Spam Detection - CS460 Final Project Report

Executive Summary

Spam has been a problem since the beginning of e-mail. Spam is defined as irrelevant messages sent out to a large number of users. They can range from minor annoyances to malicious attacks on users. Our project demonstrates two different supervised learning techniques to identify spam with: The Naïve Bayes approach and the SVM approach. For the purpose of this project we have only defined two class labels: Spam (as described above) and Ham (Messages that are not spam but may or may not pass as spam).

Supervised Learning

Our project uses training data to create a classifier that is then used on test data. Both of the algorithms below demonstrate this technique. The training set can be set to a sufficient partition of the total size of the dataset. It is important to not have an extremely large training set or there is a risk for overfitting the data. Overfitting describes the influence of random data or noise in the classifier. This reduces the accuracy of the classifier by blurring the underlying relationship between regular features with random features.

Naïve Bayes Classification

Naïve Bayes classification is the simplest way to classify spam based on the message body. As the name indicates, the algorithm uses the Bayesian Theorem to predict the nature of e-mail. To start training the classifier, we begin with an empty dictionary and the two labels. Each message body has a set of words that we input into the dictionary while increasing their total count and spam / ham counts depending on the training set label. After we go through the dataset, we end up with a classifier that contains information about the probability of a message being spam or ham given a set of words. The test data is then run on this classifier to predict its labels. Our classifier achieved a detection rate of 76% with this simple algorithm and using only 10% of the total dataset as our training set.

Support Vector Machine Benchmark

The Support Vector Machine (SVM) classification uses the concept of feature extraction to figure out if a message is spam. For our project we used a Bag of Words approach to extract features from the message body.

Features can be envisioned as a vector with *n* elements. Every e-mail has a vector associated with it that encodes its features. In the case of our project, each e-mail vector contains the word count of individual words in the message body. This feature set could also be expanded to include the presence of HTML markup, Message Recipients, Header information, Return-Path information, etc.

The algorithm designates an *n*-dimensional space to plot each of the vectors. We used the scikit library to determine the best-fit model for our training space vectors. Our features were the word count of distinct words in the message body. We achieved a detection rate of 83% with this algorithm using only 10% of the total dataset as our training set.