Assignment 5- Spam Filter

Alam, Praveen ; Suresh, Muthukumar

May 12, 2015

Contents

1 Introduction

In this project, we use Naive Bayes to classify emails as Spam and Ham, where Spam means those that are sent by advertisers, tricksters and scam artists. This problem is perfectly tailored to the Naive Bayes classifiers. In order to identify a mail as spam or not, we use the corpus of words given to perform analysis of the words that most commonly occur in spam and those that occur most commonly in Ham. Most spam emails try to gain the attention of users by using CAPITAL letters and exclamation marks. By utilizing the knowledge from the huge corpus of 9000 spam emails from 2005 TREC Public Spam Corpus. We use this as our training data, in the next section we explain how we use this data to make decisions regarding whether a mail is classified as spam and ham. In section 3, we talk about our results and the accuracy of our training methodology.

2 Algorithm Design

To perform our spam analysis, we first divide out corpus into a list of tokens. For each of these tokens, we count the number of times it occurs in mails that are classified as spam and the number of times it occurs in mails that are considered as ham.

```
There is an inherent advantage of not crowding analysis phase with unnecessary heuristics like
exclamation points, etc, because such details are inherently captured in the analysis
```

. If a certain spam sensitive word occurs regularly, then it is more likely to be marked as spam automatically by the algorithm and by avoiding interference, we make sure that we do not introduce any unwanted bias.

Once we have developed our classifier based on the training data, we use it to classify our mails according to this:

```
probSpam = learnedData['spamCount']/learnedData['mailCount']
probHam = learnedData['hamCount']/learnedData['mailCount']
```

LearnedData is our global database and contains all the information required to make a decision whether a mail should be marked as spam or ham. learnedData['spamCount'] gives the number of times a mail has been clasified as spam in out training data and viseversa for ham count. Now, for each word in the mail that has to be classified, there are two cases:

1. the word is present in our training data
2. the word is absent in our traiing data

If the word is absent in our training data, we consider the word to have equal probability of being a spam/ham mail. From our analysis, not introducing artificial bias seems to give the best results.
On the other hand, if the word is present in the training data, then we associate with the word a pobability that it belongs to spam or ham.

```
wordSpamProb = (learnedData[word]['spam']+p*m)/(learnedData[word]['ham']+learnedData[word]['spam']+m)
wordHamProb = (learnedData[word]['ham']+p*m)/(learnedData[word]['ham']+learnedData[word]['spam']+m)
```

As we can see from the snippet above, for each word, we have from the analysis performed on the training data, the counts of the number of times the word has occured as spam and as ham. we use that count divided by the total number of times the word has occured as the probability of spam and ham. Also, note that we add a small value pm to the probability, this ensures that the probability does not drop down to 0.

Finally, we multiply the probabilty with the probability of spam and probability of ham calculated previously to get the true naive bayes probability of the mail belonging to spam and ham.

```
naiveBayesSpamProb = emailSpamProb*probSpam
naiveBayesHamProb = emailHamProb*probHam
```

By doing this, we ensure that we consider the probability that a mail is spam or not is also considered in our decision.

Based on the value, we simply classify the mail as spam if the naiveBayesSpamProb is higher or viseversa.

3 Analysis

When we run our analysis on the data test data,we get an accuracy of around 90%.

```
correct 896 incorrect 104
correct 89.60000000000001 incorrect 10.4
```

Out of the 1000 mails, it correctly classifies 896 mails and misclassifies 104 emails. In general spam emails seems to have more number of words such as now and company. We also noticed that there are more numbers present in spam emails than in ham emails.

```
'now': {'spam': 909, 'ham': 527}
'marketing': {'spam': 228, 'ham': 193}
'quoted': {'spam': 1356, 'ham': 378}
'0800': {'spam': 5007, 'ham': 525}
```

This is just a small sample but as you can see most of the words in spam are related to urgency and time and money. A mail that has these features is predominantly classified as spam whereas those mails that have names and words such as technology are mostly clasified as Ham.

4 conclusion

As we can see, we can use Naive bayes classifier for the task of spam detection, many of the spam emails share common features that when extracted provide us with good indicators for if a mail is spam or not.