July 2014

IMPROVING SPAM EMAIL FILTERING EFFICIENCY USING BAYESIAN BACKWARD APPROACH PROJECT

M. SHESHIKALA

SREC Engineering College, Warangal, marthakala08@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijcsi>

Part of the [Computer Engineering Commons](#), [Information Security Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

SHESHIKALA, M. (2014) "IMPROVING SPAM EMAIL FILTERING EFFICIENCY USING BAYESIAN BACKWARD APPROACH PROJECT," *International Journal of Computer Science and Informatics*: Vol. 4 : Iss. 1 , Article 1.

DOI: 10.47893/IJCSI.2014.1164

Available at: <https://www.interscience.in/ijcsi/vol4/iss1/1>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer Science and Informatics by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

IMPROVING SPAM EMAIL FILTERING EFFICIENCY USING BAYESIAN BACKWARD APPROACH PROJECT

M.SHESHIKALA

Assistant Professor, SREC Engineering College, Warangal
Email: marthakala08@gmail.com,

Abstract- Unethical e-mail senders bear little or no cost for mass distribution of messages, yet normal e-mail users are forced to spend time and effort in reading undesirable messages from their mailboxes. Due to the rapid increase of electronic mail (or e-mail), several people and companies found it an easy way to distribute a massive amount of undesired messages to a tremendous number of users at a very low cost. These unwanted bulk messages or junk e-mails are called spam messages. Several machine learning approaches have been applied to this problem. In this paper, we explore a new approach based on Bayesian classification that can automatically classify e-mail messages as spam or legitimate. We study its performance for various datasets.

Keywords : Spam, legitimate, Hash, Words

1. INTRODUCTION

E-mail spam has become a rigorous problem that affects the usability of electronic mail as a communication means. Not only wasting users' time to scan and delete the massive amount of junk e-mails received; it also consumes network bandwidth and storage space, slows down e-mail servers, and provides a medium to distribute harmful and/or offensive content.

The amount of spam users see in their mailboxes is only a portion of total spam sent, since spammers' lists often contain a large percentage of invalid addresses and many spam filters simply delete or reject "obvious spam." A 2010 survey of US and European e-mail users showed that despite knowing the risks of opening spam e-mails, 46% of the respondents still opened them, putting their computers at risk.

Due to the rapid increase of electronic mail (or e-mail), several people and companies found it an easy way to distribute a massive amount of undesired messages to a tremendous number of users at a very low cost. These unwanted bulk messages or junk e-

mails are called spam messages. The majority of spam messages that has been reported recently are unsolicited commercials promoting services and products including sexual enhancers, cheap drugs and herbal supplements, health insurance, travel tickets, hotel reservations, and software products. They can also include offensive content such pornographic images and can be used as well for spreading rumors and other fraudulent advertisements such as make-money fast. E-mail spam has continued to increase at a very fast rate over the last couple of years. It has become a major threat for business users, network

administrators. Annual Security Report that spam activity has increased significantly in 2006 with levels that reach 86.2% of the e-mail traffic. The report has also indicated that largely due to the increased sophistication of robot networks, a.k.a. botnets, the spam volumes have increased by 70% over the last quarter of 2006 which in turn increased the overall e-mail traffic by a third. Based on projections of current analysis and trends, it was expected that by the end of 2007, spam will continue to rise, reaching a plateau at around 92% of e-mail traffic [2]. There is a prediction that by year 2015 spam will exceed 95% of all e-mail traffic [4]. Although these figures might not be accurate enough, what can be concluded is that spam volume is dramatically increasing over years.

Fig 1 : Spam as % of Email.

Spam can be very costly to e-mail recipients; it reduces their productivity by wasting their time and causing annoyance to deal with a large amount of spam.

According to Ferris Research, if an employee got five e-mails per day and consumes 30 seconds on each, then he/she will waste 15 hours a year on them. Multiplying this by the hourly rate of each employ in a company will give the cost of spam to this company [3]. In addition, spam consumes the network

bandwidth and storage space and can slow down e-mail servers. Spam software can also be used to distribute harmful content such as viruses, Trojan horses, worms and other malicious codes [5]. It can be a means for phishing attacks as well [4].

Fig 2 : Worst Junk Addressing

As a result, spam has become an area of growing concern attracting the attention of many security researchers and practitioners. In addition to regulations and legislations, various anti-spam technical solutions have been proposed and deployed to combat this problem. Front-end filtering was the most common and easier way to reject or quarantine spam messages as early as possible at the receiving server. However, most of the early anti-spam tools were static; for example using a blacklist of known spammers, a white list of good sources, or a fixed set of keywords to identify spam messages. Although these list-based methods can substantially reduce the risk provided that lists are updated periodically, they fail to scale and to adapt to spammers' tactics. They can be defeated easily by changing the sender's address each time, intentionally misspelling words, or forging the content to bypass spam filters.

The similarity of spam filters with text categorization problems and the success of machine learning techniques in solving these problems have intrigued several researchers to investigate their applicability in filtering spam. One subtle difference is that a false positive would be a more serious error than a false negative as a false positive would mean that an important e-mail was identified as spam and rejected. According to [6], a leading body in IT, inaccurate anti-spam solutions may be responsible for wasting more than five million working hours a year on checking that legitimate messages were not mistakenly quarantined. Recently, various machine-learning methods [7] have

been used to address spam filtering including support vector machines [8], memory-based learning [9, 10], rough set [11], neural networks [12], Bayesian classifiers [13–16], sparse binary polynomial hash [17], etc. Among these methods, the naïve Bayesian classifier has been widely applied as one of the most effective methods to counteract spam [18]. A recent overview and a taxonomy of current and potential solutions, both machine learning and non-machine learning, ranging from commercial implementations to ideas confined to current

research, is presented in [19].

In this paper, we propose an alternative automated spam filtering technique based on Bayesian approach. We study its effectiveness and compare the results with the naïve Bayesian approach. The rest of this paper is organized as follows. In Section 2, we briefly review related work on applying machine-learning techniques for classifying e-mail messages. Then, in Section 3, we describe the Bayesian approach to anti-spam filters. The performance evaluation of the proposed method using several datasets from a public corpus and the results are discussed in Section 4. Finally, Section 5 deals with future work, Section 6 concludes the paper.

2. RELATED WORK

Since the increase in the spam volume, spam filtering has attracted considerable attention over the past few years. Several solutions including commercial and open -source products have been proposed and deployed. We can classify spam-filtering methods into two broad categories: non-machine learning based and machine learning based [19]. Non-machine learning methods base their e-mail classification on a predefined list of known spammers and/or a list of keywords whereas machine learning takes the content of the message into its consideration and adapts its decision accordingly. Although machine-learning algorithms are vast and varying in their concepts, we will briefly discuss some of the most widely applied techniques in spam filtering.

Rule-Based Filtering: Usually, there is a certain pattern used in spam, rule-based filters examine messages for those patterns following specific rules in order to identify spam mails. The Ripper algorithm is a typical a rule -based classifier. By comparing it with C4.5 decision tree rules in [21], the author stated that it is more efficient in noisy datasets. This kind of filters often scales relatively poorly with the sample size.

Support Vector Machines (SVM): Drucker *et al.* [8] used a support vector machine for e-mail classification based on the content. Given the input as a binary feature vector, the idea behind SVM is to find a hyper plane that best separate data points into two classes with maximum margins between them. SVM is very well suited for text categorization [22]. As shown in [8], SVM has acceptable accuracy and speed, and needs significantly less training time as compared with other filtering algorithms including Ripper, Rocchio and boosting decision trees.

Memory-Based (Instance-Based) Approach : Sakkis *et al.* [9] proposed a memory-based approach to anti-spam filtering for mailing lists. In this

approach, each message in the training examples is converted into a vector representing the values of different attributes of the message. These vectors are stored in a memory structure and are used directly to classify e-mail messages. This method uses a variant of the simple k -nearest-neighbor (k -NN) method in which the classification is usually performed by assigning to each unseen instance the majority class of its k closest training instances [10]. Although various metrics can be used to calculate the distance between different instances, typically the Euclidean distance is used in k -NN. A thorough evaluation of memory-based filtering

was performed in [16], and it was found that it achieved better or comparable results to the naïve Bayesian approach. Another extensive empirical evaluation of memory-based learning in the context of anti-spam filtering is provided in [9]. It provides a thorough investigation on the effect of different parameters such as various attributes, distance weighting schemes, neighborhood size, the size of the attribute set, and the size of the training corpus. It was found that the performance of memory based is comparable to the naïve Bayesian approach and on average better particularly when the misclassification cost for non-spam messages is high.

Sparse Binary Polynomial Hash (SBPH): A generalization of Bayesian that can match mutating phrases as well as single words. By using SBPH, a large amount of features can be generated from an incoming text automatically, and then a weight is assigned to each feature according to the probability of being spam or not. As mentioned in [17], it can achieve accuracy up to 99.9% but it requires more computations.

Rough Set Theory: As suggested in [11], e-mail can be classified into three categories: spam, non-spam and suspicious. The results of their experiments showed that rough set based filters can reduce false positive classification.

Although a variety of machine learning techniques have been applied to spam filtering, Bayesian classification is one whose accuracy is above 97% and had low false positive rates.

3. Proposed Work

Bayesian Approach:

This is one of the most addressed machine learning techniques to identify spam. A spam filter that uses Bayesian approach was first proposed in [13]. Bayesian filters have been shown to generate very accurate results in finding spam messages [13 – 16, 18]. In [13], the authors claimed that it is probably the fastest anti-spam filter. In this technique, each message is described by a set of attributes (*e.g.* words or phrases). Probabilities are assigned to each attribute based on its number of occurrences in the training corpus. These probabilities are then used to

classify a message into the most probable category by applying Bayes' theorem. The filtering is done using Naïve Bayesian Spam Filter. Basically, an implementation of this is from

<http://www.paulgraham.com/spam.html>

The real advantage of the Bayesian approach, is that we know what we are measuring. Feature-recognizing filters like SpamAssassin assign a spam "score" to email. The Bayesian approach assigns an actual probability. The problem with a "score" is that no one knows what it means. The user doesn't know what it means, but worse still, neither does the developer of the filter. How many points should an email get for having the word "sex" in it? A probability can of course be mistaken, but there is little ambiguity about what it means, or how evidence should be combined to calculate it. Based on my corpus, "sex" indicates a .97 probability of the containing email being a spam, whereas "sexy" indicates .99 probability. And Bayes' Rule, equally unambiguous, says that an email containing both words would, in the (unlikely) absence of any other evidence, have a 99.97% chance of being a spam. Because it is measuring probabilities, the Bayesian approach considers all the evidence in the email, both good and bad. Words that occur disproportionately rarely in spam (like "though" or "tonight" or "apparently") contribute as much to decreasing the probability as bad words like "unsubscribe" and "opt-in" do to increasing it. So an otherwise innocent email that happens to include the word "sex" is not going to get tagged as spam.

Creating a word database for the filter

Fig 3: Word Database For the filter

3. STATISTICAL FILTERING.

We start with the collection of writings in the spam and one of nonspam mail. At the moment each one has about 4000 messages in it. We scan the entire text, including headers and embedded html and JavaScript, of each message in each list. We currently consider alphanumeric characters, dashes, apostrophes, and dollar signs to be part of tokens, and everything else to be a token separator. We ignore

the tokens that are all digits, html comments, not even considering them as token separators.

We count the number of times each token (ignoring case, currently) occurs in each corpus. At this stage we end up with two large hash tables, one for each corpus, mapping tokens to number of occurrences.

Next we create a third hash table, this time mapping each token to the probability that an email containing it is a spam, which we calculate as follows :

$$\begin{aligned} & \text{(let ((g (* 2 (or (gethash word good) 0)))} \\ & \quad \text{(b (or (gethash word bad) 0)))} \\ & \quad \text{(Unless (< (+ g b) 5)} \\ & \quad \quad \text{(max .01} \\ & \quad \quad \quad \text{(min .99 (float (/ (min 1 (/ b nbad))} \\ & \quad \quad \quad \quad \text{(+ (min 1 (/ g ngood))} \\ & \quad \quad \quad \quad \text{(min 1 (/ b nbad)))))))))) \end{aligned}$$

where word is the token whose probability we're calculating, good and bad are the hash tables created in the first step, and ngood and nbad are the number of nonspam and spam messages respectively. Once the probabilities are calculated this information is stored in text.

When a new mail arrives, the text of the body is read and probability for each word is assigned from the text that we have already calculated. Based on the probability of each word, we declare whether the mail is spam or legitimate. Once the mail is decided it is specified in different colors to the users.

4. RESULTS

Login Page.

- [1] L.F. Cranor, and B.A. LaMacchia, "Spam!" *Communications of the ACM*, vol. 41 no. 8, pp. 74-83, Aug. 1998.
- [2] MessageLabs, "MessageLabs Intelligence: 2006 Annual Security Report," 2006. <http://www.messagelabs.com/resources/mlireports>
- [3] GFI, "How to keep spam off your network," 2007. <http://www.gfi.com/whitepapers/>
- [4] B. Hoanca, "How good are our weapons in the spam wars?" *IEEE Technology and Society Magazine*, vol. 25, no. 1, pp. 22-30, Spring 2006.
- [5] M. Siponen and C. Stucke, "Effective anti-spam strategies in companies: an international study," In *Proc. of the 39th Annual Hawaii Int. Conf. on System Sciences*, 2006.
- [6] The British Computer Society. Anti-spam blamed for 5M lost hours, 2007. <http://www.bcs.org/>
- [7] X.-L. Wang, and I. Cloete, "Learning to classify e-mail: A survey," In *Proc. of the 4th Int. Conf. on Machine Learning and Cybernetics*, Guangzhou, Aug. 2005.
- [8] H. Drucker, D. Wu, and V.N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048 – 1054, Sept. 1999.
- [9] G. Sakkis, I. Androutsopoulos, G. Paliouras, "A memory-based approach to anti-spam filtering," *Information Retrieval*, vol. 6, pp. 49-73, 2003.
- [10] D.C. Trudgian, "Spam Classification Using Nearest Neighbour Techniques," In *Proc. of the Fifth Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL04)*, UK, 2004.
- [11] W. Zhao, and Z. Zhang, "An e-mail classification model based on rough set theory," In *Proc. of the Int. Conf. on Active Media Technology*, 2005.
- [12] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," In *Proc. of the IEEE/WIC Int. Conf. on Web Intelligence (WI'03)*, 2003.
- [13] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," In *Proc. of AAAI'98 Workshop on Learning for Text Categorization*, Madison, WI, July 1998.
- [14] J. Provost, "Naïve-Bayes vs. rule-learning in classification of e-mail," The University of Texas at Austin, Department of Computer Sciences Rep. AI-TR-99-284, 1999.
- [15] I. Androutsopoulos, J. Koutsias, V. Chandrinou, and D. Dpyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages," In *Proc. of the 23rd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2000.
- [16] I. Androutsopoulos, G. Paliouras, V. Karkaletsis and G. Sakkis, "Learning to filter spam e-mail: A comparison of a naïve Bayesian and a memory-based approach," In

International Journal of Computer Science and Informatics, ISSN (PRINT): 2231 –5292, Volume-4, Issue-1

- [17] GFi, "Why Bayesian filtering is the most effective anti-spam technology," 2007. <http://www.gfi.com/whitepapers/>
- [18] R. Hunt and J. Carpinter, "Current and new developments in spam filtering," In *Proc. of the 14th IEEE Int. Conf. on Networks (ICON'06)*, Sept. 2006.
- [19] P. Langley, I. Wayne and K. Thompson, "An analysis of Bayesian classifiers," In *Proc. of the 10th National Conf. on Artificial Intelligence*, San Jose, California, 1992.
- [20] W.W. Cohen, "Learning rules that classify e-mail," In *Proc. of AAAI'96 Spring Symposium on Machine Learning in Information Access*, Stanford, California, April 1996.
- [21] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," In *Proc. of the 10th European Conf. on Machine Learning (ECML-98)*, 1998.
- [22] M.M. Fuad, D. Deb, and M.S. Hossain, "A trainable fuzzy spam detection system," In *Proc. of the 7th Int. Conf. on Computer and Information Technology*, 2004.