LogReg

Michael Frasco

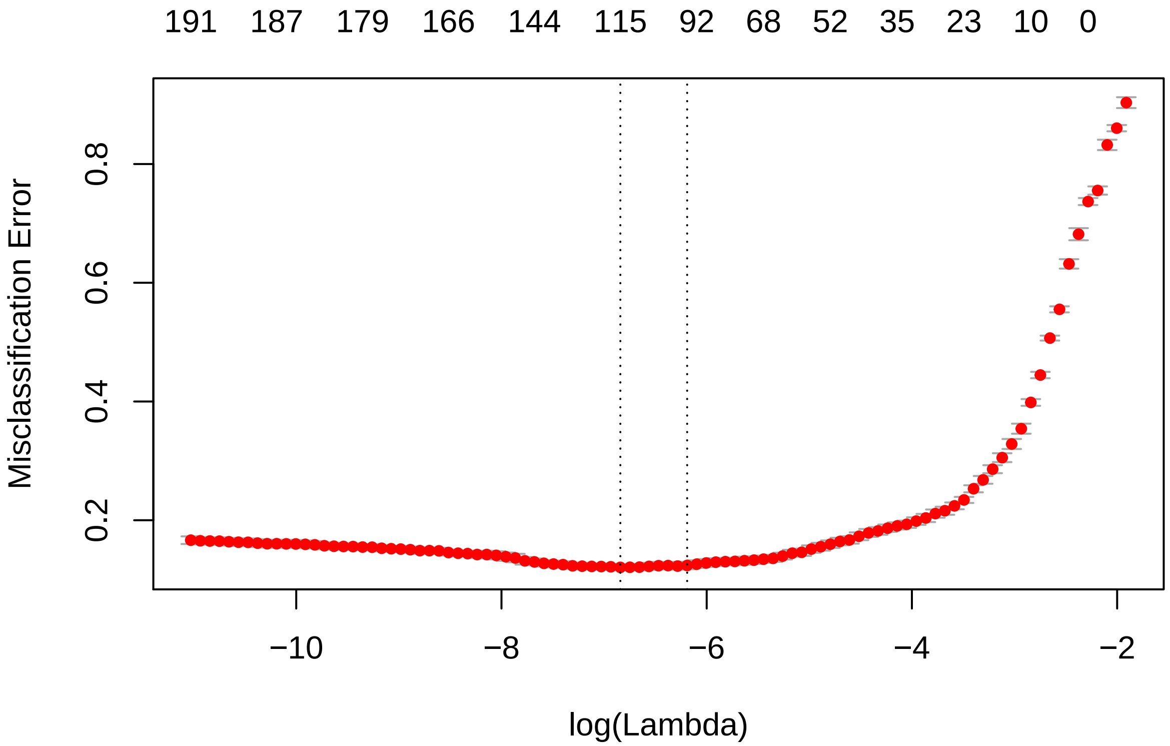
May 10, 2015

Results: We used the glmnet package to achive an accuracy of about 88% on the unseen test data. We trained our model using 10-fold cross validation to select the value of the regularization parameter that minimized the mis-classification error.

```
## Loaded glmnet 2.0-2
```

Since the glmnet package comes with a cross validation function, we do not need to seperate the 5,000 images in the training data into a 4,000 image set and a 1,000 image set for the validation data. The glmnet already does this by performing k-fold cross validation over a sequence of regularization parameters.

```
cv.fit <- cv.glmnet(x=training.data, y=as.factor(training.label),
                    family="multinomial", type.measure="class")
plot(cv.fit)
```



The plot above is the cross-validated misclassification error after running the cv.glmnet() function. The right most vertical dashed line represents the value of lambda that minimizes the error. The left vertical dashed line represents the value of lambda that should be used to avoid over-fitting. This is calculated by finding the largest value of lambda such that the misclassification error is within 1 standard error of the minimum. If we feared overfitting, we should choose the larger value of lambda. However, since we know that the pictures in the test data are of the same form as the pictures in the training data (i.e. similar variance drawn from the same distribution), we can choose the smaller lambda.

```
print(cv.fit$lambda.min)
```

```
## [1] 0.001069
```

```
print(cv.fit$lambda.1se)
```

```
## [1] 0.002049
```

We now use the smaller value of lambda to get the error of this model on the test data.

```
preds = as.numeric(predict(cv.fit, test.data,
                           s="lambda.min", type="class"))
print(mean(preds == test.label))
```

```
## [1] 0.8735
```

We achive an accuracy of about 88%.