Regularization Lab

Grant Esparza

December 12, 2018

Questions to think about

1. Penalty term. Look at the parameter alpha under the Arguments section and the objective function under the Details section of the help file for glmnet.
   a. What piece of the puzzle are they calling the penalty term?

      The penalty term is alpha. When using lasso this value is set to 1 and 0 for ridge.

   b. What piece of the puzzle are they calling the objective function?

      The objective function is the minimization function that serves as our model.

2. Ridge versus Lasso
   a. What is the main difference between Ridge regression and Lasso?

      Ridge regression penalizes the coefficents using a squared term where Lasso uses an absolute value term. This allows Lasso to shrink coefficents down to zero, which serves as a variable selection feature.

   b. When might you be interested in one or the other?

      As mentioned above, it would be useful to use Lasso when attempting to identify unecessary variables. These variables would be reduced to zero indicating they do not improve the model.

3. Notice the ouput of `cv.glmnet` produces two specific values of λ. What is the difference?

   The value of `lambda.min` will provide a model with the smallest Mean Squared Error. The value of `lambda.1se` will provide a model with smaller coefficents since as the value of lambda increases, the coefficents shrink to zero.

Prediction

Read the data

```
bikes <- read.csv("https://roualdes.us/data/bike.csv")
```

MSE

```
MSE <- function(y, yhat) {
  mean((y - yhat)^2)
}
```

Create training and testing datasets

```
train_idx <- createDataPartition(bikes$cnt, p=0.75, list=FALSE)
training <- bikes[train_idx, ]
testing <- bikes[-train_idx, ]

## Create training inputs
train_x <- model.matrix(cnt ~ temp + as.factor(season), data=training)[, -1]
train_y <- training$cnt

## Create testing inputs
test_x <- model.matrix(cnt ~ temp + as.factor(season), data=testing)[, -1]
test_y <- testing$cnt
```

Make the models

```
# Create unregularized model
unreg <- lm(cnt ~ temp + as.factor(season), data=training)

# Create regularized model
fit <- cv.glmnet(train_x, train_y, nfolds=10, alpha=1)
```

Make predictions

```
MSE(test_y, predict(unreg, newdata=testing))
```

```
## [1] 1869633
```

```
MSE(test_y, predict(fit, s = fit$lambda.min, test_x))
```

```
## [1] 1870738
```

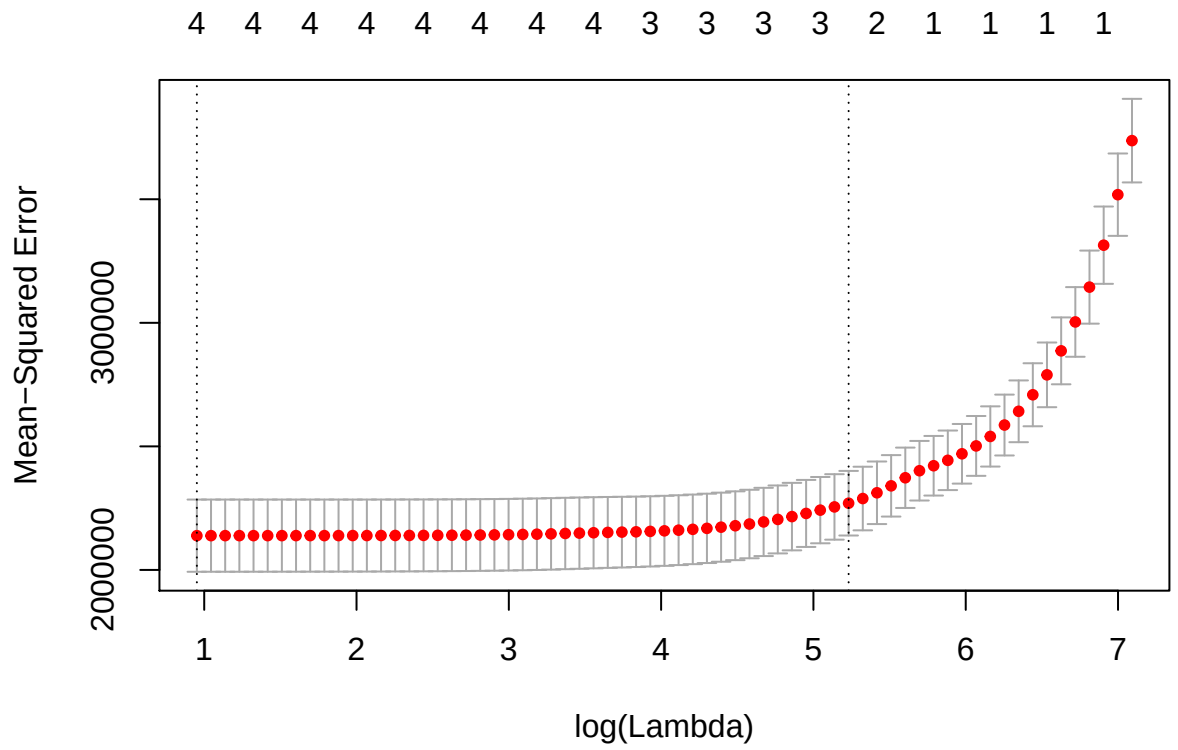
```
MSE(test_y, predict(fit, s = fit$lambda.1se, test_x))
```

```
## [1] 2115566
```

Here we can see that `lambda.min` produced the smallest MSE. The unregularized model actually had a smaller MSE than the model regularized with `lambda.1se`.

Plots

```
plot(fit)
```



This plot shows the Lasso model chose to keep three of my orignal four variables. However, I actually chose two variables, one of which had four categorical levels. It seems that summer does not contribute to reducing the MSE.

Overfitting

```
coef(unreg)
```

```
##        (Intercept)                temp as.factor(season)2
##            797.1419          6191.9905           808.2038
## as.factor(season)3 as.factor(season)4
##            520.8632          1264.2138
```

```
coef(fit)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)         1565.0750
## temp                5775.6065
## as.factor(season)2     .
## as.factor(season)3     .
## as.factor(season)4   338.1141
```

I am surprised by `cv.glmnet`'s decision to minimize the summer season. I would have guessed the summer season would be significant in terms of bike rental activity. I can believe spring has an effect on bike rentals, since people tend to enjoy the new warm weather. It also makes sense that the cold of winter would be have an impact on bike rentals.

I found it surprising that spring's coefficent was reduced by so much, however this is most likely because spring doesn't explain enough of the variance in bike rentals.