw)

for (m in 0:M)
  for ( s in 1:S){
    ## training set generation
    Xtr<-runif(N,-2,2)
    Ytr=regrF(Xtr)+rnorm(N,0,sdw)


    ParIdentification=parident(Xtr,Ytr,m)
    aEmp[s,m+1]=ParIdentification$Remp
    XXts=array(numeric(Nts)+1,c(Nts,1))
```

```
    if (m>0)
      for (j in 1:m)
        XXts=cbind(XXts,Xts^j)
    aFunct[s,m+1]=mean((Yts-XXts%*%t(ParIdentification$alpha))^2)
  }

colnames(aEmp)=0:M
colnames(aFunct)=0:M
boxplot(aEmp, ylab="Empirical risk")
```




```
boxplot(aFunct, ylab="Functional risk")
```