Exam 1 Solution

Brian Lois

2/19/2017

Load data:

```
exam_data <- read.csv('/Users/lvi819/Documents/MIS6357/Exam1/exam1.csv')
head(exam_data)
```

```
##             X0          X1          X2       X3           X4           X5
## 1  0.090126329 -0.04349048  0.10519904 2.456486  0.073953494  0.007585155
## 2 -0.109376345 -0.02384645  0.05574954 2.466756  0.095154100  0.023597037
## 3 -0.016973727  0.03048841 -0.09786889 2.570967  0.144045499 -0.119975641
## 4 -0.082213968 -0.04667470  0.08447610 2.394790 -0.083633693 -0.029218147
## 5 -0.007516208 -0.08797188 -0.02742016 2.374455 -0.009912652  0.082361541
## 6  0.004077612 -0.03239408 -0.20619137 2.535279 -0.033664276  0.115801962
##             X6            X7          X8           X9         X11
## 1 -0.027954992  0.1207047862  0.20604884 -0.157234458 -0.06129247
## 2  0.165107413  0.1377656604 -0.10636297 -0.101966277 -0.09497321
## 3  0.064749024  0.0006588395 -0.01160461 -0.052113784 -0.02970340
## 4  0.010816096 -0.0403078447 -0.02598357  0.068544697  0.01851205
## 5  0.039654492 -0.0954583260 -0.01012752  0.001810257 -0.05790568
## 6  0.009813084  0.0567211290 -0.05477181 -0.021179433 -0.10488295
##            X12          X13         X14   X15         y
## 1 -0.035021519  0.034743057 -0.11816077  True -9.132878
## 2 -0.129437140  0.025867114  0.10913276 False -7.099861
## 3 -0.008609091 -0.024907947  0.08242524  True -2.459873
## 4  0.033606110  0.028315346 -0.01406532  True -1.902250
## 5  0.000850918 -0.067332811  0.10057746  True  1.613670
## 6  0.018721941 -0.006544613  0.00839267  True -9.729452
```

Center and scale:

```
for (col in colnames(exam_data)[1:14]){
  exam_data[col] <- (exam_data[[col]] - mean(exam_data[[col]])) / (sd(exam_data[[col]]))
}
```

Dummy variables:

```
exam_data[['X15_T']] <- as.integer(exam_data[['X15']] == 'True')
exam_data <- exam_data[ ,-15]
```

Test/train split:

```
train_indices <- createDataPartition(exam_data$y, p = .8, list = FALSE)
train_data <- exam_data[train_indices, ]
test_data <- exam_data[-train_indices, ]
nrow(train_data)
```

```
## [1] 80
```

```
nrow(test_data)
```

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

Cross validation:

```
four_folds <- createFolds(train_data$y, k = 4, list = TRUE, returnTrain = TRUE)
```

Define RMSE:

```
RMSE <- function(y, y_hat){
  return(sqrt(mean((y - y_hat)^2)))
}
```

Model fitting:

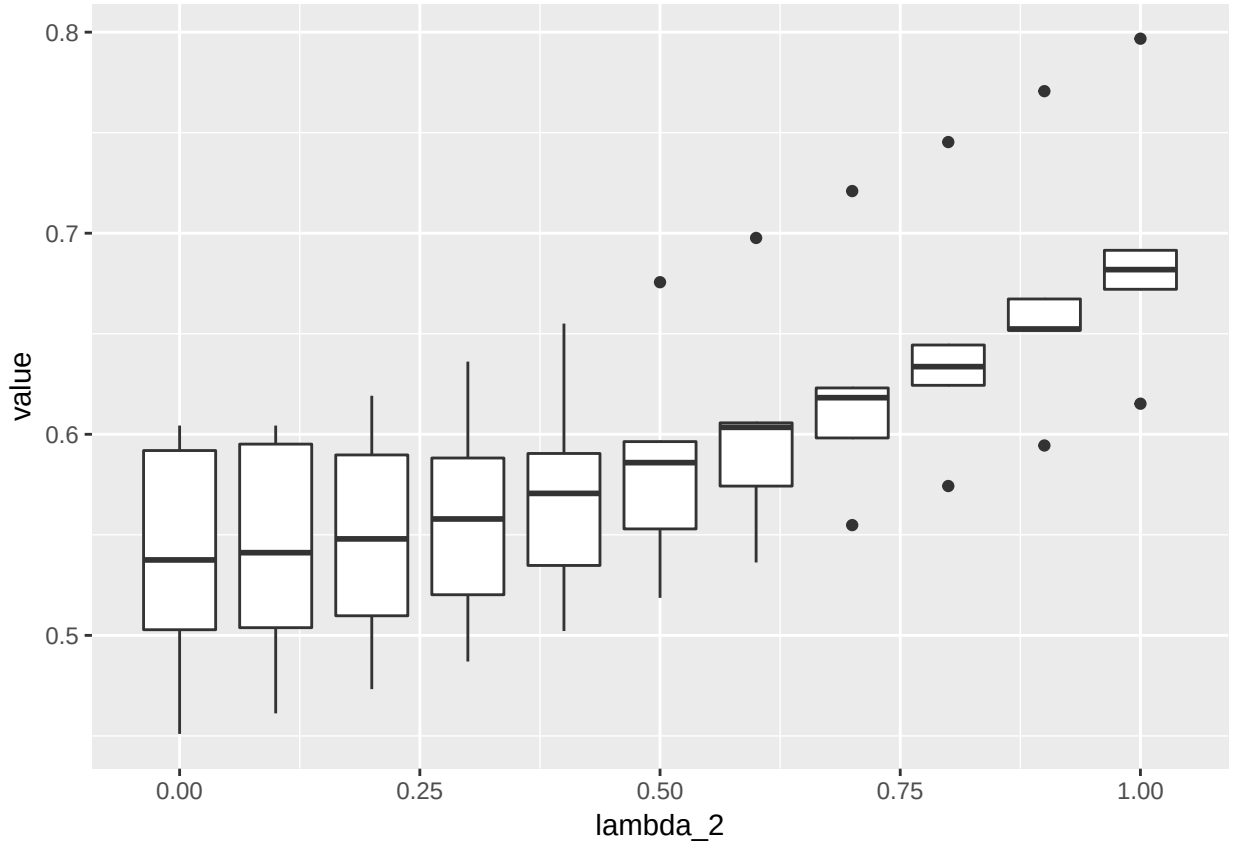
```
lambda_2_range <- seq(0, 1, .1)
tuning <- data.frame()
for (i in 1:length(lambda_2_range)){
  lam <- lambda_2_range[i]
  for (j in 1:4){
    model <- penalized(train_data[four_folds[[j]], 15],
                       as.matrix(train_data[four_folds[[j]], -15]),
                       lambda2 = lam,
                       model = 'linear')
    preds <- predict(model, as.matrix(train_data[-four_folds[[j]], -15]))
    error <- RMSE(train_data[-four_folds[[j]], 15], preds[,1])
    tuning[i,j] <- error
  }
}
```

Plot:

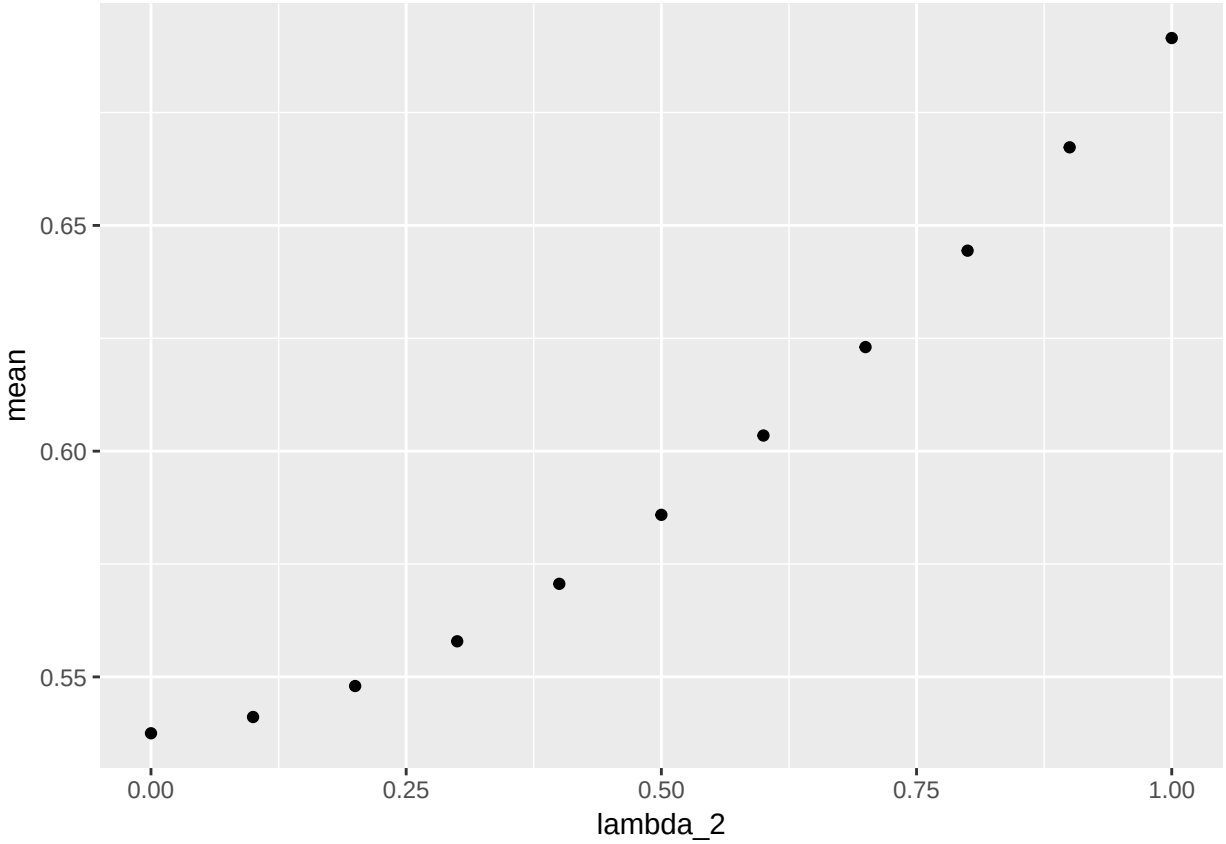
```
tuning <- cbind(lambda_2_range, tuning)
colnames(tuning) <- c('lambda_2', 'Fold 1 RMSE', 'Fold 2 RMSE', 'Fold 3 RMSE', 'Fold 4 RMSE')
tuning$mean <- apply(tuning[ ,2:5], 1, mean)
print(tuning)
```

```
##    lambda_2 Fold 1 RMSE Fold 2 RMSE Fold 3 RMSE Fold 4 RMSE      mean
## 1       0.0   0.6043556   0.5919190   0.5028127   0.4509870 0.5375186
## 2       0.1   0.5951173   0.6043345   0.5038290   0.4612126 0.5411234
## 3       0.2   0.5897363   0.6191640   0.5097471   0.4732792 0.5479817
## 4       0.3   0.5882110   0.6361527   0.5202085   0.4869969 0.5578923
## 5       0.4   0.5904673   0.6550528   0.5347699   0.5021805 0.5706176
## 6       0.5   0.5963624   0.6756297   0.5529450   0.5186550 0.5858980
## 7       0.6   0.6056947   0.6976662   0.5742417   0.5362581 0.6034652
## 8       0.7   0.6182176   0.7209646   0.5981906   0.5548419 0.6230537
## 9       0.8   0.6336561   0.7453472   0.6243623   0.5742736 0.6444098
## 10      0.9   0.6517217   0.7706561   0.6523760   0.5944350 0.6672972
## 11      1.0   0.6721261   0.7967519   0.6819011   0.6152213 0.6915001
```

```
long_tuning <- melt(tuning, id.var='lambda_2')
ggplot(data=long_tuning) + geom_boxplot(mapping = aes(x=lambda_2, y = value, group = lambda_2))
```




```
ggplot(data=tuning) + geom_point(mapping = aes(x = lambda_2, y = mean))
```



The error appears to be increasing in λ_2, so I will choose $\lambda_2 = 0$ for the final model.

Final model:

```
final_model <- penalized(train_data$y,
                         as.matrix(train_data[, -15]),
                         lambda2 = 0,
                         model = 'linear')
```

Test set evaluation:

```
test_preds <- predict(final_model, as.matrix(test_data[,-15]))[,1]
(test_rmse <- RMSE(test_data$y, test_preds))
```

```
## [1] 0.7201892
```

```
(test_R2 <- cor(test_preds, test_data$y))
```

```
## [1] 0.9985974
```