Practical 6 Ex1

Alexander Maslev

2018 M04 9

a)

First, we load the SIS package and the data set.

```
rm(list=ls())
require(MASS)
```

```
## Loading required package: MASS
```

```
set.seed(11)
library(SIS)
leuk_dat=read.csv("leukemia_big.csv",header=T)
```

We create the X matrix by transposing the dataset to have rows representing the subjects, then we create the vector y of responses by assigning 1 to all ALL subjects and 0 to AML subjects.

```
X_leuk=t(as.matrix(leuk_dat))
y_leuk = c(rep(1,20),rep(0,14), rep(1,27), rep(0,11))
```

b)

We split the data in half into a train set and a test set.

```
num=1:length(leuk_dat)
index = sample(x = num,size = length(num)/2,replace = F)
train_X = X_leuk[index,]
test_X = X_leuk[-index,]
train_y=y_leuk[index]
test_y=y_leuk[-index]
```

c)

We create 3 models using 3 possible combinations of penalized methods and tuning constants; the selected covariates and their coefficients are shown below.

```
mod1=SIS(train_X,train_y,family="binomial",penalty = "SCAD",tune="bic")
```

```
## Iter 1 , screening:  6854
## Iter 1 , selection:  6854
## Iter 1 , conditional-screening:  4489
## Iter 2 , screening:  4489 6854
## Iter 2 , selection:  6854
## Model already selected
```

```
mod2=SIS(train_X,train_y,family="binomial",penalty = "MCP",tune="aic")
```

```
## Iter 1 , screening:  6854
## Iter 1 , selection:  6854
## Iter 1 , conditional-screening:  4489
## Iter 2 , screening:  4489 6854
## Iter 2 , selection:  6854
## Model already selected
```

```
mod3=SIS(train_X,train_y,family="binomial",penalty = "lasso",tune="cv")
```

```
## Iter 1 , screening:  6854
## Iter 1 , selection:  6854
## Iter 1 , conditional-screening:  4489
## Iter 2 , screening:  4489 6854
## Iter 2 , selection:  4489 6854
## Maximum number of variables selected
```

```
mod1$coef.est
```

```
## (Intercept)       X6854
##   -3.039013    3.695333
```

```
mod2$coef.est
```

```
## (Intercept)       X6854
##   -3.629016    4.319310
```

```
mod3$coef.est
```

```
## (Intercept)       X4489       X6854
##  -3.9244378  -0.1566402   4.8488812
```

Models 1 and 2 select identical covariates, while model 3 has an additional variable. We use the models to get the predictions and test the error rate of each model.

```
pred1=predict(mod1,test_X,type="class")
sum(pred1!=test_y)/36 #error rate
```

```
## [1] 0.1111111
```

```
pred2=predict(mod2,test_X,type="class")
sum(pred2!=test_y)/36 #error rate
```

```
## [1] 0.1111111
```

```
pred3=predict(mod3,test_X,type="class")
sum(pred3!=test_y)/36 #error rate
```

```
## [1] 0.1111111
```

All methods perform equally well in terms of prediction error.