SAGA Test Problems

Luofeng Liao

March 13, 2018

Easy

Use `glmnet::cv.glmnet` to compute a L1-regularized linear model of the spam data in library(ElemStatLearn). What features are selected for the prediction function?

```
library(ElemStatLearn)
library(glmnet)
library(pROC)
```

```
sum(is.na(spam)) # make sure data is valid
```

```
## [1] 0
```

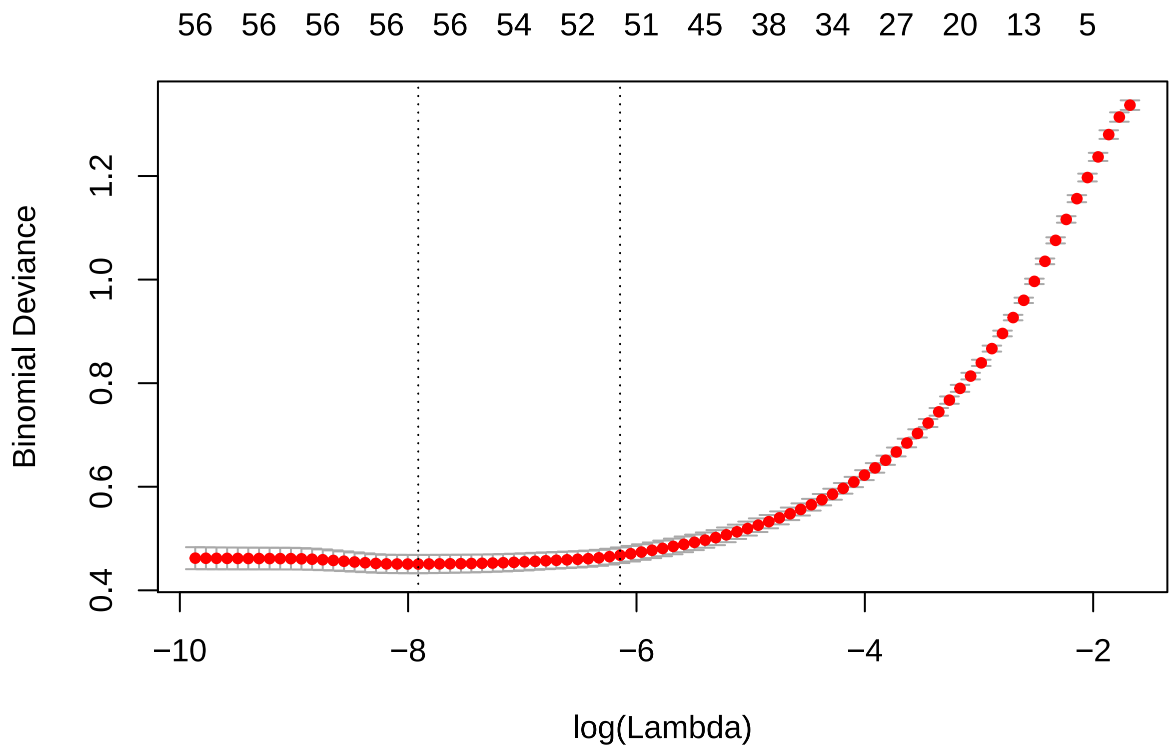
```
p <- 57 # Each email is represented by 57 features
n <- length(spam$spam)

set.seed(123)
class <- factor(spam$spam, labels=c(0,1)) # email = 0, spam = 1

trainIdx <- sort(sample(1:n, floor(n*0.75)))
train_class <- spam$spam[trainIdx]
train_feat <- data.matrix(spam[trainIdx,1:p])
test_class <- spam$spam[-trainIdx]
test_feat <- data.matrix(spam[-trainIdx,1:p])


logistic.fitted <- cv.glmnet(train_feat,train_class,
                             family = "binomial",
                             alpha=1)

plot(logistic.fitted)
```




```r
# extract coefficients
coefficients <- coef(logistic.fitted, s="lambda.1se")
names(coefficients[which(coefficients != 0),])
```

```
##  [1] "(Intercept)" "A.1"         "A.2"         "A.3"         "A.4"
##  [6] "A.5"         "A.6"         "A.7"         "A.8"         "A.9"
## [11] "A.10"        "A.12"        "A.13"        "A.14"        "A.15"
## [16] "A.16"        "A.17"        "A.18"        "A.19"        "A.20"
## [21] "A.21"        "A.22"        "A.23"        "A.24"        "A.25"
## [26] "A.26"        "A.27"        "A.28"        "A.29"        "A.30"
## [31] "A.31"        "A.33"        "A.35"        "A.36"        "A.39"
## [36] "A.40"        "A.41"        "A.42"        "A.43"        "A.44"
## [41] "A.45"        "A.46"        "A.47"        "A.48"        "A.49"
## [46] "A.50"        "A.51"        "A.52"        "A.53"        "A.54"
## [51] "A.56"        "A.57"
```

To reduce complexity of the model, I choose the lambda one SE away from the lambda.min.

What is the test error and test AUC of the learned model?

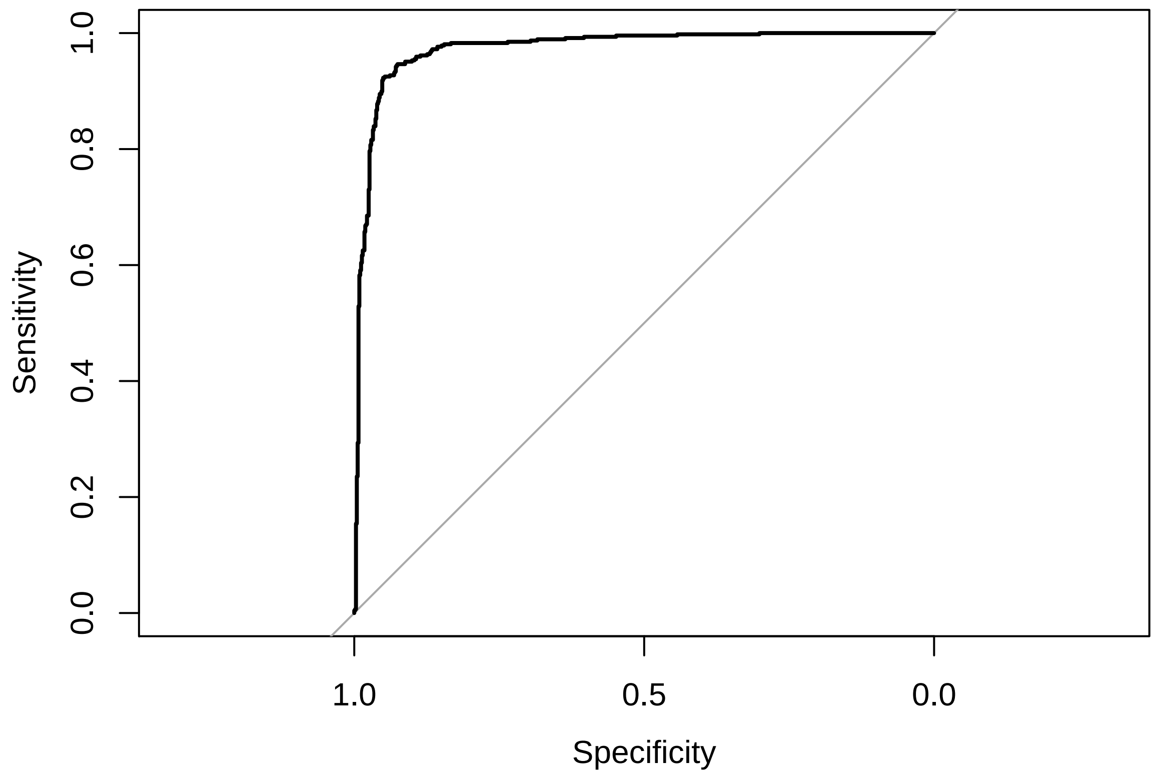
```r
test_pred <- predict(logistic.fitted, test_feat, s="lambda.1se")

roc_lasso <- roc(test_class, test_pred)
```

```
## Warning in roc.default(test_class, test_pred): Deprecated use a matrix
## as predictor. Unexpected results may be produced, please pass a numeric
```

```
## vector.
```

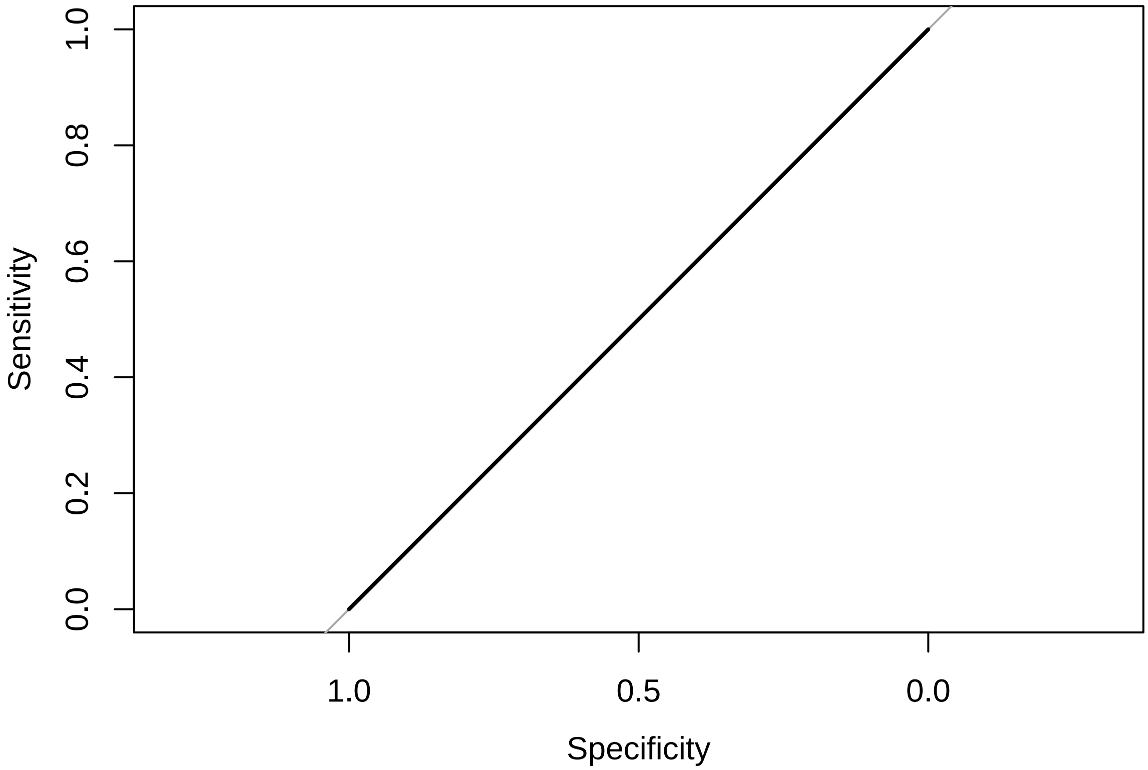
```r
plot(roc_lasso)
```




```r
auc(roc_lasso)
```

```
## Area under the curve: 0.974
```

```r
const_pre <- rep(0,length(test_class))
roc_const <- roc(test_class, const_pre)
plot(roc_const)
```




```
auc(roc_const)
```

```
## Area under the curve: 0.5
```

Is it significantly better than the trivial model which predicts the most frequent class in the training data? Answer these questions by using K-fold cross-validation in the spam data.

Yes, it is.