Ridge Regression

Gregoire Gasparini, Aurora Hofman, Sarah Musiol, Beatriu Tort

25/02/20

Choosing the penalization parameter

Testdate (to be removed later!)

```
prostate <- read.table("/Users/aurorahofman/Documents/Utveksling/Stat_lær/week_2/prostate_data.txt", hea
plot(prostate)
train.sample <- which(prostate$train==TRUE) ##separate trainingsdata from testdata
val.sample <- which(prostate$train==FALSE)


Y_t <- scale( prostate$lpsa[train.sample], center=TRUE, scale=FALSE) ## center but not scale for respon
X_t <- scale( as.matrix(prostate[train.sample,1:8]), center=TRUE, scale=TRUE) ##scale and center for

Y_val <- scale( prostate$lpsa[val.sample], center=TRUE, scale=FALSE) ## center but not scale for respon
X_val <- scale( as.matrix(prostate[val.sample,1:8]), center=TRUE, scale=TRUE)

#predictors
p <- dim(X)[2]

XtX <- t(X)%*%X
d2 <- eigen(XtX,symmetric = TRUE, only.values = TRUE)$values #eigenvalues of xtx

(cond.number <- sqrt(max(d2)/min(d2)))

lambda.max = 1e4
n_lambdas <- 25 ## look at 25 different values
lambda.v <- exp(seq(0,log(lambda.max+1),length=n_lambdas))-1 #lambda vector

n_val <- length(Y_val)
```

```
#make m_hat

beta.path <- matrix(0,nrow=n.lambdas, ncol=p) ##making an empthy matrix

PMSE_vec <- vector("numeric", length = n.lambdas)
for(l in 1:n_lambdas){
  lambda <- lambda.v[l]
  beta_hat <- solve(XtX + lambda*diag(1,p)) %*% t(X) %*% Y_t
  #y_hat = X %*% beta_hat
  m_hat_vec <- vector("numeric", length = n_val)

  for (n in 1:n_val){
    m_hat_vec[n] <- (Y_val[n]-(X_val[n,]%*%beta_hat))^2
```

```
  }
  PMSE_vec[l]<- sum(m_hat_vec)/n_val

}



lambda.CV <- lambda.v[which.min(PMSE_vec)]
plot(log(1+lambda.v), PMSE_vec)
abline(v=log(1+lambda.CV),col=2,lty=2)
```

```
PMSE_vs <- function(x_t, y_t, x_val, y_val, lambda){

}
```

Ridge Regression for the Boston Housing data