Exercise 4

Strongly and weakly relevant features

G. Bontempi

Question

The .Rdata file **https://www.dropbox.com/s/kfevu16cf5mxptc/bonus4.Rdata?dl=0** contains a regression dataset with $N = 200$ samples, $n = 50$ input features (in the matrix `X`) and one target variable (vector `Y`).

Knowing that there are 3 strongly relevant variables and 2 weakly relevant variables, the student has to define and implement a strategy to find them.

No existing feature selection code has to be used. However, the student may use libraries to implement supervised learning algorithms.

The student code should

- return the position of the 3 strongly relevant variables and 2 weakly relevant variables,
- discuss what strategy could have been used if the number of strongly and weakly variables was not known in advance.

Data generation

Let us see first how the input-output dataset was generated. The knowledge of the stochastic process generating the data will allow us to define the correct set of strongly and weakly relevant features.

```r
rm(list=ls())
set.seed(0)
N<-200
n<-50
strong<-c(1,7,n)
weak<-c(8,9)
irr<-setdiff(1:n,c(strong,weak))
ns<-length(strong)
nw<-length(weak)

Xw<-array(rnorm(N*nw),c(N,nw))

X=array(rnorm(N*n),c(N,n))


X[,strong[1]]=apply(abs(Xw),1,sum)+rnorm(N,sd=0.1)
X[,strong[2]]=apply(abs(Xw),1,prod)+rnorm(N,sd=0.1)
X[,strong[3]]=log(apply(abs(Xw),1,prod))+rnorm(N,sd=0.1)

X[,weak]=Xw

X=scale(X)
Y=apply(abs(X[,strong]),1,sum)+rnorm(N,sd=0.1)
save(file="bonus4.Rdata",list=c("X","Y"))
```

The relationship between $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{50}\}$ and $\mathbf{y}$ is given by

$$\mathbf{y} = |x_1 + x_7 + x_{50}| + \mathbf{w} \tag{1}$$

where $\mathbf{x}_1 = |x_8 + x_9| + \mathbf{w}_1$, $\mathbf{x}_7 = |x_8 x_9| + \mathbf{w}_7$, $\mathbf{x}_{50} = \log |x_8 x_9| + \mathbf{w}_{50}$ and $\mathbf{w}, \mathbf{w}_1, \mathbf{w}_7, \mathbf{w}_{50}$, are all Normal with zero mean and standard deviation 0.1.

Definition of strongly and weakly relevant features

In the course a strongly relevant feature is defined as a feature $\mathbf{x}_j$ such that

$$I(\mathbf{X}_{-j}, \mathbf{y}) < I(\mathbf{X}, \mathbf{y})$$

or equivalently

$$H(\mathbf{y}|\mathbf{X}_{-j}) > H(\mathbf{y}|\mathbf{X})$$

By removing a strongly relevant feature from the input set, the conditional variance of $\mathbf{y}$ increases.

From (1) it follows that

$$p(y|X) = p(y|x_1, x_7, x_{50})$$

or equivalently that $\mathbf{y}$ is conditionally independent of all the other variables when the value of $\{x_1, x_7, x_{50}\}$ is known.

The set of strongly relevant variables (which is also the Markov blanket) is then $\{\mathbf{x}_1, \mathbf{x}_7, \mathbf{x}_{50}\}$.

A weakly relevant feature is a feature $\mathbf{x}_j$ that is not strongly relevant and such that

$$I(\mathbf{S}_{-j}, \mathbf{y}) < I(\mathbf{S}, \mathbf{y})$$

or equivalently

$$H(\mathbf{y}|\mathbf{S}_{-j}) > H(\mathbf{y}|\mathbf{S})$$

for a certain context $S \subset X$. If we consider $S = X \setminus \{x_1, x_7, x_{50}\}$ then

$$p(y|S) = p(y|x_8, x_9)$$

It follows that $\mathbf{y}$ is conditionally independent of all the other features of S when the value of $\{x_8, x_9\}$ is known.

The set of weakly relevant variables is then $\{\mathbf{x}_8, \mathbf{x}_9\}$.

In other terms the set of weakly relevant variables $\{x_8, x_9\}$ provides information about $\mathbf{y}$ for some contexts, e.g. the contexts where $\{x_1, x_7, x_{50}\}$ are not available.

All the other features are irrelevant since they play no role in the dependency between $\mathbf{X}$ and $\mathbf{y}$.

Data-driven estimation of conditional entropy

In the real setting (i.e. the one the student is confronted with) the conditional probability (1) and the relationships between input features is not accessible. It is not then possible to compute analytically the information or the entropy terms.

Nevertheless, it is possible to estimate the conditional probability $p(y|S)$ and consequently the conditional entropy term $H(\mathbf{y}|\mathbf{S})$ for a subset S of features by making some assumptions:

1. we have a learning model able to return an unbiased and low variant estimation of the regression function. In this case the estimated MISE returns a good approximation of the conditional variance (i.e. the noise variance)
2. the conditional probability is Gaussian. In this case there is a direct link between the conditional variance and the conditional entropy.

In other terms we make the assumption that

$$H(\mathbf{y}|\mathbf{S}_1) < H(\mathbf{y}|\mathbf{S}_2)$$

if

$$\widehat{\text{MISE}}_1 < \widehat{\text{MISE}}_2$$

where $\widehat{\text{MISE}}_i$ is the estimated (e.g. by leave-one-out) generalization error of a learner trained with the input set S_i.

Data-driven identification of strongly relevant features

Here we identify in a data-driven manner the set of strongly relevant features by choosing as learner a Random Forest and by using a holdout strategy to estimate the generalization error.

In practice, we

1. remove a single input feature at the time,
2. split the dataset in training and validation set and learn a Random Forest with the training set
3. compute the Random Forest generalization error for the validation set
4. rank the features to select the ones that induced a largest increase of the generalization error

```
library(gbcode)
load("bonus4.Rdata")
Itr<-sample(1:N,round(N/2))
Ival<-setdiff(1:N,Itr)
Yhat<-pred("rf",X[Itr,],Y[Itr],X[Ival,],class=FALSE)
Ehat=(Y[Ival]-Yhat)^2
MISEhat=mean(Ehat)  ## Holdout MISE computation
print( mean(MISEhat^2))
```

```
## [1] 0.1551874
```

```
MISEhatj=numeric(n)
Ehatj=array(NA,c(length(Ival),n))
for (j in 1:n){
  Yhatj<-pred("rf",X[Itr,-j],Y[Itr],X[Ival,-j],class=FALSE)
  ## we use the wrapper available in the gbcode package
  Ehatj[,j]=(Y[Ival]-Yhatj)^2
  ## estimation of the generalization error with the validation set
  MISEhatj[j]=mean(Ehatj[,j])
}

stronghat=sort(MISEhatj-MISEhat,decr=TRUE,index=TRUE)$ix[1:ns]
## Ranking of the features according to the increase of the validation error

cat("Strongly relevant identified=",stronghat,"\n")
```

```
## Strongly relevant identified= 50 1 7
```

According to the procedure above, by knowing that there are **3** strongly relevant variables, the set of strongly relevant variables is in the columns **50, 1, 7** of the input matrix X.

Data-driven identification of weakly relevant features

The identification of weakly relevant variables would need a search in the space of all possible contexts. Here we limit to consider the context $S = X \setminus \{x_1, x_7, x_{50}\}$ obtained by removing the strongly relevant features from the input set. The hold-out procedure is similar to the one in the previous section.

```
Yhat<-pred("rf",X[Itr,-strong],Y[Itr],X[Ival,-strong],class=FALSE)

wMISEhat=mean((Y[Ival]-Yhat)^2)
print( mean(wMISEhat^2))
```

```
## [1] 7.583091
```

```
wMISEhatj=numeric(n)-100
for (j in setdiff(1:n,strong)){
  Yhatj<-pred("rf",X[Itr,-c(strong,j)],Y[Itr],X[Ival,-c(strong,j)],class=FALSE)
  wMISEhatj[j]=mean((Y[Ival]-Yhatj)^2)
}
weakhat=sort(wMISEhatj-wMISEhat,decr=TRUE,index=TRUE)$ix[1:nw]
print(sort(wMISEhatj-wMISEhat,decr=TRUE,index=TRUE)$x[1:nw])
```

```
## [1] 0.11752119 0.04354846
```

```
cat("Weakly relevant identified=",weakhat,"\n")
```

```
## Weakly relevant identified= 8 9
```

According to the procedure above we see that there are **2** features that, once removed, increase the generalization error of the context $S = X \setminus \{x_1, x_7, x_{50}\}$. We may deduce then that the set of weakly relevant variables is in the columns **8, 9** of the input matrix X.

What to do in the general case

The solution in this exercise has been facilitated by the knowledge of the number of strongly and weakly relevant features. Unfortunately, this information is hardly available in real settings.

The main issue related to identification of relevant features is that we cannot compute the analytical exact value of the conditional entropy (or conditional information terms) because of the stochastic finite-data setting. In practice we have only rough estimates of those terms. Nevertheless, most of the time we are not interested in the actual values of those terms but in their relative values: for instance we may be interested to know if

$$H(\mathbf{y}|\mathbf{S}_1) < H(\mathbf{y}|\mathbf{S}_2)$$

of if their difference is smaller than zero.

Since those values are only estimated the fact that

$$\hat{H}(\mathbf{y}|\mathbf{S}_1) < \hat{H}(\mathbf{y}|\mathbf{S}_2)$$

does not necessarily provide enough evidence to draw a conclusion. Given the stochastic setting a solution could be the adoption of statistical tests. For instance if H is approximated by $\widehat{\text{MISE}}$ we could use a statistical test to check whether the mean $\widehat{\text{MISE}}_1$ is significantly smaller than $\widehat{\text{MISE}}_2$.

Let us see how this could be done in practice.

Data-driven identification of the number of strongly relevant features

In this case we do not know exactly where to stop in the decreasing ranking of the vector `MISEhatj` $-$ `MISEhat`.

In what follows we use a t-test comparing the vector of test errors (stored in the R variable `Ehatj`) of each feature set X_{-j} to the the one of X (stored in the R variable `Ehat`). This checks if the mean $\widehat{\text{MISE}}_{-j}$ is significantly larger (pvalue smaller than 0.01) than $\widehat{\text{MISE}}$.

```
pv=numeric(n)
for (j in 1:n)
  pv[j]=t.test(Ehatj[,j],Ehat,alternative="greater",paired=TRUE)$p.value

stronghat.test=which(pv<0.01)
print(sort(pv,index=TRUE))
```

```
## $x
##  [1] 6.899891e-09 4.907560e-07 1.244413e-06 1.068473e-02 1.047014e-01
##  [6] 1.064333e-01 1.724474e-01 1.788466e-01 1.806556e-01 2.227288e-01
## [11] 2.497905e-01 3.232423e-01 3.650209e-01 3.726473e-01 4.720321e-01
## [16] 5.799603e-01 6.222948e-01 6.459867e-01 6.638367e-01 7.172664e-01
## [21] 7.694386e-01 7.863708e-01 8.187977e-01 8.396133e-01 8.875101e-01
## [26] 9.260295e-01 9.356320e-01 9.510724e-01 9.523181e-01 9.692637e-01
## [31] 9.696870e-01 9.755537e-01 9.769522e-01 9.795273e-01 9.889207e-01
## [36] 9.903859e-01 9.907410e-01 9.961684e-01 9.972564e-01 9.974957e-01
## [41] 9.978702e-01 9.984824e-01 9.988223e-01 9.991719e-01 9.993707e-01
## [46] 9.995593e-01 9.998722e-01 9.999618e-01 9.999731e-01 9.999862e-01
##
## $ix
##  [1]  1 50  7  6  2 25  9 35 41 18 22 36 19  5 47 29  3 33 21 27 31 30 43 11  8
## [26] 16 20 23 32 38 15 28 49 14 44 34 45 46 42 17  4 48 24 37 40 13 12 39 10 26
```

It follows that (for the given pvalue threshold) the set of strongly relevant features is **1, 7, 50**. Of course this number could be different for different pvalue thresholds.

Data-driven identification of the number of weakly relevant features

The procedure above can be used as well for detecting weakly relevant features for a given context. Nevertheless, since the number of weakly features is not given in advance, the problem of finding the set of weakly relevant features would remain much harder. In fact, we are not supposed to stop the search until we have not considered all the possible contexts.