Introduction to Machine Learning - Lab 1

Gustav Sternelöv

Thursday, October 22, 2015

## Assignment 1

### 1.1 - 1.2

To get started with the analysis of the e-mail data file the data first is divided into a train and a test data set. Each data set conatins 50 percent each of the original data set.

In the next step the distance matrix is computed and used to implement a k-nearest-neighbor function, called *knearest*. The distance matrix is given by 1 - c(X,Y) where c(X,Y) is defined as:

$$ X^T Y \over \sqrt{\sum\_{i=1}X\_i^2}\sqrt{\sum\_{i=1}Y\_i^2} $$

The code that creates the knearest function can be seen in the appendix "R-Code".

### 1.3

The implemented k-nearest-neighbor function is then used to classify the e-mails in both the training and test data. The classification principle that decides how to classify each e-mail is stated as following:  
 *if p* (Y = 1|X) > 0.5, *otherwise*   
With K, the number of neighbors, set to 5 the confusion matrix generated from classifying the training data looks like this:

##   
## 0 1  
## 0 1324 87  
## 1 80 809

The misclassification rate for the training data then is 0.0726087. It can be seen that it is approximately the same amount of non-spam and spam e-mails that has been misclassificated, but that a higher rate of the spam e-mails has been misclassificated.  
For the classification of the e-mails in the test data set this is the obtained confusion matrix:

##   
## 0 1  
## 0 1263 114  
## 1 108 816

For the test data the misclassification rate is 0.0964798, and the same pattern as for the training data regarding which e-mails that has been misclassificated can be seen for the test data.  
Unsurprisingly the misclassification rate is higher for the test data than the training data. Since the training data is used to make the classification of the e-mails it should work better for the training data, although the low rate for the test data implies that the model is quite good for the whole data set.

### 1.4

With K set to 1 instead of 5 the classification of the training data improves, and the misclassification rate is now 0.0143478. A summary over the classifications can be seen in the confusion matrix below.

##   
## 0 1  
## 0 1407 4  
## 1 29 860

For the test data set the opposite result is obtained. The misclassification rate when K is set to 1 is 0.1038679 which is a little bit worse than before, but still the model seem to work pretty good.

##   
## 0 1  
## 0 1262 115  
## 1 124 800

### 1.5

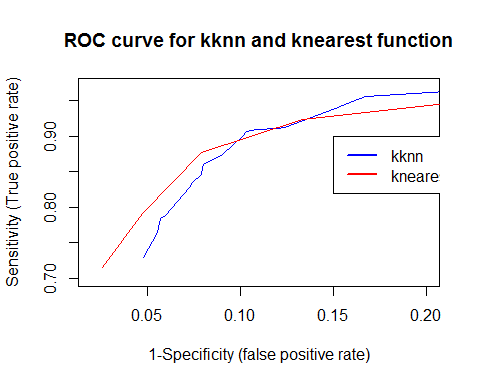
When the function *kknn* from the package **kknn** is used to classify the e-mails with the k-nearest-neighbor algorithm, the follwing results are obtained. K is set to 5 and for the test data the confusion matrix looks like this:

##   
## 0 1  
## 0 1262 115  
## 1 125 799

This gives a misclassification rate of 0.1043025, which is the highest misclassification rate received so far for the test data set. It therefore seems like the *knearest* function performs slightly better than the *kknn* function.

### 1.6

The sensitivity and specificity is computed for both the *knearest* and the *kknn* function. For each classification for the respective methods the sensitivity and specificity is computed according to the following formulas.  
Sensitivity = $True Positive\over All Positive$  
Specificity = $True Negative\over All Negative$

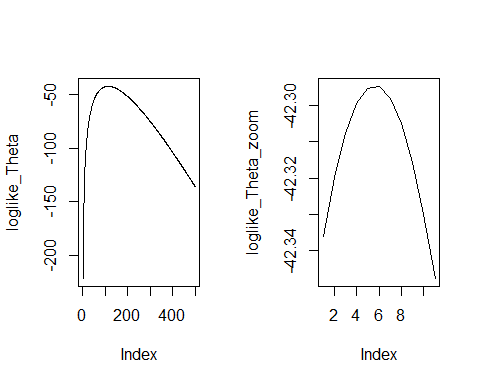
The ROC curve for both models is then fitted and plotted in the graph below. 

Some conclusions that can be drawn out of the ROC curves is that both models seem to be pretty good classifiers of spam and non-spam e-mails. There are no major differences between the models, the respective lines follow each other rather well. When the decision value for the classification principle is low the *knearest* function performs better and for high values the *kknn* function performs a little bit better. Looking more closely to the respective lines it can be seen that the red one, *knearest*, only change direction when the decision value reaches 0.2, 0.4, 0.6 and so on. That is related to how the function works when it classifies the e-mails, and apparently there is some difference in how the functions use the information about the five nearest neighbors.

## Assignment 2

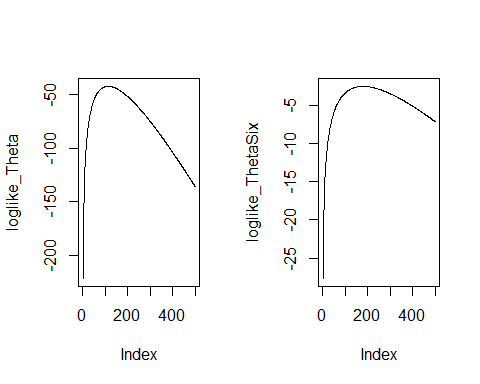
In the second assignment a specific type of machine and information about its lifetime is analysed. The main purpose of the analysis is to get more information about the underlying process for the lifetimes of the machines.

### 2.2

The x-values in the data set follows the exponential distribution. To investigate the dependence of the log-likelihood on , the log-likelihood is computed for a range of values for . A plot that shows this dependence can be seen below to the left where the values for goes from 0.01 to 5 by steps of 0.01.  
To find the maximum likelihhod value for the plot on the right side is used. It is zoomed in and only covers the range of going from 1.08 to 1.18 by 0.01. By the look of this plot the maximum likelohood value of seem to be 1.13. 

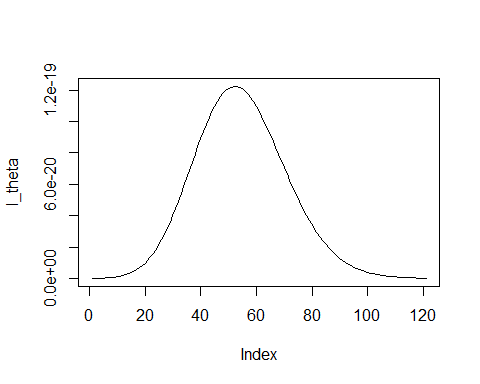
### 2.3

When comparing the log-likelihood curves it is evident that the estimation of the value is more reliable when more observations are available. The curve recieved when only six observations are used is more smooth which makes it harder to see what the optimal maximum likelihood value of is. A rough interpretation of the the plot gives that the maximum likelihood value of lies somewhere around 1.6-1.9. By looking at the exakt value for the log likelihood it is given that the optimal value of is 1.79.

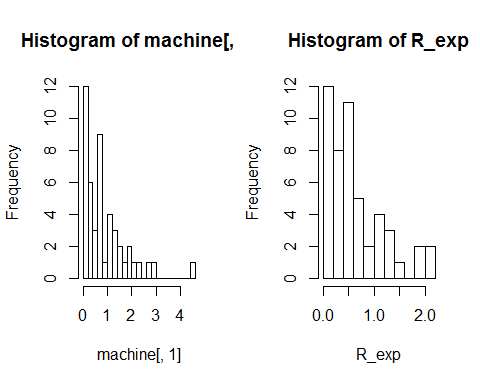


### 2.4

The measure computed by this function is the posterior distribution (?).  
The plot showing the dependence of l() on can be seen below, and by looking more closely to the values of l() it is concluded that the optimal is 1.11. That is near the result obtained for in 2.2, 1.13, and quite far away from the result in 2.3, 1.79.  
(?) The prior has its value of set to 0.5 and the "true" value of for the data is known to be 1.13. When the prior is combined with the observed data the optimal value value for ends up being almost equal to the true value.



### 2.5

In step 5 50 new observations are generated from the exponential distribution with the set to the optimal value found in step 2, 1.13. A comparsion between the histograms for the original data and the new data can be seen below. 

In both histograms a somewhat decaying pattern can be seen, which is typical for values following a exponential distribution. In the original data the frequency decays a bit slower, but in general the data sets seem to be rather similar distributed. Since both the original data and the simulated data has the same "true" value for the parameter this also was the expected result.