Spam Classification Documentation

What is SPAM?
"Unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient."

Objective:
1. Develop an algorithm apart from Bayesian probabilities, i.e. through Frequent item set Mining, Support Vector Machines (SVM).
2. Compare the accuracy of the algorithms (Bayesian, frequent item set mining, Support Vector Machines) on a corpus with filter having no prior options. This helps in finding the best algorithm.

Problems in spam detection:
1. The perception of spam differs from one person to another.
2. It's difficult to come up with a traditional machine learning algorithm to detect spam.
3. Traditional spam filters such as spambayes, spamassasin, bogofilter use Naïve’ bayes for machine learning to detect spam.
4. However we believe efficient machine learning with personalization leads to better spam filter.

Problems with Probability Models (Naïve’ Bayes):
Spam filters using Bayes theorem for classification will actually use Naive Bayes as it assumes independence among all the words.

We observe two problems with probability models
i. Bayes Poisoning
ii. Learning Rate

Bayes Poisoning: Bayes Poisoning is a technique used by spammers to attempt to degrade the effectiveness of spam filters that rely on bayesian spam filtering. In other words, Spammers are intelligent enough to know that the filters use Bayes theorem as an integral part and they along with regular Spam words simply use legitimate Ham words to decrease the Spam probability of the mail and thereby escape the spam filter. This process is known as BAYES POISONING.

Learning Rate:
The learning rate of spam classifier using Naïve Bayes as machine learning algorithm is low as it depends on probability model to learn.

Our Approach:
We followed two approaches for efficient identification of Spam
1. Frequent Item set Mining approach (To Nullify the effect of Bayes Poisoning)
2. Support Vector Machines (SVM’s are known to work well for 2 class problems and as Spam problem is a 2 class problem we thought of using SVM’s)
“Frequent Item Word” Approach:
   a. How this approach Nullifies Bayes Poisoning
      Explanation with an Example for Frequent word combination:

Suppose 'Viagra' and 'Money' are frequently occurring Spam words and both the words are present at different parts of the mail. As the Spam probability of the mail is calculated assuming independence between the words, there is a possibility that the mail would escape the filter if some Ham words are used deliberately by the Spammer.

However there is a little chance for escaping the filter, if we generate frequently occurring combinations of Spam words (Though they are present at different positions in the mail) and use them in the scoring function as such a combination would generate more meaning.

Work Done by Us:

1. We generated frequent word combinations of Hamwords and Spam words and updated their probabilities using a modified Apriori algorithm.
2. This generation of frequent word combinations is integrated with the Spam Bayes Open source Spam filter. This part is done during Training.
3. We tried 2 or 3 naive approaches for using these results in the scoring function and the accuracy improved a little.
4. Though there is a little improvement in accuracy we gave up this approach due to its time complexity.

Example:

A new mail came for classification and it has n words. To generate a maximum of x-length combination we have to generate

\[ n \choose 1 + n \choose 2 + n \choose 3 + \ldots \ldots \ldots n \choose x \] combination of words and check if these word combinations are frequent with training data and use the frequent word combinations in the scoring function.

Points to Note:

1. There is a small increase in accuracy; however the algorithm is slower than normal Bayes theorem.
2. This accuracy might improve significantly if we have used the Frequent word combination in an optimised way in the scoring function. We have been limited to use
them effectively because it cannot be integrated with Spam Bayes as some Mathematical functions (Chi Square Probability and Central Limit theorem) are used on the top of Naive Bayes in Spam Bayes Filter.

Classification of Spam using Support Vector Machines (approach -2):

While implementing the previous method of Frequent Items data set method for the future pruning we explored the spam classification in the different way from the Spambayes. Many people advocated the using of the machine learning approaches for the spam classification. One of the recent approaches advocates is by D.Sculley et al [1] in SIGIR 2007. They proposed the algorithm for attacking online spam using SVM’s.

Not many people explored the spam classification using the SVM’s. We referred the work by Qiang Wang et al [2] titled SVM-Based Spam Filter with Active and Online Learning. Another work we referred was Batch and Online Spam filter comparison by Gordon et al [3]. We implemented the spam classification using svms. The results on the TREC 2005 and 2007 datasets are reported.

Support Vector Machines Theory:

Support vector machines (SVMs) are a set of related supervised learning methods used for classification. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers. Each data point will be represented by a p-dimensional vector (a list of p numbers). Each of these data points belongs to only one of two classes. The idea of SVM classification is find a linear separation boundary $W^T x + b = 0$ that correctly classifies training samples (and, as it was mentioned, we assume that such a boundary exists). We don’t search for any separating hyperplane, but for a very special maximal margin separating hyperplane, for which the distance to the closest training sample is maximal. Unlike perceptron which tries to find a possible linear separating boundary SVM try to find the optimal separating hyper plane. There are soft margin SVM’s where if the linear separating boundary does not exist. The SVM allows some level of misclassification in this soft margin but its more efficient than finding any of the complex boundary.

Our Approach:

The SVM needs the data to be in the numeric form to perform the mathematical calculation. So the TREC data is all converted to the numeric form by indexing (giving indices) to all possible vocabulary in the dataset. Replace the word with its index number corresponding to that word. All the mails are converted to the non-trivial number format and the we will use this as the dataset. Now we have spam numbers instead of spam words! The mail is converted
into the word stream using the spambayes and we use that wordstream to construct the vocabulary.

The Features are extracted from this numeric-mail dataset. We used the normal measure of the word frequency. Now all the mails are converted to the feature space. The feature space is the in the dimension of the vocabulary where each word represents one axis. This is similar to the vector space model. So each mail is represented as a point in the feature space. Now SVM is used for classification.

There is a online C++ library called SVMLight which is used to implement the svms. The complete svm based classification code is written in C++. We took help some online libraries to make implementation efficient and user friendly. The Dataset used are TREC 2005 dataset and second one is TREC 2007. The results for the few significant experiments we conducted.

Experiment 1

Training set size: 84482
Validation size: 92189
Number of Support Vectors: 4445
False Positives: 11
False Negatives: 696
Training Time : 80 sec
Validation Time : 1313.secs
False Positives %: 0.0119
False Negatives %: 0.754
Accuracy : 99.245

Experiment 2:

Training set size: 16K
Validation size : 16k+35K(new mails)
False Positives: 2
False Negatives: 336
Training Time : 30 sec
Validation Time : 308 secs
False Negatives %: 0.88
Accuracy : 99.12

Experiment 3:

Trained on 35K and tested on 35K + 92K
Validation set size: 127826
False positives: 66
False negatives: 1346
Training time: 30
Validation time: 1091.67
Accuracy: 98.94

Experiment 4 on Trec-07
Results (Support Vector Machine Classifier):
Training and testing set are same
Validation set size: 75419
False positives: 19
False negatives: 76
Training time: 69.97
Validation time: 621.12
Accuracy: 99.89
Svm's can also overfit the data like in the above case.

Experiment -5 on Trec – 07
Training on 50k mails and testing on 8888 new mails
Validation set size: 8888
False positives: 6
False negatives: 7
Training time: 68.49
Validation time: 72.04

For all the above experiments we have used Softmargin svm with a vocabulary of 90K
But the actual vocabulary size is greater than 8lakhs.

Important Points to Note:

- The results are good compared to that of normal Bayesian classification.
- This SVM's offer more generalization than any of the other classifiers.
- The Vocabulary uses is 10% compared to that of the actual even then the results are very good.
- Data is not linearly separable. So hardmargin svm failed to get good results. The accuracy percentage is somewhere around 50%.
• The most important thing is that the false positives are very less which is very important in spam classification.
• The implementation is very naïve and does not use any optimization techniques
• The results are efficient in both time of execution and accuracy.
• It is one of the potential direction to work for classification of the spam.
• The Spam classification can be done as learning the spam vocabulary on addition of the new mails.
• This has potential use online because of its fastness

References:
[1] “Relaxed Online SVMs for Spam Filtering” by D. Sculley and Gabriel M. Wachman

[2] “SVM-Based Spam Filter with Active and Online Learning” by Qiang Wang, Yi Guan and Xiaolong Wang

[3] “Batch and Online Spam Filter Comparison” by Gordon V. Cormack and Andrej Bratko