# 3301HW6

Joshua O’Neill

2024-04-17

## 1. (8pts) You will analyze a dataset about divorce in the United States from 1920 to 1996.The data are in the file “divorce.txt” posted on canvas. The predictors are

## • year, year;

## • unemployed, unemployment rate;

## • femlab, percent female participation in labor force aged 16+;

## • marriage, marriages per 1000 unmarried women aged 16+;

## • birth, births per 1000 women aged 15-44;

## • military, military personnel per 1000 people.

## The response is divorce, which is the number of divorces per 1000 women that are at least 15 years old.

**(a) (3 pts) Randomly shuffle the rows of the dataframe and then use AIC, BIC, and 5-fold CV (minimizing validation MSPE) to find the best subset of the predictors.**

```
divorce_data <- read.table("divorce.txt", header = TRUE)
head(divorce_data)
```

```
##   year divorce unemployed femlab marriage birth military
## 1 1920     8.0        5.2  22.70     92.0 117.9   3.2247
## 2 1921     7.2       11.7  22.79     83.0 119.8   3.5614
## 3 1922     6.6        6.7  22.88     79.7 111.2   2.4553
## 4 1923     7.1        2.4  22.97     85.2 110.5   2.2065
## 5 1924     7.2        5.0  23.06     80.3 110.9   2.2889
## 6 1925     7.2        3.2  23.15     79.2 106.6   2.1735
```

```
olscv <- function(X, y, K=5, permute=FALSE) {
  n <- length(y)
  if (permute) {
    ind <- sample(n)
  } else {
    ind <- 1:n
  }
  total.sqr.err <- 0
  for (k in 1:K) {
    leave.out <- ind[ (1+floor((k-1)*n/K)):floor(k*n/K) ]
    X.tr <- X[-leave.out,,drop=FALSE]
    y.tr <- y[-leave.out]
    X.va <- X[leave.out,,drop=FALSE]
    y.va <- y[leave.out]
    bhat.tr <- lm.fit(x=X.tr, y=y.tr)$coefficients
    total.sqr.err <- total.sqr.err + sum((y.va - X.va %*% bhat.tr)^2)
  }
  return(total.sqr.err/n)
}
```

```
get.ic <- function(X, y, K, model) {
  n <- dim(X)[1]
  p <- dim(X)[2]
  # print(X)
  # ERROR: First Beta calls X "singular" when it is not.
  # ERROR: Second Beta yield 10^-26 for rss.
  # beta <- qr.solve(crossprod(X),crossprod(X,y))
  beta <- lm.fit(x=X, y=y)$coefficients
  rss <- sum((y - X %*% beta)^2)
  common <- n * log(2 * pi) + n * log(rss/n) + n
  aic <- common + 2* (p+1)
  bic <- common + log(n) * (p + 1)
  mspe.cap <- olscv(X=X,y=y, K=K, permute=TRUE)
  # print(cbind(model, aic, model, bic, model, mspe.cap))
  return(cbind(model, aic, model, bic, model, mspe.cap))
}
```

```r
compare.ic <- function(model1, model2) {
  #model1 is a vector of a the best aic model, its aic, the best bic model
  #, its bic, the best mspe.cap model, its mspe.cap.

  aic1 <- model1[1,2]
  bic1 <- model1[1,4]
  mspe.cap1 <- model1[1,6]
  aic2 <- model2[1,2]
  bic2 <- model2[1,4]
  mspe.cap2 <- model2[1,6]

  final_aic <- aic1
  aic_beta <- model1[,1]
  if (aic1 > aic2) {
    final_aic <- aic2
    aic_beta <- model2[,1]
  }
  final_bic <- bic1
  bic_beta <- model1[,3]
  if (bic1 > bic2) {
    final_bic <- bic2
    bic_beta <- model2[,3]
  }
  final_mspe.cap <- mspe.cap1
  mspe.cap_beta <- model1[,5]
  if (mspe.cap1 > mspe.cap2) {
    final_mspe.cap <- mspe.cap2
    mspe.cap_beta <- model2[,5]
  }
  # print(cbind(aic_beta, final_aic, bic_beta, final_bic, mspe.cap_beta, final_mspe.cap))
  return(cbind(aic_beta, final_aic, bic_beta, final_bic, mspe.cap_beta, final_mspe.cap))
}
```

```
get.best.ic <- function(X, y, K, consideration_index, model) {
  p <- dim(X)[2]
  if (is.null(p)) {
    X = cbind(X)
    return(get.ic(X, y, K, model))
  }
  if (p < consideration_index) {
    return(get.ic(X, y, K, model))
  }
  model1 <- get.best.ic(X, y, K, consideration_index+1, model)

  X <- X[,-consideration_index]
  model[consideration_index + (length(model)-p)] = 0
  model2 <- get.best.ic(X, y, K, consideration_index, model)
  final_model <- compare.ic(model1, model2)
  return(final_model)
  # num_param <- 0
  # x <- model_size
  # while (model_size >= 2) {
  #  model = model / 2
  #  num_param = num_param + 1
  #}

}
```

```
set.seed(3301)
data <- divorce_data[sample(nrow(divorce_data)), ]
X <- cbind(1,data$year, data$unemployed, data$femlab, data$marriage, data$birth, data
$military)
y <- data$divorce
print(get.best.ic(X, y, 5, 2, c(1,1,1,1,1,1,1)))
```

```
##      aic_beta final_aic bic_beta final_bic mspe.cap_beta final_mspe.cap
## [1,]        1  289.8468        1  306.2534             1       2.391983
## [2,]        1  289.8468        1  306.2534             1       2.391983
## [3,]        0  289.8468        0  306.2534             1       2.391983
## [4,]        1  289.8468        1  306.2534             1       2.391983
## [5,]        1  289.8468        1  306.2534             1       2.391983
## [6,]        1  289.8468        1  306.2534             1       2.391983
## [7,]        1  289.8468        1  306.2534             1       2.391983
```

According to my recursive search, the best model is (divorce = β_1 + β_2*year + β_4*femlab + β_5*marriage + β_6*birth + β_7*military), which is agreed upon using AIC and BIC. MSPE, however, chooses the full model: (divorce = β_1 + β_2*unemployment + β_3*year + β_4*femlab + β_5*marriage + β_6*birth + β_7*military)

```
# Testing measures:
full_model <- lm(divorce ~ year + unemployed + femlab + marriage + birth + military,
data)
summary(full_model)
```

```
##
## Call:
## lm(formula = divorce ~ year + unemployed + femlab + marriage +
##     birth + military, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9087 -0.9212 -0.0935  0.7447  3.4689
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 380.14761   99.20371   3.832 0.000274 ***
## year         -0.20312    0.05333  -3.809 0.000297 ***
## unemployed   -0.04933    0.05378  -0.917 0.362171
## femlab        0.80793    0.11487   7.033 1.09e-09 ***
## marriage      0.14977    0.02382   6.287 2.42e-08 ***
## birth        -0.11695    0.01470  -7.957 2.19e-11 ***
## military     -0.04276    0.01372  -3.117 0.002652 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.513 on 70 degrees of freedom
## Multiple R-squared:  0.9344, Adjusted R-squared:  0.9288
## F-statistic: 166.2 on 6 and 70 DF,  p-value: < 2.2e-16
```

```
step(full_model)
```

```
## Start:  AIC=70.41
## divorce ~ year + unemployed + femlab + marriage + birth + military
## 
##              Df Sum of Sq    RSS     AIC
## - unemployed  1     1.925 162.12  69.330
## <none>                    160.20  70.410
## - military    1    22.231 182.43  78.417
## - year        1    33.199 193.40  82.912
## - marriage    1    90.468 250.66 102.884
## - femlab      1   113.214 273.41 109.572
## - birth       1   144.897 305.10 118.015
## 
## Step:  AIC=69.33
## divorce ~ year + femlab + marriage + birth + military
## 
##            Df Sum of Sq    RSS     AIC
## <none>                  162.12  69.330
## - military  1    20.957 183.08  76.691
## - year      1    42.054 204.18  85.089
## - marriage  1   126.643 288.77 111.779
## - femlab    1   158.003 320.13 119.718
## - birth     1   172.826 334.95 123.203
```

```
## 
## Call:
## lm(formula = divorce ~ year + femlab + marriage + birth + military, 
##     data = data)
## 
## Coefficients:
## (Intercept)         year       femlab     marriage        birth     military  
##    405.6167      -0.2179       0.8548       0.1593      -0.1101      -0.0412
```

```
set.seed(3301)
X <- cbind(1,data$year, data$unemployed, data$femlab, data$marriage, data$birth, data$military)
y <- data$divorce
best_model <- get.ic(X,y,5,c(1,1,1,1,1,1,1))
for (a in 0:1) {
  for (b in 0:1) {
    for (c in 0:1) {
      for (d in 0:1) {
        for (e in 0:1) {
          for (f in 0:1) {
            model = c(1,a,b,c,d,e,f)
            new_X = X
            if (f == 0) {
              new_X = new_X[,-7]
            }
            if (e == 0) {
              new_X = new_X[,-6]
            }
            if (d == 0) {
              new_X = new_X[,-5]
            }
            if (c == 0) {
              new_X = new_X[,-4]
            }
            if (b == 0) {
              new_X = new_X[,-3]
            }
            if (a == 0) {
              new_X = new_X[,-2]
            }
            if (is.null(dim(new_X)[2])){
              new_X = cbind(new_X)
            }
            performance <- get.ic(new_X,y,5,model)
            best_model <- compare.ic(performance, best_model)
          }
        }
      }
    }
  }
}
print(best_model)
```

```
##      aic_beta final_aic bic_beta final_bic mspe.cap_beta final_mspe.cap
## [1,]        1  289.8468        1  306.2534             1       2.440191
## [2,]        1  289.8468        1  306.2534             1       2.440191
## [3,]        0  289.8468        0  306.2534             1       2.440191
## [4,]        1  289.8468        1  306.2534             1       2.440191
## [5,]        1  289.8468        1  306.2534             1       2.440191
## [6,]        1  289.8468        1  306.2534             1       2.440191
## [7,]        1  289.8468        1  306.2534             1       2.440191
```

According to my iterative search, the best model is (divorce = β_1 + β_2*year + β_4*femlab + β_5*marriage + β_6*birth + β_7*military), which is agreed upon using AIC and BIC. MSPE, however, chooses the full model: (divorce = β_1 + β_2*unemployment + β_3*year + β_4*femlab + β_5*marriage + β_6*birth + β_7*military)

**(b) (5 pts) Use the first 70 years of data as the training set and the final 7 years as the test set. Randomly shuffle the rows of the training datset and then use AIC, BIC, and 5-fold CV (minimizing validation MSPE) to find the best subset of the predictors. Then compare the selected model to the full model that uses all of the explanatory variables by fitting both to the training data, and then use both to predict the responses for the subjects in the test set. Report the average squared distance between the predictions and the actual response values in the test set for both models. Do these models tend to overestimate the response for the subjects in the test set?**

```
set.seed(3301)
X.tr <- X[1:70,]
y.tr <- y[1:70]
X.va <- X[-7,]
y.va <- y[-7]
ideal <- get.best.ic(X.tr, y.tr, 5, 2, c(1,1,1,1,1,1,1))
print(ideal)
```

```
##      aic_beta final_aic bic_beta final_bic mspe.cap_beta final_mspe.cap
## [1,]        1  264.5592        1  280.2986             1       2.537159
## [2,]        1  264.5592        1  280.2986             1       2.537159
## [3,]        0  264.5592        0  280.2986             1       2.537159
## [4,]        1  264.5592        1  280.2986             1       2.537159
## [5,]        1  264.5592        1  280.2986             1       2.537159
## [6,]        1  264.5592        1  280.2986             1       2.537159
## [7,]        1  264.5592        1  280.2986             1       2.537159
```

```
results <- compare.ic(get.ic(X.tr,y.tr,5,c(1,1,1,1,1,1,1)),ideal)
print(results)
```

```
##      aic_beta final_aic bic_beta final_bic mspe.cap_beta final_mspe.cap
## [1,]        1  264.5592        1  280.2986             1       2.537159
## [2,]        1  264.5592        1  280.2986             1       2.537159
## [3,]        0  264.5592        0  280.2986             1       2.537159
## [4,]        1  264.5592        1  280.2986             1       2.537159
## [5,]        1  264.5592        1  280.2986             1       2.537159
## [6,]        1  264.5592        1  280.2986             1       2.537159
## [7,]        1  264.5592        1  280.2986             1       2.537159
```

```
full.beta <- qr.solve(crossprod(X.tr),crossprod(X.tr,y.tr))
rss.full <- sum((y.va - X.va %*% full.beta)^2)/7
ideal.X <- cbind(1,data$year,data$femlab,data$marriage,data$birth, data$military) # Once code works, use code to create ideal X.
ideal.X.tr <- ideal.X[1:70,]
ideal.X.va <- ideal.X[-7,]
ideal.beta <- qr.solve(crossprod(ideal.X.tr),crossprod(ideal.X.tr,y.tr))
rss.ideal <- sum((y.va - ideal.X.va %*% ideal.beta)^2)/7
print(paste("Full model average square error:", rss.full))
```

```
## [1] "Full model average square error: 23.1384935725717"
```

```
print(paste("Selected model average square error:", rss.ideal))
```

```
## [1] "Selected model average square error: 23.3268604685585"
```

It appears that the selected model has a higher average square error than the full model. This may be the case since, it appears our models overestimate the responses of the subjects. The average error in our sample data were around 2 and 3, whereas both of the errors for these models in predicting the remaining 7 validation values are around 23 and 24, which is a significant difference.

## 2. (5pts) The following variables were measured for n = 31 trees:

## • D, tree diameter in inches;

## • H, tree height in feet;

## • V, volume of timber produced in cubic feet.

# The dataset is in the file “trees.txt” posted on canvas. Use log(V), the natural logarithm of the volume of timber produced, as the response and let

$$\{D, D^2, H, H^2, D * H\}$$

###be the full set of explanatory variables in a linear regression model. Randomly shuffle the rows of the dataframe and then use AIC, BIC, and 5-fold CV (minimizing validation MSPE) to find the best subset of these predictors. Only select from subsets that respect the hierarchy of terms.

```
trees <- read.table("trees.txt",header=TRUE)
head(trees)
```

```
##      D  H    V
## 1  8.3 70 10.3
## 2  8.6 65 10.3
## 3  8.8 63 10.2
## 4 10.5 72 16.4
## 5 10.7 81 18.8
## 6 10.8 83 19.7
```

```
set.seed(3301)
D <- trees$D
H <- trees$H
V <- trees$V

full.X <- cbind(1, D, D^2, H, H^2, D*H)
best.model <- compare.ic(get.ic(full.X,log(V),5,c(1,1,1,1,1,1)), get.ic(full.X[,-6],l
og(V),5,c(1,1,1,1,1,0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-6:-5],log(V),5,c(1,1,1,1,0,0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-c(3,6)],log(V),5,c(1,1,0,1,1,
0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-c(3,5,6)],log(V),5,c(1,1,0,1,0,
0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-c(3,5,6)],log(V),5,c(1,1,0,1,0,
0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-6:-4],log(V),5,c(1,1,1,0,0,0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-c(2,3,6)],log(V),5,c(1,0,0,1,1,
0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-6:-3],log(V),5,c(1,1,0,0,0,0)))
best.model <- compare.ic(best.model, get.ic(full.X[,-c(2,3,5,6)],log(V),5,c(1,0,0,1,
0,0)))
best.model <- compare.ic(best.model, get.ic(cbind(full.X[,-6:-2]),log(V),5,c(1,0,0,0,
0,0)))
print(best.model)
```

```
##      aic_beta final_aic bic_beta final_bic mspe.cap_beta final_mspe.cap
## [1,]        1 -61.31366        1 -54.14373             1    0.008321643
## [2,]        1 -61.31366        1 -54.14373             1    0.008321643
## [3,]        1 -61.31366        1 -54.14373             1    0.008321643
## [4,]        1 -61.31366        1 -54.14373             1    0.008321643
## [5,]        0 -61.31366        0 -54.14373             1    0.008321643
## [6,]        0 -61.31366        0 -54.14373             1    0.008321643
```

From this, it appears that the model chosen by AIC and BIC was:

$$\{1, D, D^2, H\}$$

OR

$$\log(V) = \beta_1 + \beta_2 D + \beta_3 D^2 + \beta_4 H$$

and the model chosen by mspe is:

$$\{1, D, D^2, H, H^2, D * H\}$$

OR

$$\log(V) = \beta_1 + \beta_2 D + \beta_3 D^2 + \beta_4 H + \beta_5 H^2 + \beta_6 (D * H)$$

# (7pts) An experiment was conducted where paper brightness (measured by a reflectance meter) was recorded for paper samples produced using 4 different settings (a, b, c, d). The data are in the file “paper.txt” posted on canvas

```
paper <- read.table("paper.txt",header=TRUE)
head(paper)
```

```
##   bright operator
## 1   59.8        a
## 2   60.0        a
## 3   60.8        a
## 4   60.8        a
## 5   59.8        a
## 6   59.8        b
```

**(a) (4pts) Is there statistical evidence, at the 1% significance level, that the predictor operator is significant in the linear regression model with response bright? Perform the appropriate hypothesis test and state the assumptions that you are making about the data.**

I am making no assumptions about the data.

```
X <- cbind(paper$operator == 'a', paper$operator == 'b', paper$operator == 'c', paper
$operator == 'd')
y <- paper$bright
beta <- qr.solve(crossprod(X),crossprod(X,y))
null.X <- cbind(paper$operator == 'a') + cbind(paper$operator == 'b') + cbind(paper$o
perator == 'c') + cbind(paper$operator == 'd')
null.beta <- qr.solve(crossprod(null.X),crossprod(null.X,y))
rssf <- sum((y - X %*% beta)^2)
rss0 <- sum((y - null.X %*% null.beta)^2)

F.stat <- ((rss0 - rssf)/(dim(X)[2]-dim(null.X)[2]))/(rssf/(dim(X)[1]-dim(X)[2]))
P.value <- 1 - pf(F.stat, dim(X)[2]-dim(null.X)[2], dim(X)[1]-dim(X)[2])
print(P.value)
```

```
## [1] 0.0226089
```

At the 1% level of significance, we fail to reject the null hypothesis–that is, our model that predicts paper brightness with no predictors–in favor of the alternative hypothesis, being our model that predicts paper brightness using the paper type, or “operators”.

**(b) (3pts) Use AIC and BIC to select between the model with the predictor operator and the model with no predictors (the intercept-only model). How do these selections compare to the hypothesis test result in part 3a?.**

```
n <- dim(X)[1]
common <- n * log(2*pi) + n
AIC.full <- common + n * log(rssf/n) + 2 * (dim(X)[2] + 1)
BIC.full <- common + n * log(rssf/n) + log(n) * (dim(X)[2] + 1)
AIC.null <- common + n * log(rss0/n) + 2 * (dim(cbind(null.X))[2] + 1)
BIC.null <- common + n * log(rss0/n) + log(n) * (dim(cbind(null.X))[2] + 1)
print (paste("AIC full:", AIC.full))
```

```
## [1] "AIC full: 17.4554608783505"
```

```
print (paste("BIC full:", BIC.full))
```

```
## [1] "BIC full: 22.4341222461205"
```

```
print (paste("AIC null:", AIC.null))
```

```
## [1] "AIC null: 23.0800461654697"
```

```
print (paste("BIC null:", BIC.null))
```

```
## [1] "BIC null: 25.0715107125777"
```

As one can observe, AIC and BIC both favor the full model. This contradicts the model favored by the hypothesis testing in 3a, however, our level of significance was fairly low, making the rejection of a poor model unlikely. That is, our type two error was high, serving as a trade-off for our type one error being low. If we had instead chosen a 5% level of significance, we would see our answers in part 3b and 3a align.