SPAM FILTERING – A COMPARATIVE STUDY OF THE PERFORMANCE OF DIFFERENT CLASSIFIERS FOR EFFECTIVE FILTERING

Dhananjay Tyagi[1]

[1]*Computer Science and Engineering Department, Guru Tech Bahadur Institute of Technology, New Delhi*

Abstract

Electronic mail is used daily by billions of people to interact and communicate around the world and is a critical application for many businesses. Over the last couple of decades unsolicited bulk email has become a headache for the email user. A staggering amount of spam is streaming into user's mailboxes daily. Spam is not only irritating for most email users but it also overtaxes the IT infrastructure of businesses and costs billions of dollars in wasted productivity. The need of effective spam filtering techniques increases. Machine learning algorithms can be used with current spam filtering schemes for increased efficiency. This paper presents a comparative study of the performance of different Machine Learning Algorithms which can be used to filter a mail as spam or ham.

***Keywords**- Spam, Machine Learning, AdaBoost, Naïve Bayes, K-NN, SVM*

--***--

1. INTRODUCTION

The internet has become an essential part of life and email has become an important tool for the exchange of information. Along with the progress of the Internet and e-mail, there has been a substantial rise of spam. Spam can be created from any part of the world where Internet is accessible. Although the anti-spam services and technologies exist, the number of spam messages continues to flourish. In order to tackle the growing problem, each organization must analyze the tools available to determine how best to counter spam in its environment. Users cannot avert the grave problem of attempting to handlemyriad of spams regularly. If no measures are taken against spam it will flood network systems, destroy employee productivity and appropriate bandwidth. Spam is one of the major threats posed to email users. In 2015, 69.6% of all email exchanged were spam. Links sent in spam emails may drive users to websites with malicious or phishing schemes, which can access and obstruct the user's computer system. These sites can also garner information sensitive to the user. Thus, an efficient spam filtering technology is important for the cyberspace and society.

Currently, different approaches to spam detection exists. These approaches consists of blacklisting, greylisting, scanning message headings, detecting bulk emails and content-based filtering:

- Blacklisting is a scheme which identifies IP addresses that send myriad of spams.

These IP addresses are noted in the DNS-Blackhole List and further email from these IP addresses are discarded.

- Another way to filter spam is by detecting emails that are sent in bulk. This scheme uses the number of recipients to discern if an email is spam or not.
- Another effective way to detect spam is to scan message headings.
- Greylisting is a scheme in which the email is not sent and an error message is sent back to the sender. Spammers will disregard this and not re-send the email, while legitimate people are more probably going to resend the email. However, this process is irritating and is not a solution.

Current spam filtering methods can be merged with content-based spam filtering methods to increase efficacy. Content-based schemes examine the content of the email to predict if the email is spam or ham. The objective of this paper is to analyze and compare different machine learning algorithms and determine their efficacy as content-based spam filters.

2. MACHINE LEARNING FOR CLASSIFICATION

Spam filtering, from the aspect of machine learning, is essentially a classification problem in which we aim to classify an email as spam orham which is dependent on its feature. For instance, (x,y) can be a data point where x is a d-dimensional vector containing the features and y is either 1 or 0, which indicates spam or ham. Systems with machine learning can be taught or trained to classify emails.

Training

While training, the system is provided with labeled data from a training data set. In this paper, the labeled data are a huge set of emails that are labeled with a solution spam or ham. In the training phase, the classifier (the thing that actually predicts further labels) learns from the training data by discovering connections between the label of emails and its features.

Testing

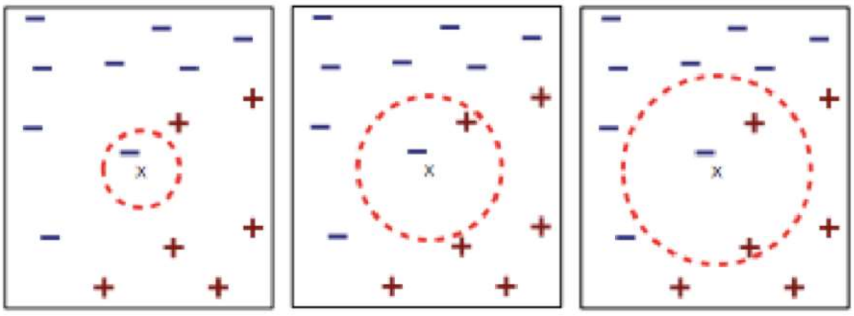
While testing, the system is given unlabeled data. The classifier determines whether an email is spam or ham based of the feature. The predicted classification is then compared to the true value of the label to compute the performance.

3. CLASSIFIERS

A machine learning system can be taught or trained to classify emails as spam or ham. For this purpose, learning system must use some measure to furnish its decision. The algorithms that are outlined below are distinct ways of deciding the label of the email.

3.1 k-Nearest Neighbors

The fundamental idea for the *k*-Nearest Neighbors (*k*NN) algorithm is that data points that are similar will have same labels. *k*NN sees the *k* closest (and therefore, most similar) training data points near the testing data point. The algorithm then processes the labels of those training data to decide the label of the testing data point.



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Fig-1: The *k*-Nearest Neighbors algorithm with (a) $k = 1$, (b) $k = 2$, and (c) $k = 3$.

This paper combines the labels by employing a simple majority vote, that is, the label of the neighbor is considered equally likely for the new data point.

3.2 Naïve Bayesian

The roots of the Naïve Bayesian classifier lie in the Bayes Theorem:

$$P(H/e) = \frac{P(e/H)\,P(H)}{P(e)}$$

Bayes Theorem basically describes how much we should modify the probability so that our hypothesis (H) transpires, given some novel evidence (e).

This paper determines the probability that an email is spam, given the evidence of the email's feature values F_1,F_2….F_n. These features are just a boolean value (0 or 1) dependent on whether the feature is present in the email or not. ThenP(Spam| features) to P(Ham| features) are determined and then decided which is more likely. The "C" in the below equation is the class, that is, spam or ham. The probabilities are calculated using the following equation:

$$P(C|F_1, \ldots\ldots, F_n) = \frac{p(C)p(F_1,\ldots\ldots,F_n|C)}{p(F_1,\ldots\ldots,F_n|C)}$$

3.3 AdaBoost

Boosting is a meta-algorithm employed in machine learning whose objective is to repeatedly build a group, that is, the weighted sum of the weak learners. High precision of weak learners is ensured as each is weighted and trained to generate an overall strong model. A ‘weak learner’ provides an accuracy somewhat better than random guesswork.
Each classifier from the weak learner family uses a threshold over the "within-document-frequency" of a specific word. It tries all frequencies appearing in the training set, and takes the one minimizing the current "weighted error" of misclassification. The weights are then updated to emphasize misclassified rows. This is done for a few tens of iterations. The final classifier is a linear combination of all those detected weak-learners.

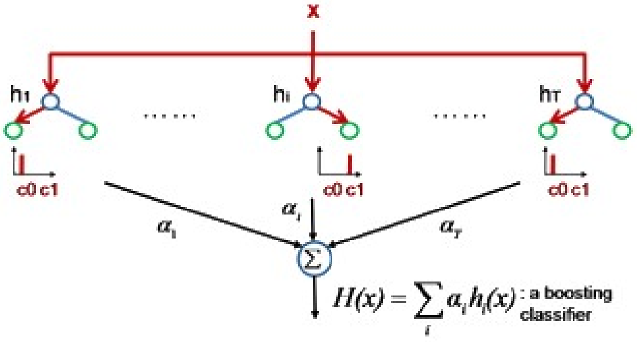


Fig-2: Boosting Framework

AdaBoost which is short for Adaptive Boosting, works by attaching a ‘weight’ to each point present in the training set which depends on how misjudged the output is relative to the earlier step. Then a new week learner and their weights are trained to minimize the weighted sum of errors against the output of the week learner.

3.4 Support Vector Machine

Support Vector Machines with associated learning algorithms analyse data for classification. The SVM algorithm for training constructs a model which allocates the new examples into one of the categories. In an SVM model examples are represented as points in n- dimensional space which are mapped so that the points of the different categories are separated by a gap that should be as broad as possible. Then the untrained examples are mapped in that space and a decision is made, that is, to which category does it belongs to depending upon the side of the plane they fall.

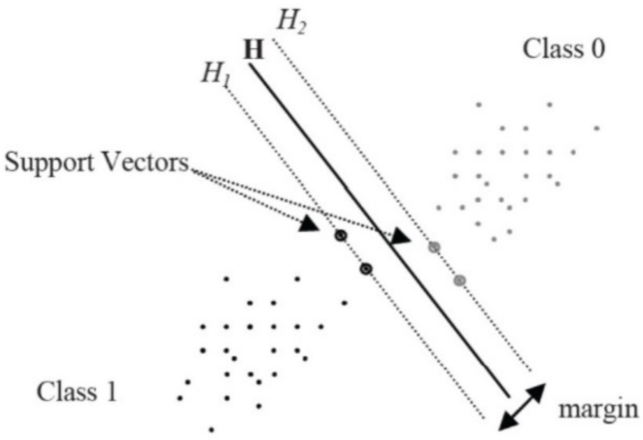


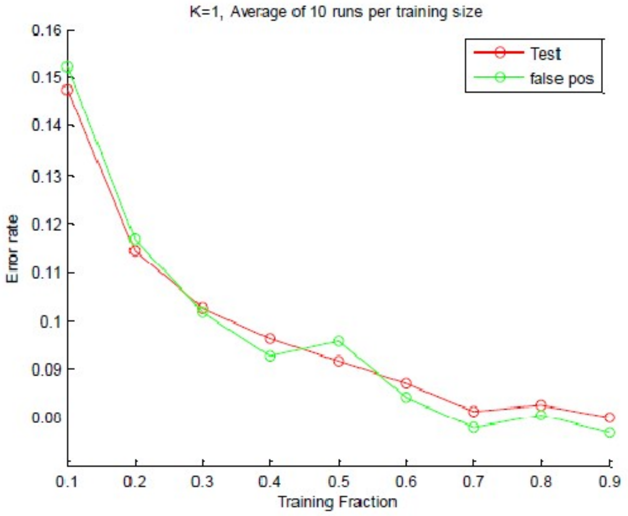
Fig-3: A hyperplane separating two classes

4. RESULTS

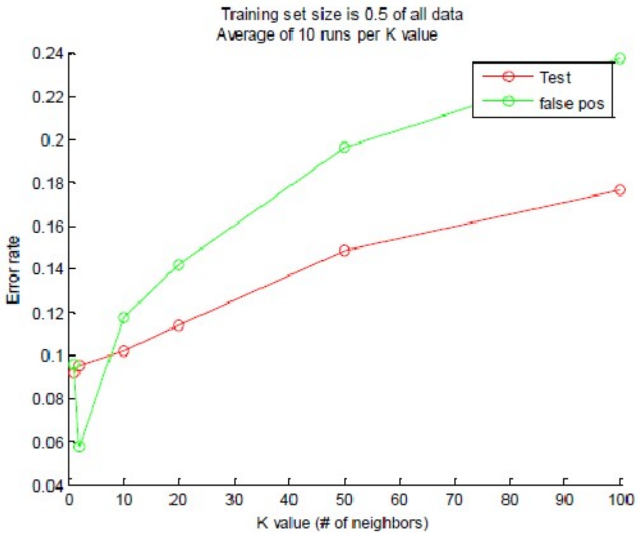
This paper measure the efficiency of the algorithms by two criteria-

- Error Rate
- FalsePositive Ratio.

4.1 k-Nearest Neighbors

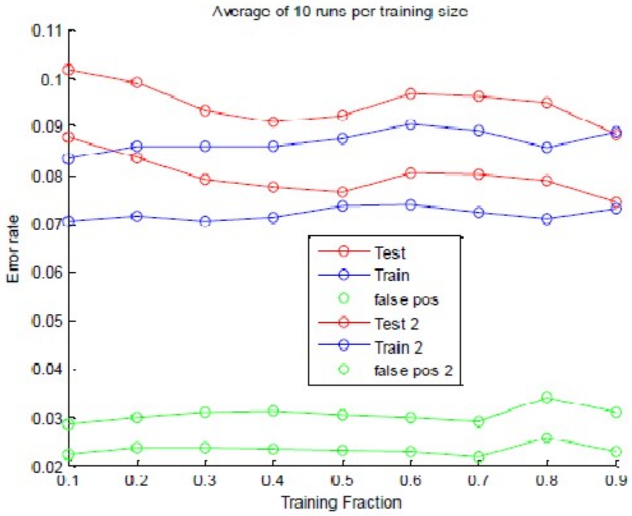


Graph-1: Error rate vs Training fraction for K=1



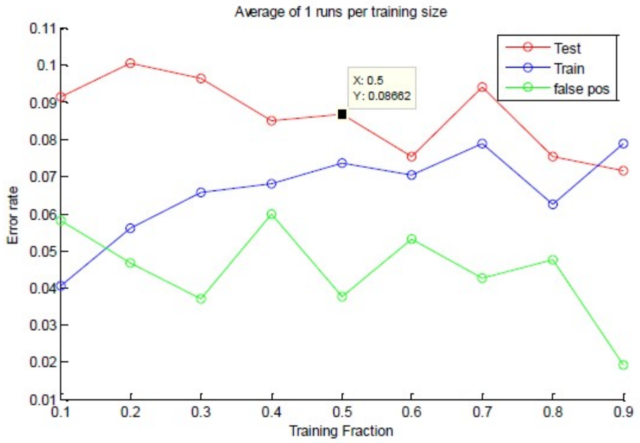
Graph-2: Error rate vs the values of K

4.2 Naïve Bayes

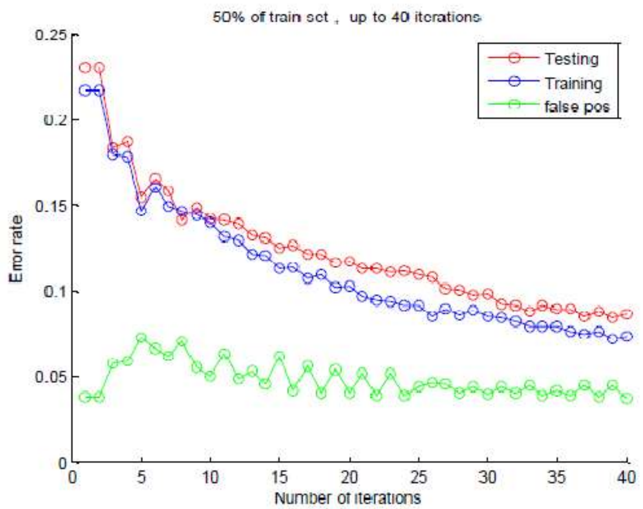


Graph-3: Error rate vs Training fraction

4.3 AdaBoost

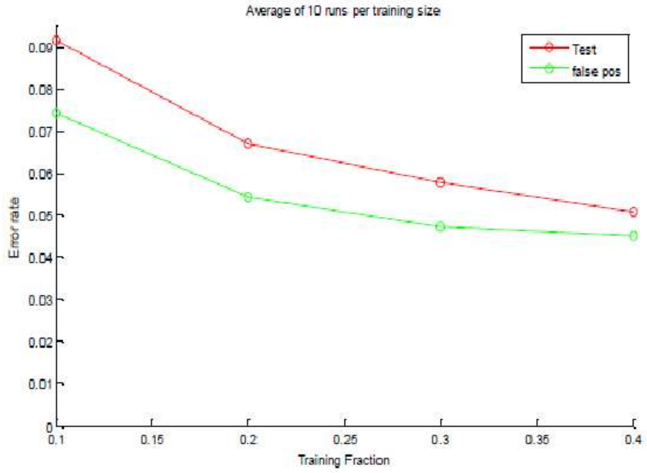


Graph-4: Error rate vs training fraction for average of 1 run



Graph-5: Error rate vs Number of iterations

4.4 Support Vector Machine

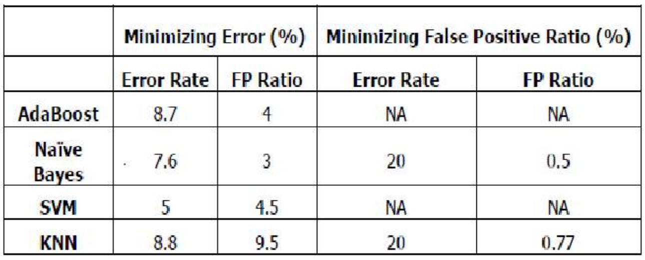


Graph-6: Error rate vs Training fraction

5. CONCLUSION

Following are the tabulated results of the algorithms under their optimal conditions. All results are trained with 0.5 fraction of the data set excluding SVM for which the training fraction is 0.4 (the maximum that could be tested in the scope of this paper).

Table-1: Comparative results of the different algorithms.



	Minimizing Error (%)		Minimizing False Positive Ratio (%)	
	Error Rate	FP Ratio	Error Rate	FP Ratio
AdaBoost	8.7	4	NA	NA
Naïve Bayes	7.6	3	20	0.5
SVM	5	4.5	NA	NA
KNN	8.8	9.5	20	0.77

Naïve Bayes and SVM are the two best algorithms as per the results. Naïve Bayes owing to its low false positive ratio and low run time is the widely employed in real world. Also Naïve Bayes is easier than other algorithms to implement. It's hard to reduce the false-positive ratio of SVM and AdaBoost because they do not adjust the balance of the errors.

REFERENCES

[1] Ahmed Khorsi, "An Overview of Content-Based Spam Filtering Techniques", Informatics 31 (2007) 269-277 269

[2] David Mertz, "Comparing a Half-Dozen Approaches to Eliminating Unwanted Email", August 2002

[3] Wikipedia, "Spam". http://en.wikipedia.org/wiki/Spam_(electronic)

[4] Symantec, "State of Spam and Phishing. A Monthly Re-port 2010," 2010. http://symantec.com/content/en/us/enterprise/other_resources/b-state_of_spam_and_phishing_report_09-2010.en-us.pdf.

[5] M. Sahami, "Learning Limited Dependence Bayesian Classifiers," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, The AAAI Press, Menlo Park, 1996, pp. 334-338.

[6] R. A. Fisher, "On Some Extensions of Bayesian Inference Proposed by Mr. Lindley," *Journal of the Royal Statisti-cal Society*: *Series B*, Vol. 22, No. 2, 1960, pp. 299-301.

[7] Kaspersky Security Bulletin 2015.https://securelist.com/analysis/kaspersky-security-bulletin/73038/kaspersky-security-bulletin-2015-overall-statistics-for-2015/

[8] R. Segal, J. Crawford, J. Kephart and B. Leib, "Spam-Guru: An Enterprise Anti-Spam Filtering System," IBM Thomas J. Watson Research Center. http://www.research.ibm.com/people/r/rsegal/papers/spamguru-overview.pdf.

[9] Microsoft Antispam Technologies. http://www.microsoft.com/mscorp/safety/technologies/antispam/default.mspx.

[10] W. Lauren, "Spam Wars," *Communications of the ACM —Program Compaction*, Vol. 46, No. 8, 2003, p. 136.

[11] M. F. Saeddian and H. Beigy, "Spam Detection Using Dynamic Weighted Voting Based on Clustering," *Pro-ceedings of the* 2008 *Second International Symposium on Intelligent Information Technology Application*, Vol. 2, pp. 122-126. doi:10.1109/IITA.2008.140

[12] S. M. Lee, D. S. Kim and J. S. Park, "Spam Detection Using Feature Selection and Parameters Optimization," *IEEE International Conference on Intelligent and Soft-ware Intensive Systems*, Krakow, 15-18 February 2010, pp. 883-888. doi:10.1109/CISIS.2010.116

[13] G. Pawel and M. Jacek, "Fighting the Spam Wars: A Re-Mailer Approach with Restrictive Aliasing," *ACM Transactions on Internet Technology* (*TOIT*), Vol. 4, No. 1, 2004, pp. 1-30.

BIOGRAPHY

The author has a bachelor's degree in Computer Science and Engineering from GGSIPU. His areas of interest are Machine Learning, Neural Networks and Data Mining.