Outline

Training and Test error

Important distinction:

- **Training error** is the empirical risk

$$n^{-1} \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

  For 0-1 loss in classification, this is the misclassification error on the training data, **which were used in fitting** $\hat{y}$.

- **Test error** is the empirical risk on new, previously unseen, observations

$$m^{-1} \sum_{i=1}^{m} L(y_i, \hat{y}_i)$$

  which were NOT used in fitting.

The test error is in general larger than the training error (as we are fitting partially noise – depending on the complexity of the classifier). It is a much better gauge of how well the method will do on future data.

Success rate is calculated on the same data that the GLM is trained on!
Separate in training and test set.

```
n <- length(Y)
intrain <- sample( rep(c(TRUE,FALSE),each=n/2) ,
                              round(n/2) ,replace=TRUE )
train <- (1:n)[intrain]
test  <- (1:n)[!intrain]
```

Fit only on training set and predict on both training and test set.

```
gl <- glm(Y[train] ~ ., data=X[train,],family=binomial)

proba_train <- predict(gl,newdata=X[train,],type="response")
proba_test  <- predict(gl,newdata=X[test,],type="response")

predicted_spam_train <- as.numeric(proba_train > 0.95)
predicted_spam_test  <- as.numeric(proba_test > 0.95)
```

Results for training and test set:

```
> table(predicted_spam_train, Y[train])
predicted_spam_train    0    1
                   0 1403  354
                   1   11  567


> table(predicted_spam_test, Y[test])
predicted_spam_test    0    1
                  0 1346  351
                  1   28  541
```

Its no coincidence that the success rate is worse on the test data.

Compare with LDA.

```
library(MASS)
ldares <- lda(x=X[train,],grouping=Y[train])
```

With following result

```
> Call:
lda(X, grouping = Y)

Prior probabilities of groups:
        0         1
0.6059552 0.3940448

...
```

```
...

Coefficients of linear discriminants:
                              LD1
make                -0.2053433845
address             -0.0496520077
all                  0.1618979041
num3d                0.0491205095
our                  0.3470862316
over                 0.4898352934
remove               0.8776953914
internet             0.3874021379
order                0.2987224576
mail                 0.0621045827
receive              0.2343512301
will                -0.1148308781
people               0.0490659059
....
charHash             0.1141464080
capitalAve           0.0009590191
capitalLong          0.0002751450
capitalTotal         0.0003291749
```

Compare prediction on test set.

```
library(MASS)
lda_res <- lda(x=X[train,],grouping=Y[train])

proba_lda <- predict(lda_res,newdata=X[test,])$posterior[,2]
predicted_spam_lda <- as.numeric(proba_lda > 0.95)

> table(predicted_spam_test, Y[test])
predicted_spam_test    0    1
                  0 1346  351
                  1   28  541


> table(predicted_spam_lda, Y[test])
predicted_spam_lda    0    1
                 0 1364  533
                 1   10  359
```

It seems as if LDA beats Linear Regression here, but would need to adjust cutpoint to get proper comparison. Use ROC curves.

ROC curves

We can change the cutpoint c

```
predicted_spam_lda <- as.numeric(proba_lda > c)
```

to get different tradeoffs between

- Sensitivity: probability of predicting spam given true state is spam
- Specificity: probability of predicting non-spam given true state is non-spam

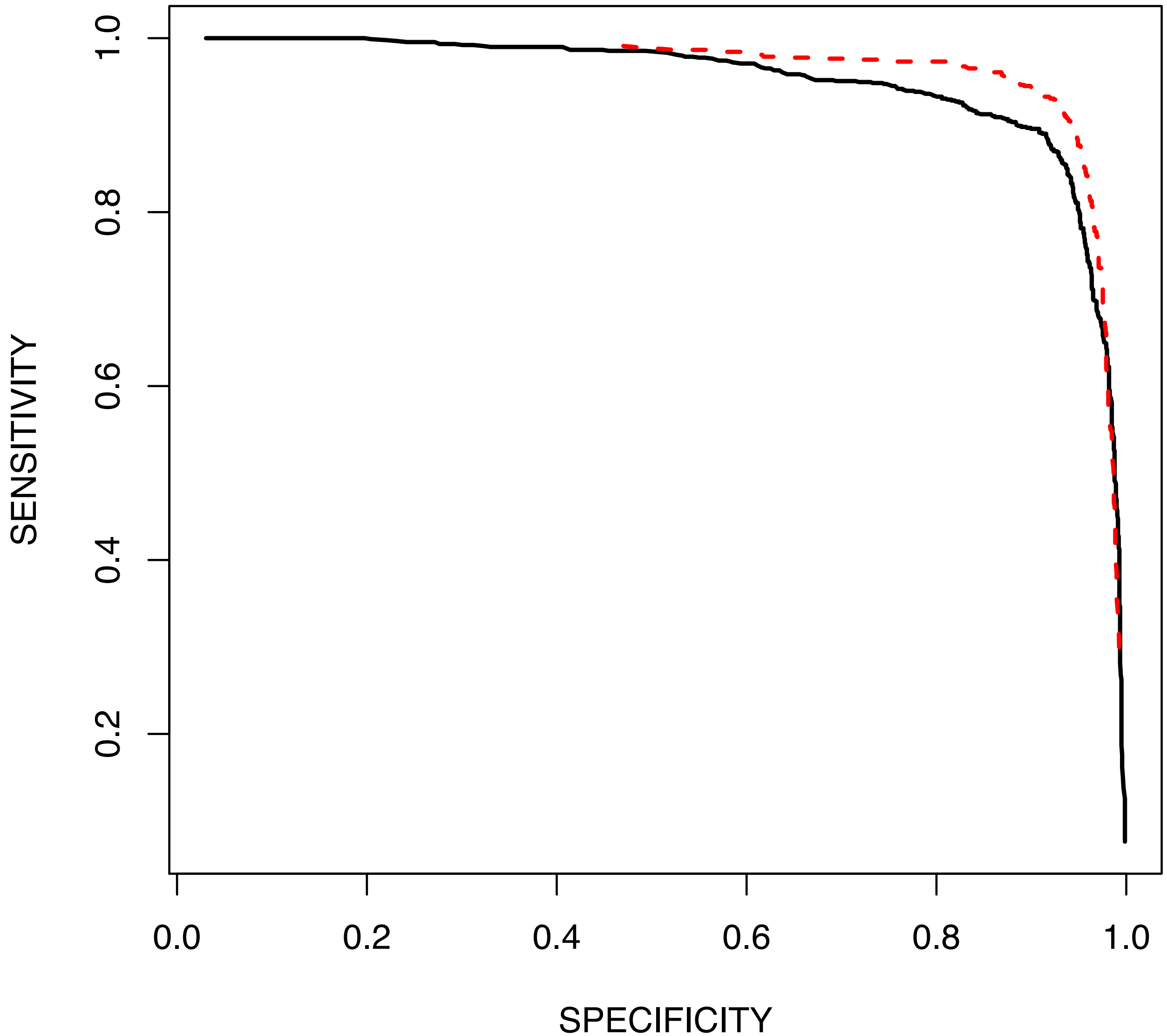
```
      TRUE STATE 0    1                            0       1
  PREDICTION   0 1364   533   normalize      0  0.9972  0.5975
               1   10   359    ---->         1  0.0072  0.4024
     TOTAL       1374   892                            1       1
```

ROC curve is sensitivity versus specificity

```
cvec <- seq(0.001,0.999,length=1000)
specif <- numeric(length(cvec))
sensit <- numeric(length(cvec))

for (cc in 1:length(cvec)){
  sensit[cc] <- sum( proba_lda> cvec[cc] & Y[test]==1)/sum(Y[test]==1)
  specif[cc] <- sum( proba_lda<=cvec[cc] & Y[test]==0)/sum(Y[test]==0)
}
plot(specif,sensit,
            xlab="SPECIFICITY",ylab="SENSITIVITY",type="l",lwd=2)
```

ROC curve for LDA and Logistic Regression classification of spam dataset.
LDA = unbroken black line; LR = broken red line.



Obvious now that LR is better for this dataset than LDA, contrary to the first impression.