Model selection

In this notebook we're going to analyse different techniques for model selection and afterwards we're going to discuss their shortcomings.

Selection criteria

First of all, we're going to look at different criteria to compare models based on their performance and complexity.

```r
require(ISLR)
```

```
## Loading required package: ISLR

## Warning: package 'ISLR' was built under R version 3.6.3
```

```r
head(Hitters)
```

```
##                   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun
## -Andy Allanson      293   66     1   30  29    14     1    293    66      1
## -Alan Ashby         315   81     7   24  38    39    14   3449   835     69
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     63
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575    225
## -Andres Galarraga   321   87    10   39  42    30     2    396   101     12
## -Alfredo Griffin    594  169     4   74  51    35    11   4408  1133     19
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Andy Allanson       30   29     14      A        E     446      33     20
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
## -Andre Dawson       828  838    354      N        E     200      11      3
## -Andres Galarraga    48   46     33      N        E     805      40      4
## -Alfredo Griffin    501  336    194      A        W     282     421     25
##                   Salary NewLeague
## -Andy Allanson        NA         A
## -Alan Ashby        475.0         N
## -Alvin Davis       480.0         A
## -Andre Dawson      500.0         N
## -Andres Galarraga   91.5         N
## -Alfredo Griffin   750.0         A
```

```r
summary(Hitters)
```

```
##      AtBat            Hits          HmRun            Runs
##  Min.   : 16.0   Min.   :  1   Min.   : 0.00   Min.   :  0.00
##  1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25
##  Median :379.5   Median : 96   Median : 8.00   Median : 48.00
##  Mean   :380.9   Mean   :101   Mean   :10.77   Mean   : 50.91
##  3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00
##  Max.   :687.0   Max.   :238   Max.   :40.00   Max.   :130.00
##
##       RBI             Walks            Years            CAtBat
```

```
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0
##  1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.:  816.8
##  Median : 44.00   Median : 35.00   Median : 6.000   Median : 1928.0
##  Mean   : 48.03   Mean   : 38.74   Mean   : 7.444   Mean   : 2648.7
##  3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.: 3924.2
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0
##
##      CHits            CHmRun           CRuns             CRBI
##  Min.   :   4.0   Min.   :  0.00   Min.   :   1.0   Min.   :   0.00
##  1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.:  88.75
##  Median : 508.0   Median : 37.50   Median : 247.0   Median : 220.50
##  Mean   : 717.6   Mean   : 69.49   Mean   : 358.8   Mean   : 330.12
##  3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.: 426.25
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.00
##
##      CWalks          League  Division    PutOuts          Assists
##  Min.   :   0.00   A:175   E:157    Min.   :   0.0   Min.   :  0.0
##  1st Qu.:  67.25   N:147   W:165    1st Qu.: 109.2   1st Qu.:  7.0
##  Median : 170.50                    Median : 212.0   Median : 39.5
##  Mean   : 260.24                    Mean   : 288.9   Mean   :106.9
##  3rd Qu.: 339.25                    3rd Qu.: 325.0   3rd Qu.:166.0
##  Max.   :1566.00                    Max.   :1378.0   Max.   :492.0
##
##      Errors          Salary        NewLeague
##  Min.   : 0.00   Min.   :  67.5   A:176
##  1st Qu.: 3.00   1st Qu.: 190.0   N:146
##  Median : 6.00   Median : 425.0
##  Mean   : 8.04   Mean   : 535.9
##  3rd Qu.:11.00   3rd Qu.: 750.0
##  Max.   :32.00   Max.   :2460.0
##                  NA's   :59
```

```
# removing the NA
dim(Hitters)
```

```
## [1] 322  20
```

```
Hitters<- na.omit(Hitters)
dim(Hitters)
```

```
## [1] 263  20
```

We're going to use cross-validation to compare the results from different selection criteria.

```
nfolds <- 10
n <- dim(Hitters)[1]
folds <- cut(1:n, nfolds, labels = F)
# a bit of shuffling
indices <- sample(1:n, size=n, replace=F)
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.6.3
```

```
get.bss.test.error<- function(train, test, cv.best){
  # estimates the error on the test dataset for the best model
  # according to each criteria
  all.best<- regsubsets(x=Salary~.,data=train,nbest=1,
```

```r
                        nvmax=dim(train)[2]-1, # using all variables
                        method="forward" )
  s <- summary(all.best)
  r2 <- coef(all.best, id=which.max(s$rsq))
  adjr2 <- coef(all.best, id=which.max(s$adjr2))
  cp <- coef(all.best, id=which.min(s$cp))
  bic <- coef(all.best, id=which.min(s$bic))
  cv.coefs <- coef(all.best, id=cv.best)
  # test predictions
  r2.pred <- model.matrix(Salary~.,test)[,names(r2)]%*%r2
  adjr2.pred <- model.matrix(Salary~.,test)[,names(adjr2)]%*%adjr2
  cp.pred <- model.matrix(Salary~.,test)[,names(cp)]%*%cp
  bic.pred <- model.matrix(Salary~.,test)[,names(bic)]%*%bic
  cv.pred <- model.matrix(Salary~.,test)[,names(cv.coefs)]%*%cv.coefs
  # test errors
  errors <- mean((r2.pred - test$Salary)**2)
  errors <- c(errors,mean((adjr2.pred - test$Salary)**2))
  errors <- c(errors,mean((cp.pred - test$Salary)**2))
  errors <- c(errors,mean((bic.pred - test$Salary)**2))
  errors <- c(errors,mean((cv.pred - test$Salary)**2))
  return(errors)
}


get.cv.error <- function(ncv, nmodels, data){
  # evaluates the mean cross-validation error of the linear model
  # with the selected coefficients
  n.cv <- dim(data)[1]
  folds.cv <- cut(1:n.cv, ncv, labels=F)
  cv.errors <- matrix(nrow = ncv, ncol = nmodels)
  indices.cv <- 1:n.cv
  for(m in 1:nmodels){
    for(j in 1:ncv){
      test.indices.cv <- indices.cv[folds.cv==j]
      test.cv <- data[test.indices.cv,]
      train.cv <- data[-test.indices.cv,]
      cv.all.best<- regsubsets(x=Salary~.,data=train.cv,
                               nbest=1,nvmax=nmodels, # using all variables
                               method="forward" )
      cv.coefs <- coef(cv.all.best, id=m)
      cv.preds <- model.matrix(Salary~.,test)[,names(cv.coefs)]%*%cv.coefs
      # test errors
      cv.errors[j,m] <- mean((cv.preds - test$Salary)**2)
    }
  }
  # selecting the model with the least mean error
  # expected test MSE estimated by CV for each model
  return(which.min(colMeans(cv.errors)))
}

test.errors <- matrix(nrow=nfolds, ncol=5)

for(i in 1:nfolds){
```

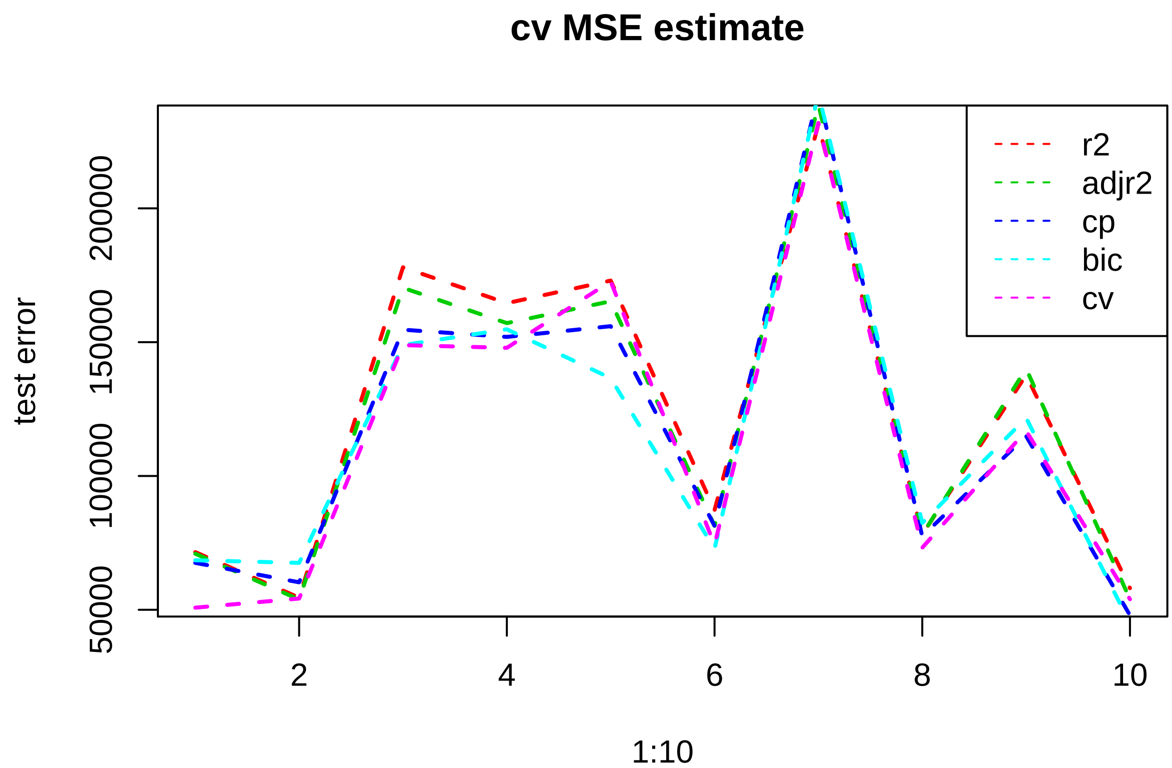
```
  test.indices <- indices[folds==i]
  test <- Hitters[test.indices,]
  train <- Hitters[-test.indices,]
  # Now we'll use BSS on the train dataset
  # And we'll record the error on the test set
  # get best cv model
  cv.best <- get.cv.error(ncv=5, nmodels=(dim(Hitters)[2]-1),data = train)
  test.errors[i,] <- get.bss.test.error(train=train, test=test, cv.best=cv.best)

}
```

Let's look at the results.

```
test.errors <- data.frame(test.errors)
names(test.errors) <- c("r2","adjr2","cp","bic","cv")
test.errors
```

```
##            r2     adjr2        cp       bic        cv
## 1   71560.04  70956.14  67531.03  68505.97  50789.23
## 2   54542.35  53841.13  60274.42  67532.46  54196.36
## 3  177976.55 170310.27 154657.06 148891.50 148891.50
## 4  164565.53 157115.47 151962.32 154818.18 147891.02
## 5  173013.67 165288.54 155974.41 136563.18 172702.85
## 6   87228.87  81371.23  81371.23  72738.66  74254.61
## 7  231359.27 238745.34 242912.60 242912.60 231915.57
## 8   78148.94  78008.45  77441.34  82436.21  73195.58
## 9  137720.38 140130.85 114918.56 121679.32 116867.69
## 10  58077.59  53937.54  47937.25  45000.49  53937.54
```

```
plot(1:10, test.errors$r2, type="l", lty="dashed", col=2, ylab="test error", main="cv MSE estimate ", l
lines(1:10, test.errors$adjr2, type="l", lty="dashed", col=3, lwd=2)
lines(1:10, test.errors$cp, type="l", lty="dashed", col=4, lwd=2)
lines(1:10, test.errors$bic, type="l", lty="dashed", col=5, lwd=2)
lines(1:10, test.errors$cv, type="l", lty="dashed", col=6, lwd=2)
legend("topright", legend = c("r2", "adjr2","cp","bic","cv"), col=c(2,3,4,5,6), lty="dashed")
```




```r
colMeans(test.errors)
```

```
##       r2    adjr2       cp      bic       cv
## 123419.3 120970.5 115498.0 114107.9 112464.2
```

```r
which.min(colMeans(test.errors))
```

```
## cv
##  5
```

So the cross validation criteria seems to be the most reliable in model selection.