Satinitigan_Karl_HW3

Karl Satinitigan

2/8/2020

MACS30100

Conceptual Exercises

1 - Generating a data set

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.2.1 --
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(tidymodels)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
## -- Attaching packages -------------------------------------------------- tidymodels 0.0.3 --
## v broom     0.5.4     v recipes   0.1.9
## v dials     0.0.4     v rsample   0.0.5
## v infer     0.5.1     v yardstick 0.0.4
## v parsnip   0.0.5
## -- Conflicts ----------------------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x dials::margin()   masks ggplot2::margin()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
```

```r
library(ggplot2)
library(dplyr)
library(leaps)

set.seed(1234)

x <- matrix(rnorm(1000*20),1000,20)

b <- rnorm(20)
```

```r
b[14] <- 0
b[15] <- 0
b[18] <- 0

e <- rnorm(1000)

y <- x%*%b + e
```

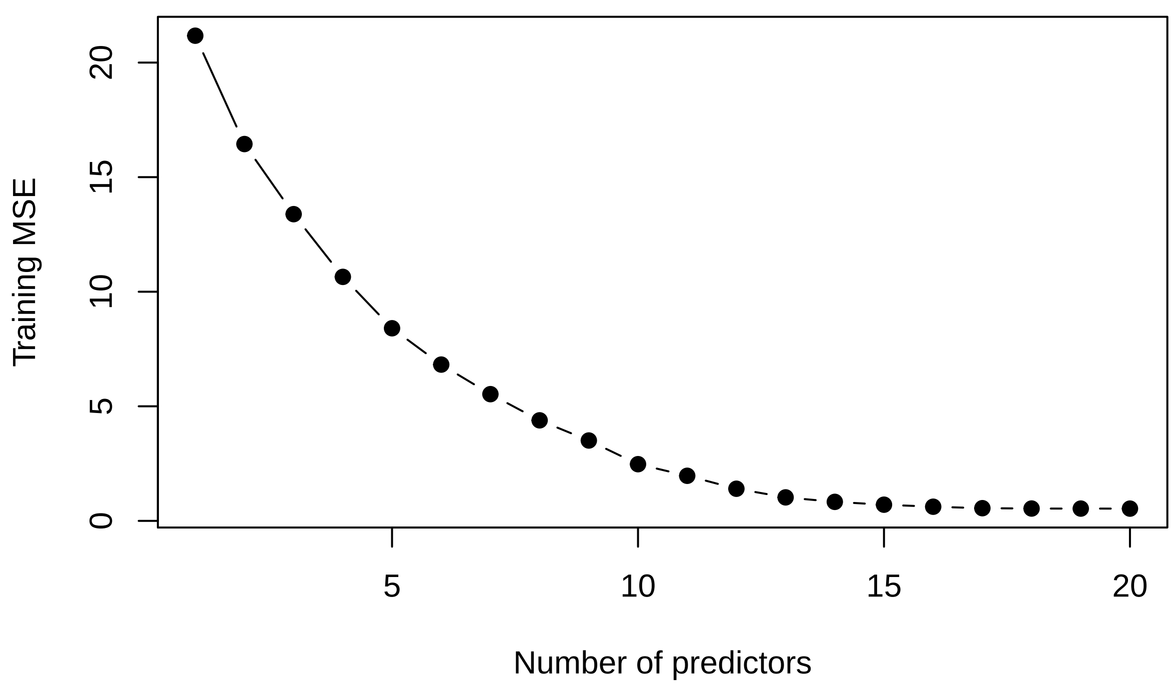
2 - Splitting into training set and test set

```r
train <- sample(seq(1000),100,replace=FALSE)
test <- (-train)

x.train <- x[train,]
y.train <- y[train]
x.test <- x[test,]
y.test <- y[test]
```

3 - Performing best subset selection

```r
trainset <- data.frame(y = y.train, x = x.train)
regfit <- regsubsets(y ~ ., data = trainset, nvmax = 20)
trainmat <- model.matrix(y ~ ., data = trainset, nvmax = 20)
errors <- rep(NA, 20)
for (i in 1:20) {
    coefi <- coef(regfit, id = i)
    pred <- trainmat[, names(coefi)] %*% coefi
    errors[i] <- mean((pred - y.train)^2)
}
plot(errors, xlab = "Number of predictors", ylab = "Training MSE", pch = 19, type = "b")
```



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

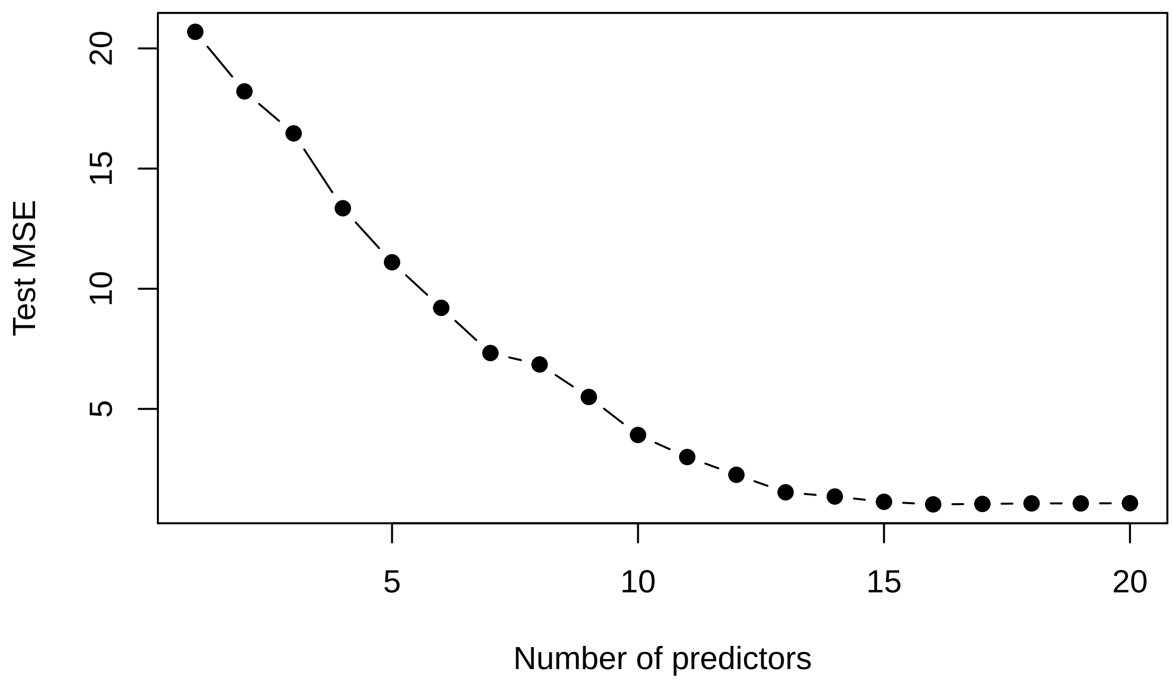
```
## [1] 20
```

The training set MSE takes its minimum value for model size 20.

4 - Plotting the test set MSE

```r
testset <- data.frame(y = y.test, x = x.test)
testmat <- model.matrix(y ~ ., data = testset, nvmax = 20)
errors <- rep(NA, 20)

for (i in 1:20) {
    coefi <- coef(regfit, id = i)
    pred <- testmat[, names(coefi)] %*% coefi
    errors[i] <- mean((pred - y.test)^2)
}
plot(errors, xlab = "Number of predictors", ylab = "Test MSE", pch = 19, type = "b")
```



5 - Model size with minimum value

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

```
## [1] 16
```

The training set MSE takes its minimum value for model size 16.

6 - Comparing to the true model

```r
coef(regfit,which.min(errors))
```

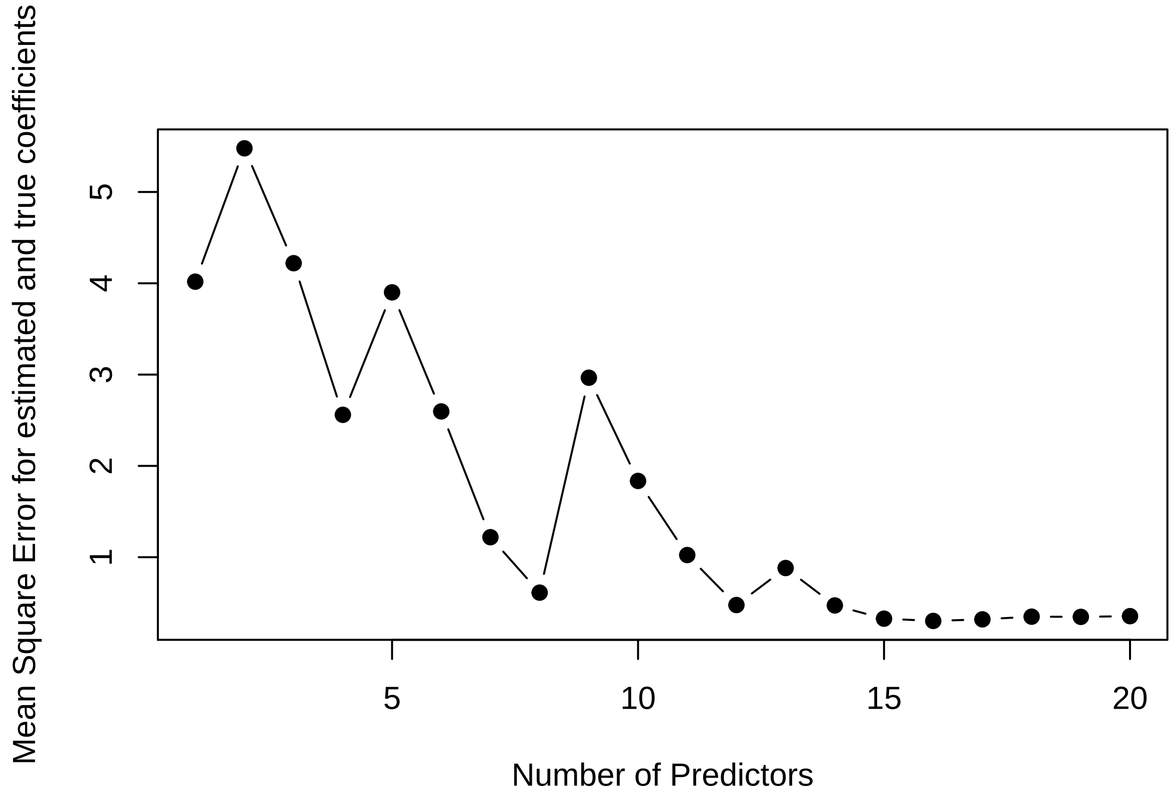
```
## (Intercept)          x.1          x.2          x.3          x.4          x.5
```

```
## -0.06330569 -1.74878421 -0.93805659 -0.77277768  0.29412205 -1.17146767
##         x.6         x.7         x.8         x.9        x.10        x.11
## -1.62084951  0.87137895  1.70268183  1.30937019 -1.22912918 -1.21104633
##        x.12        x.16        x.17        x.19        x.20
##  0.79828148  1.13748628 -0.40189392 -0.40740412  1.33216613
```

The best subset model was able to identify the betas that I hand-coded (14, 15, and 18).

7 - Creating a new plot and comparing to test MSE plot

```
errors <- rep(NA, 20)
x_cols = colnames(x, do.NULL = FALSE, prefix = "x.")
for (i in 1:20) {
    coefi <- coef(regfit, id = i)
    errors[i] <- sqrt(sum((b[x_cols %in% names(coefi)] - coefi[names(coefi) %in% x_cols])^2) + sum(b[!(i
}
plot(errors, xlab = "Number of Predictors", ylab = "Mean Square Error for estimated and true coefficient
```




```
which.min(errors)
```

```
## [1] 16
```

In this plot, the MSE fluctuates as the number of predictors increases.

Application Exercises

1 - Fitting a least squares linear model

```
library(readr)
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 3.0-2
```

```
gsstrain <- read_csv(url("https://raw.githubusercontent.com/ksatinitigan/problem-set-3/master/data/gss_t
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )

## See spec(...) for full column specifications.
```

```
gsstest <- read_csv(url("https://raw.githubusercontent.com/ksatinitigan/problem-set-3/master/data/gss_te
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
## See spec(...) for full column specifications.
```

```
linear <- lm(egalit_scale ~., gsstrain)
linearpred <- predict(linear, gsstest)
linearMSE <- mean((gsstest$egalit_scale - linearpred)^2)
linearMSE
```

```
## [1] 63.21363
```

The test MSE is 63.21363.

2 - Fitting a ridge regression model

```
gsstrainmatrix <- model.matrix(egalit_scale~., gsstrain)
gsstestmatrix <- model.matrix(egalit_scale~., gsstest)

grid <- 10^seq(4, -2, length=100)

ridge <- glmnet(gsstrainmatrix, gsstrain$egalit_scale, alpha=0, lambda=grid)

ridgeCV <- cv.glmnet(gsstrainmatrix, gsstrain$egalit_scale, alpha=0, lambda=grid)

bestlamridge <- ridgeCV$lambda.min

ridgelam <- glmnet(gsstrainmatrix, gsstrain$egalit_scale, alpha=0, lambda=bestlamridge)
coef(ridgelam)
```

```
## 79 x 1 sparse Matrix of class "dgCMatrix"
##                                     s0
## (Intercept)                29.67334628
## (Intercept)                 .
## age                        -0.03749510
## attend                     -0.02348204
## authoritarianism            0.01338599
## black                       1.27835016
## born                        0.33185281
## childs                      0.24312593
## colath                      0.34240727
## colrac                      0.10810689
## colcom                      0.01674451
## colmil                     -0.66802523
## colhomo                     0.83643970
## colmslm                     0.11822364
## con_govt                   -0.14767289
## evangelical                -0.19777369
## grass                      -1.71976317
## happy                       0.46951019
## hispanic_2                  0.31826913
## homosex                     0.08682238
## income06                   -0.10508600
## mode                        0.20351851
## owngun                      0.95715625
## polviews                   -1.37044190
## pornlaw2                   -0.35572891
## pray                        0.09175916
## pres08                     -3.51690696
## reborn_r                    0.04770485
## science_quiz               -0.10606680
## sex                         1.08542790
## sibs                        0.13723612
## social_connect              0.02960515
## south                      -0.34516780
## teensex                    -0.12716284
## tolerance                  -0.18899290
## tvhours                     0.19467872
## vetyears                   -0.26194907
## wordsum                    -0.03147217
## degree_HS                   0.10264465
## degree_Junior.Coll         -0.98337237
## degree_Bachelor.deg        -1.67812539
## degree_Graduate.deg         0.09869042
## marital_Widowed            -1.03630486
## marital_Divorced           -0.20458390
## marital_Separated          -0.53268012
## marital_Never.married       0.15400812
## news_FEW.TIMES.A.WEEK       0.24333676
## news_ONCE.A.WEEK            0.28411100
## news_LESS.THAN.ONCE.WK      0.15219378
## news_NEVER                  0.65175721
```

```
## partyid_3_Ind                    -1.37505067
## partyid_3_Rep                    -3.02194393
## relig_CATHOLIC                   -0.59684092
## relig_JEWISH                      0.46053598
## relig_NONE                       -0.42010164
## relig_OTHER                       0.73946814
## relig_BUDDHISM                   -0.16213120
## relig_HINDUISM                   -3.70949321
## relig_OTHER.EASTERN               1.39770256
## relig_MOSLEM.ISLAM                2.23480056
## relig_ORTHODOX.CHRISTIAN         -3.57953373
## relig_CHRISTIAN                  -0.25357166
## relig_NATIVE.AMERICAN            -1.63425950
## relig_INTER.NONDENOMINATIONAL     1.59994883
## social_cons3_Mod                  0.14759034
## social_cons3_Conserv             -0.06950727
## spend3_Mod                        0.61393225
## spend3_Liberal                    1.56496383
## zodiac_TAURUS                     0.54733663
## zodiac_GEMINI                    -0.21527568
## zodiac_CANCER                     0.57978415
## zodiac_LEO                        0.22171970
## zodiac_VIRGO                      0.97625575
## zodiac_LIBRA                     -0.09149974
## zodiac_SCORPIO                   -0.63439762
## zodiac_SAGITTARIUS               -0.16578600
## zodiac_CAPRICORN                  0.26376878
## zodiac_AQUARIUS                   0.81960628
## zodiac_PISCES                    -0.33648753
```

```r
predictridge <- predict(ridge, s=bestlamridge, newx=gsstestmatrix)

ridgeMSE <- mean((gsstest$egalit_scale - predictridge)^2)
ridgeMSE
```

```
## [1] 61.01179
```

The test MSE is 61.01179.

3 - Fitting a lasso regression model

```r
lasso <- glmnet(gsstrainmatrix, gsstrain$egalit_scale, alpha=0, lambda=grid)
lassoCV <- cv.glmnet(gsstrainmatrix, gsstrain$egalit_scale, alpha=0, lambda=grid)

bestlamlasso <- lassoCV$lambda.min

lassolam <- glmnet(gsstrainmatrix, gsstrain$egalit_scale, alpha=0, lambda=bestlamlasso)
coef(lassolam)

predictlasso <- predict(lasso, s=bestlamlasso, newx=gsstestmatrix)

lassoMSE <- mean((gsstest$egalit_scale - predictlasso)^2)
lassoMSE
```

The test MSE is 60.90691. The number of nonzero coefficient estimates is 79.

4 - Fitting an elastic net regression model

```
gsstrainx <- model.matrix(egalit_scale ~ ., gsstrain)[, -1]
gsstrainy <- gsstrain$egalit_scale

gsstestx <- model.matrix(egalit_scale ~., gsstest)[, -1]
gsstesty <- gsstest$egalit_scale

for (i in seq(0, 1, .1))
elasticCV <- cv.glmnet(gsstrainx, gsstrainy, alpha=i)
bestlamelastic = elasticCV$lambda.min

elastic <- glmnet(gsstrainx, gsstrainy, alpha=1, lambda=bestlamelastic)
elastic$beta
```

```
## 77 x 1 sparse Matrix of class "dgCMatrix"
##                                      s0
## age                         -0.04120380
## attend                       .
## authoritarianism             .
## black                        0.91626909
## born                         .
## childs                       0.17709446
## colath                       .
## colrac                       .
## colcom                       .
## colmil                       .
## colhomo                      0.15166661
## colmslm                      .
## con_govt                     .
## evangelical                  .
## grass                       -1.34862059
## happy                        0.23938037
## hispanic_2                   .
## homosex                      .
## income06                    -0.11073956
## mode                         .
## owngun                       0.73124595
## polviews                    -1.55707080
## pornlaw2                     .
## pray                         .
## pres08                      -4.28030718
## reborn_r                     .
## science_quiz                -0.06237421
## sex                          0.98544135
## sibs                         0.10086576
## social_connect               .
## south                        .
## teensex                      .
## tolerance                   -0.20201221
## tvhours                      0.18415004
## vetyears                    -0.08424743
## wordsum                      .
## degree_HS                    .
```

```
## degree_Junior.Coll            -0.46681980
## degree_Bachelor.deg           -1.69157837
## degree_Graduate.deg             .
## marital_Widowed               -0.16166997
## marital_Divorced                .
## marital_Separated               .
## marital_Never.married           .
## news_FEW.TIMES.A.WEEK           .
## news_ONCE.A.WEEK                .
## news_LESS.THAN.ONCE.WK          .
## news_NEVER                     0.13882807
## partyid_3_Ind                 -1.01788543
## partyid_3_Rep                 -2.59244801
## relig_CATHOLIC                  .
## relig_JEWISH                    .
## relig_NONE                      .
## relig_OTHER                     .
## relig_BUDDHISM                  .
## relig_HINDUISM                -1.74866732
## relig_OTHER.EASTERN             .
## relig_MOSLEM.ISLAM              .
## relig_ORTHODOX.CHRISTIAN      -0.09591758
## relig_CHRISTIAN                 .
## relig_NATIVE.AMERICAN           .
## relig_INTER.NONDENOMINATIONAL   .
## social_cons3_Mod                .
## social_cons3_Conserv            .
## spend3_Mod                     0.13365960
## spend3_Liberal                 1.28308489
## zodiac_TAURUS                   .
## zodiac_GEMINI                   .
## zodiac_CANCER                   .
## zodiac_LEO                      .
## zodiac_VIRGO                   0.40403642
## zodiac_LIBRA                    .
## zodiac_SCORPIO                -0.22315268
## zodiac_SAGITTARIUS              .
## zodiac_CAPRICORN                .
## zodiac_AQUARIUS                0.22912649
## zodiac_PISCES                   .
```

```r
predictelastic <- predict(elastic, s=bestlamelastic, newx = gsstestx)
elasticMSE <- mean((predictelastic - gsstesty)^2)
elasticMSE
```

```
## [1] 61.22212
```

The test MSE is 61.19281. The number of nonzero coefficient estimates are 29.

5 - Comments

The lasso regression model yielded the lowest test MSE, so it is best at predicting an individual's egalitarianism. However, there is not much difference among the test errors.