Spam Classification

Curtis Belmonte
curtislb@princeton.edu

Dorothy Chen
dschen@princeton.edu

Abstract

Spam, or unsolicited email, is becoming an increasingly large problem as email becomes more and more popular. In this assignment, we discuss various machine learning methods and use them to create spam classifiers. We then analyze the results and effectiveness of these methods.

1 Introduction and background

Email is something used daily by a large number of people. Advertisers and other people trying to sell things have taken advantage of this fact by sending out unsolicited emails, which is referred to as spam. In response to this, spam filters were created to identify these emails and to keep them from clogging up inboxes.

2 Description of data and data processing

The training data set consists of 22,500 spam emails and 22,500 non-spam emails from the trec07p data set. We used the provided script to define a vocabulary create a bag-of-words representation for each email. The resulting vocabulary contained 9579 words. The classifiers are built using these bag-of-words representations as features for the training data. The testing data set consists of 2,500 spam emails and 2,500 non-spam emails from the same corpus, and they are processed in a similar manner to also create bag-of-words representations.

3 Methods

We used the methods implemented in the scikit-learn package. [1]

3.1 Naive Bayes, using multinomial implementation

As a baseline, we trained a Naive Bayes multinomial model on all of the training data using the default parameters provided by scikit-learn. The default parameters included Laplace smoothing, learning class prior probabilities, and no previously provided class priors.

3.2 Decision tree

We also used the default parameters, which meant that splits were determined using Gini impurity, the best split at each node was chosen, and the depth of the tree was not constrained. However, nodes were only split if it contained at least two samples; a leaf could only exist if it had at least one sample.

Because of the large number of features and because the depth of the tree was not constrained, this method was significantly slower than the other two.

3.3 Random forests with 10 trees

We once again used the default parameters. This includes building 10 random trees using Gini impurity.

This method was much faster than the decision tree.

4 Results

4.1 Multinomial naive Bayes

Using this classifier, we predicted labels for the testing data and got an accuracy rate of 98.74%. The ROC curve can be seen in figure 1(a) and indicates a high true positive and low false positive rate.

4.2 Decision tree

Training this classifier on all available training data and testing on the provided testing set resulted in an accuracy rate of of 99.5%. The ROC curve can be seen in figure 1(b). Like the ROC curve for naive Bayes, it indicates that this method has a high true positive and low false positive rate. This method's higher accuracy (relative to naive Bayes) is shown by the fact that the ROC curve is barely visible because it's almost entirely on top of the axis.

4.3 Random forests

Using this method, we got 99.58% accuracy. The ROC curve can be seen in figure 1(c).

In an effort to view the data more clearly, we plotted the data on log scales. Specifically, we plotted the x-axis on a log10 scale in order to examine lower false positive rates. However, this did not significantly change the quality of the graph so we decided not to include it in this report.

5 Analysis of results

6 Conclusion and possible extensions

We feel that using a different model to represent the emails' content would be an interesting and potentially useful extension. While the bag-of-words model was certainly effective, many features were lost; in particular, we feel that things such as word order, capitalization, and punctuation are especially important in language and that their exclusion may have negatively impacted performance.

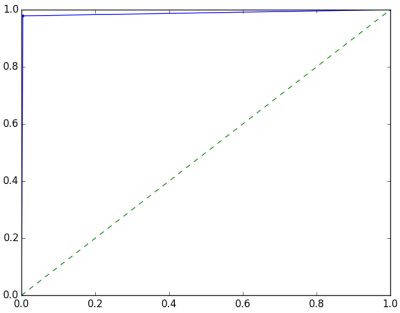
To elaborate on the above points, word order is important because it encodes meaning and context of sentences–single words on their own may be ambiguous. Capitalization also matters, as emails in all capitals, for example, are probably more likely to be spam. Punctuation also matters, because emails with punctuation are more likely to have proper grammar, and correct grammar intuitively seems indicative of non-spam (or, at least non-computer generated) emails.

Another possible extension is using a different data set. While our classifier performed very well on this particular data set, that performance might not necessarily hold for use on other data sets. This is because the set itself might have some trait that's highly indicative of spam versus not-spam that is not indicative of or present in the population of emails as a whole. It would be helpful to use a classifier trained on one set to test a different email corpus.

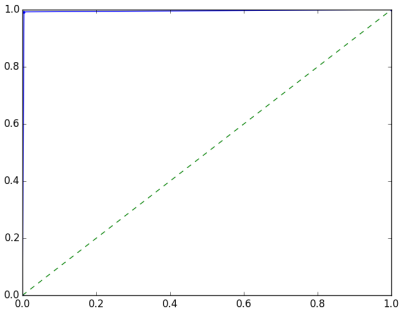
7 Acknowledgments

8 References

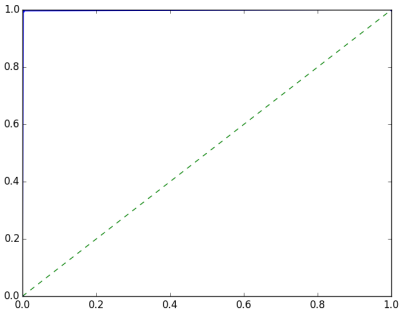
[1] http://scikit-learn.org/stable/



(a) Multinomial Naive Bayes



(b) Decision Tree



(c) Random Forest

Figure 1: ROC curves for various methods