HW1

jiaying Ning

2/15/2021

Contents

In this exercise, we will predict solubility of compounds using their chemical structures.

Among the 228 predictors, 208 are binary variables that indicate the presence or absence of a particular chemical substructure, 16 are count features, such as the number of bonds or the number of bromine atoms, and 4 are continuous features, such as molecular weight or surface area.

```
#load package
library(pls)
library(readxl)
library(glmnet)
library(plotmo)
library(dplyr)
```

Load Data

```
training = read.csv("./data/solubility_train.csv")
test = read.csv("./data/solubility_test.csv")
```

```
training2 <- model.matrix(Solubility ~ ., training)[ ,-1]
test2 = model.matrix(Solubility ~ ., test)[ ,-1]
y <- training$Solubility
```

(a) Fit a linear model using least squares on the training data and calculate the meansquared error using the test data.

```
LinearMod <- lm(Solubility ~ .,
                data = training)
```

```
 pred_lm <- predict(LinearMod, test)
 mean((test$Solubility - pred_lm)^2)
```

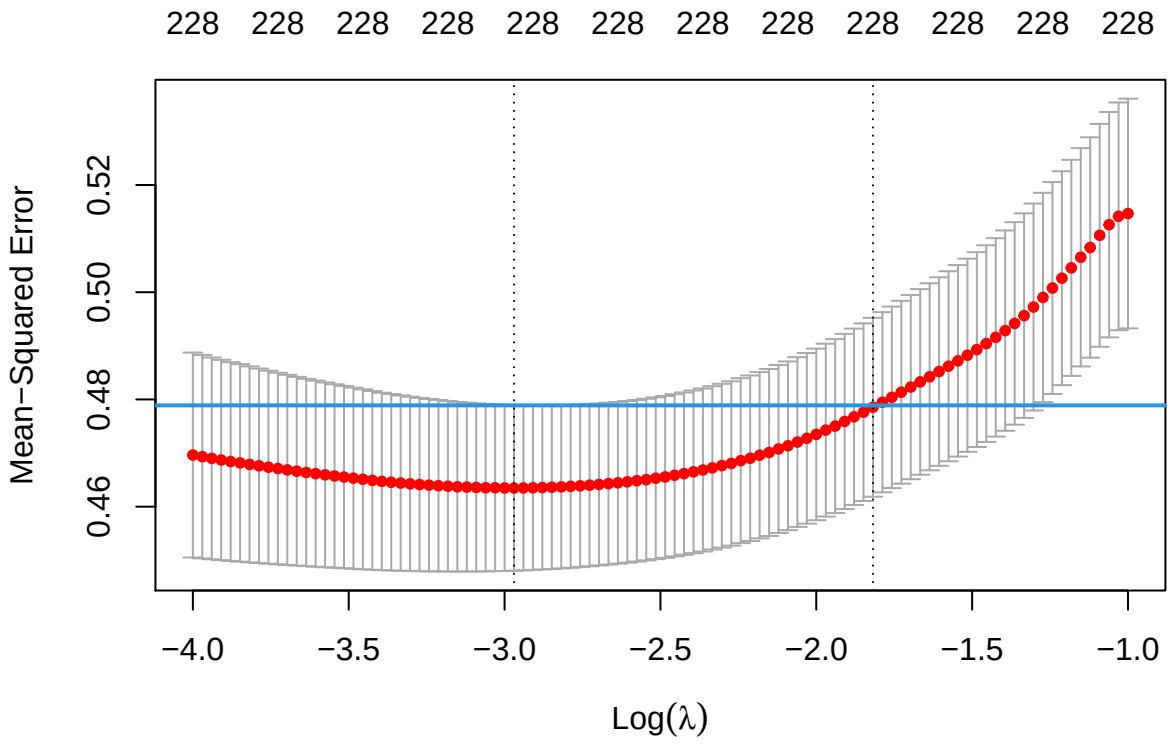
```
## [1] 0.5558898
```

MSE is 0.5558898

(b) Fit a ridge regression model on the training data, with lambda chosen by cross-validation.Report the test error.

chose lambda by cross-validation

```
set.seed(2761)
cv.ridge <- cv.glmnet(training2, y,
                      alpha = 0,
                      lambda = exp(seq(-1, -4, length = 100)))

plot(cv.ridge)
abline(h = (cv.ridge$cvm + cv.ridge$cvsd)[which.min(cv.ridge$cvm)], col = 4, lwd = 2)
```




```
# min CV MSE
cv.ridge$lambda.min
```

```
## [1] 0.05131886
```

```
# the 1SE rule
cv.ridge$lambda.1se
```

```
## [1] 0.1623206
```

fit chosen lambda into model to get the optimal coefficients

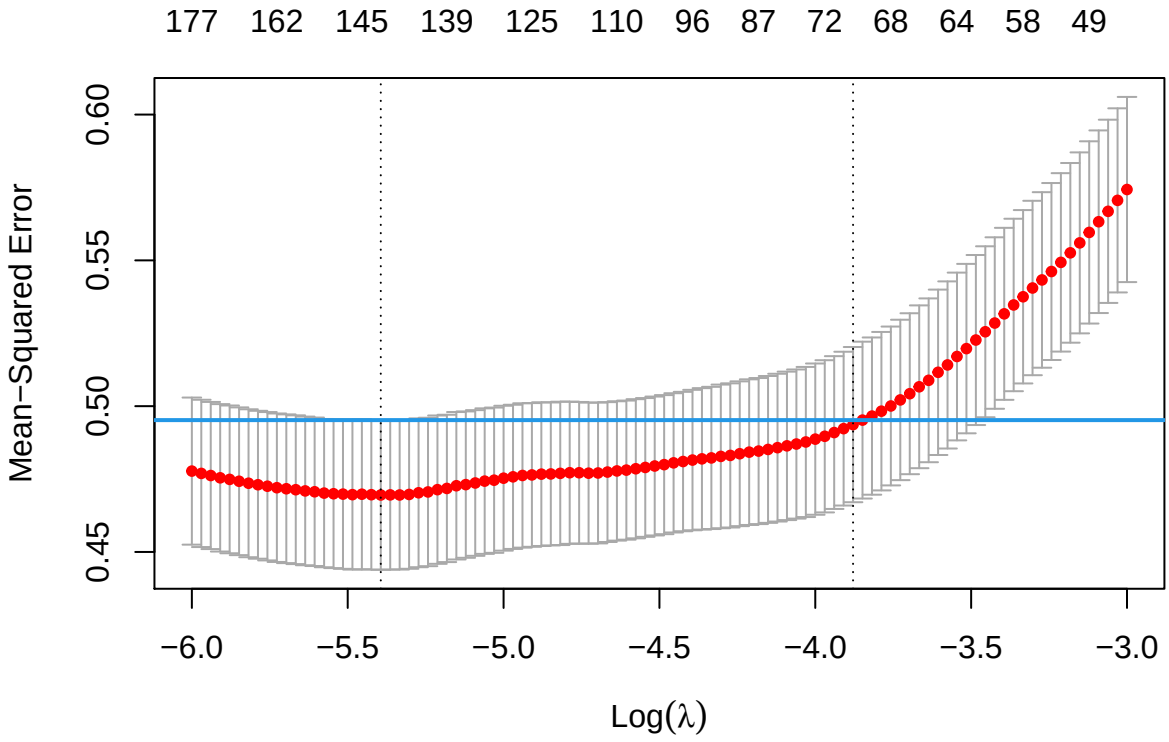
```
# make prediction
ridge_pred=(predict(cv.ridge, newx = test2,
             s = "lambda.min", type = "response"))

mean((ridge_pred - test$Solubility)^2)
```

```
## [1] 0.5148783
```

test error is 0.5148783

(c) Fit a lasso model on the training data, with lambda chosen by cross-validation. Report the test error and the number of non-zero coeficient estimates in your model.

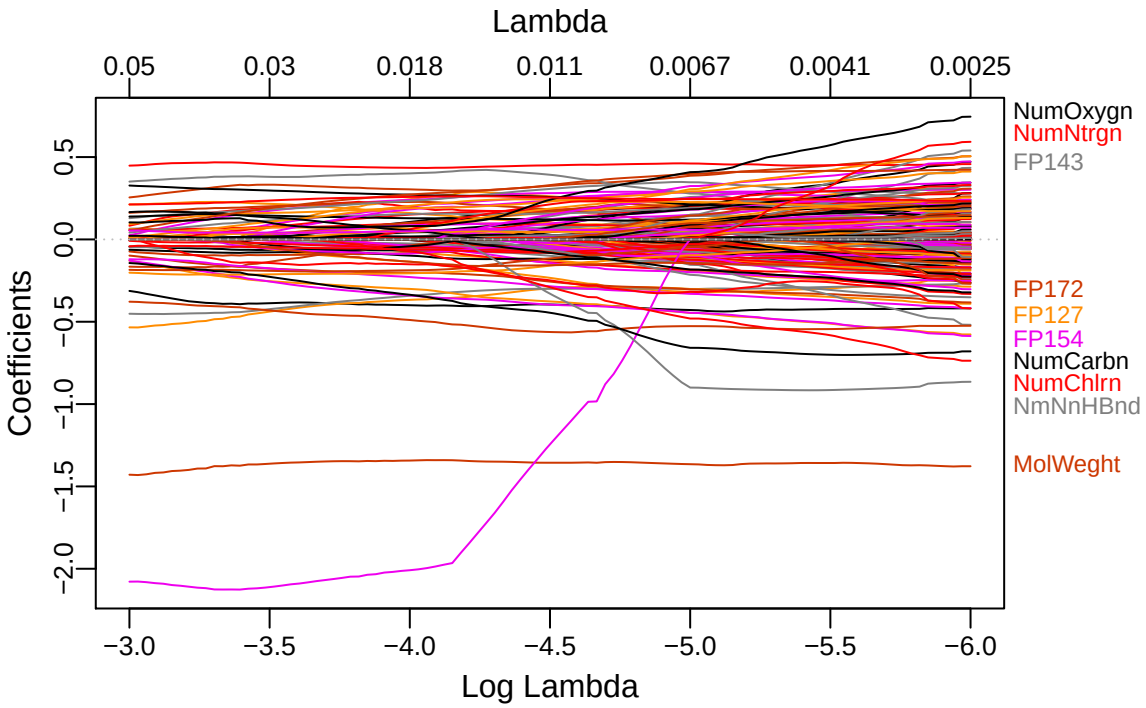
```
cv.lasso <- cv.glmnet(training2, y,
                      alpha = 1,
                      lambda = exp(seq(-3, -6, length = 100)))
plot(cv.lasso)
abline(h = (cv.lasso$cvm + cv.lasso$cvsd)[which.min(cv.lasso$cvm)], col = 4, lwd = 2)
```




```
cv.lasso$lambda.min
```

```
## [1] 0.004544037
```

```
plot_glmnet(cv.lasso$glmnet.fit)
```




```
Lasso_pred=predict(cv.lasso, newx = test2, s = "lambda.min", type = "response")

mean((Lasso_pred - test$Solubility)^2)
```

```
## [1] 0.4992432
```

There are total of 140 non-zero parameter and the test error is 0.4945886

(d) Fit a principle component regression model on the training data, with M chosen by cross-validation. Report the test error and the value ofM selected by cross-validation.

```
cv.mse <- RMSEP(pcr.mod)
ncomp.cv <- which.min(cv.mse$val[1,,])-1
ncomp.cv
```

```
## 157 comps
##       157
```

The value M selected by CV is 157 since it has the samllest mean squared error

```
predy2.pcr <- predict(pcr.mod, newdata = test,
                      ncomp = ncomp.cv)
# test MSE
mean((test$Solubility - predy2.pcr)^2)
```

```
## [1] 0.549917
```

test error is 0.549917 and the value of M selected by cross-validation is 157

(e) Which model will you choose for predicting solubility?

According to the test error:

- Linear model has test error = 0.5558898
- Ridge model has test error = 0.5148783
- Lasso model has test error = 0.4945886
- PCR model has test error = 0.549917

both ridge and lasso has relatively low test error compare to the rest of model, and I would agree applying regularization is appropriate since our data has relatively large number of predictors. Between Ridge and Lasso, I would prefer to use Lasso model. Lasso model not only give us a smaller test error, but also provide a "model selection" effect on current model so that we can continue further analyses with a simpler model, this is especially beneficial when a complex model like this is given to us.