Spam Filtering: An Analysis of the SpamBase Data Set

Girish Sastry

April 4, 2011

1 Introduction

Spam filtering is an important classification problem in today's digital age. It is becoming ever more paramout to find ways to distinguish real, valuable information from worthless information, and as the firehose of information increases, this problem becomes even more relevant. With "Web 2.0" companies such as Twitter and Facebook dealing with petabytes of user generated data, distinguishing the wheat from the chaff translates directly to monetary value.

2 SpamBase Data

The SpamBase dataset is a publicly available data set from the UCI machine learning repository. It was originally curated by Hewlett Packard Labs, and contains spam data from two months in the summer of 1999. The problem is a binary classification problem (spam or not-spam). The data contains fifty-five features of continuous word and symbol statistics.

3 Plan for Data Analysis

Generally in classification problems we wish to minimize the misclassification error rate. In the development of a spam filter specifically, there is a class of misclassification errors that is egregious: false positives. Thus, we wish to minimize false positives. According to the data set's documentation, a previous analysis that enforced a constraint of zero false positives had a 25% misclassification error, which is pretty bad. In this project, we seek a good balance between misclassification error and number of false positives.

There are several classification methods from lecture that we wish to carry out on the data. These include: k-NN, LDA, QDA, reduced-rank LDA, and logistic regression. A couple of these methods may not make sense given that the dimensionality of the data set is not super high.

In addition to the methods discussed in class, we plan to use some simpler methods that may give better results: perceptron classification and naive bayes.

4 Tools

The methods discussed in class (k-NN, LDA, QDA, logistic regression) will be implemented in R, while perceptron classification and naive bayes may be implemented in Python.