Global Journal of researches in engineering
Numerical Methods
Volume 11 Issue 7 Version 1.0 December 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

Bayesian Spam Filtering Using Statistical Data Compression

By Gumpina V V Satya Prasad, Satya P Kumar Somayajula
Avanthi Institute of Engineering and Technology, Makavarapalem, Visakhapatnam

Abstract - The Spam e-mail has become a major problem for companies and private users. This paper associated with spam and some different approaches attempting to deal with it. The most appealing methods are those that are easy to maintain and prove to have a satisfactory performance. Statistical classifiers are such a group of methods as their ability to filter spam is based upon the previous knowledge gathered through collected and classified e-mails. A learning algorithm which uses the Naive Bayesian classifier has shown promising results in separating spam from legitimate mail.

GJRE Classification : *FOR Code: 050299*

Strictly as per the compliance and regulations of :

Bayesian Spam Filtering Using Statistical Data Compression

Gumpina V V Satya Prasad[α], Satya P Kumar Somayajula[Ω]

Abstract - The Spam e-mail has become a major problem for companies and private users. This paper associated with spam and some different approaches attempting to deal with it. The most appealing methods are those that are easy to maintain and prove to have a satisfactory performance. Statistical classifiers are such a group of methods as their ability to filter spam is based upon the previous knowledge gathered through collected and classified e-mails. A learning algorithm which uses the Naive Bayesian classifier has shown promising results in separating spam from legitimate mail.

I. Introduction

Spam has become a serious problem because in the short term it is usually economically beneficial to the sender. The low cost of e-mail as a communication medium virtually guaranties profits. Even if a very small percentage of people respond to the spam advertising message by buying the product, this can be worth the money and the time spent for sending bulk e-mails. Commercial spammers are often represented by people or companies that have no reputation to lose. Because of technological obstacles with e-mail infrastructure, it is difficult and time-consuming to trace the individual or the group responsible for sending spam. Spammers make it even more difficult by hiding or forging the origin of their messages. Even if they are traced, the decentralized architecture of the Internet with no central authority makes it hard to take legal actions against spammers. The statistical filtering (especially Bayesian filtering) has long been a popular anti-spam approach, but spam continues to be a serious problem to the Internet society. Recent spam attacks expose strong challenges to the statistical filters, which highlights the need for a new anti-spam approach. The economics of spam dictates that the spammer has to target several recipients with identical or similar e-mail messages. This makes collaborative spam filtering a natural defense paradigm, wherein a set of e-mail clients share their knowledge about recently receivedspame-mails, providing a highly effective defense against a substantial fraction of spam attacks. Also, knowledge sharing can significantly alleviate the burdens of frequent training stand-alone spam filters. However, any large-scale collaborative anti-spam approach is faced with a fundamental and important challenge, namely ensuring the privacy of the e-mails among un trusted e-mail entities. Different from the e-mail service providers such as Gmail or Yahoo mail, which utilizes spam or ham(non-spam) classifications from all its users to classify new messages, privacy is a major concern for cross-enterprise collaboration, especially in a large scale. The idea of collaboration implies that the participating users and e-mail servers have to share and exchange information about the e-mails (including the classification result). However, e-mails are generally considered as private communication between the senders and the recipients, and they often contain personal and confidential information. Therefore, users and organizations are not comfortable sharing information about their e-mails until and unless they are assured that no one else (human or machine) would become aware of the actual contents of their e-mails. This genuine concern for privacy has deterred users and organizations from participating in any large-scale collaborative spam filtering effort. To protect e-mail privacy, digest approach has been proposed in the collaborative anti-spam systems to both provide encryption for the e-mail messages and obtain useful information (fingerprint) from spam e-mail. Ideally, the digest calculation has to be a one-way function such that it should be computationally hard to generate the corresponding e-mail message. It should embody the textual features of the e-mail message such that if two e-mails have similar syntactic structure, then their fingerprints should also be similar.Afew distributed spam identification schemes, such as Distributed Checksum Clearinghouse (DCC) [2] and Vipul's Razor [3] have different ways to generate fingerprints. However, these systems are not sufficient to handle two security threats: 1) Privacy breach as discussed in detail in Section 2 and 2) Camouflage attacks, such as character replacement and good word appendant, make it hard to generate the same e-mail fingerprints for highly similar spam e-mails.

II. Statistical Data Compression

Probability plays a central role in data compression: Knowing the exact probability distribution governing an information source allows us to construct optimal or near-optimal codes for messages produced by the source. A statistical data compression algorithm

Author α : M.Tech, Asst. Professor, IT Dept, Sir. C. R. Reddy College of Engg, Eluru, A.P. India. E-mail : prasad17gumpina@gmail.com
Author Ω : M.Tech, Asst. Professor, CSE Dept, Avanthi Institute of Engineering and Technology, Makavarapalem, Visakhapatnam. E-mail : balasriram1982@gmail.com

exploits this relationship by building a statistical model of the information source, which can be used to estimate the probability of each possible message. This model is coupled with an encoder that uses these probability estimates to construct the final binary representation. For our purposes, the encoding problem is irrelevant. We therefore focus on the source modeling task

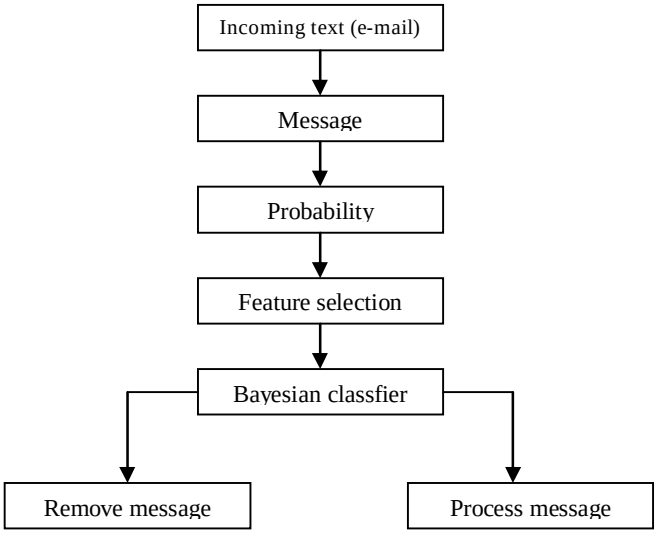
III. Preliminaries

We denote by *X* the random variable associated with the source, which may take the value of any message the source is capable of producing, and by *P* the probability distribution over the values of *X* with the corresponding probability mass function *p*. We are particularly interested in modeling of text generating sources. Each message **x** produced by such a source is naturally represented as a sequence $\mathbf{X}=x1^{n} = x1....xn$ $\mathbf{\Sigma}^*$of symbols over the source alphabet $\mathbf{\Sigma}$. The length of a sequence can be arbitrary. For text generating sources, it is common to interpret a symbol as a single character, but other schemes are possible, such as binary (bitwise) or word-level models. The entropy *H (X)* of a source *X* gives a lower bound on the average per-symbol code length required to encode a message without loss of information: $H(x) = E_{x\sim p}(-\frac{1}{|x|}\log p(x))$

This bound is achievable *only* when the true probability distribution *P* governing the source is known. In this case, an average message could be encoded using no less than *H(X)* bits per symbol. However, the true distribution over all possible messages is typically unknown. The goal of any statistical data compression algorithm is then to infer a probability mass function over sequences $f:\sum^* \rightarrow [0,1]$, which matches the true distribution of the source as accurately as possible. Ideally2, a sequence **x** is then encoded with $L(\mathbf{x})$ bits, where $L(\mathbf{x}) = -\log f(\mathbf{x})$. The compression algorithm must therefore *learn* an approximation of *P* in order to encode messages efficiently. A better approximation will, on average, lead to shorter code lengths. This simple observation alone gives compelling motivation for the use of compression algorithms in text categorization.

IV. Bayesian Spam Filtering

Bayesian spam filtering can be conceptualized into the model presented in Figure 1. It consists of four major modules, each responsible for four different processes: message tokenization, probability estimation, feature selection and Naive Bayesian classification.



When a message arrives, it is firstly tokenized into a set of features (tokens), F . Every feature is assigned an estimated probability that indicates its spaminess. To reduce the dimensionality of the feature vector, a feature selection algorithm is applied to output a subset of the features. The Naive Bayesian classifier combines the probabilities of every feature in 1 F , and estimates the probability of the message being spam. In the following text, the process of Naive Bayesian classification is described, followed by details concerning the measuring performance. This order of explanation is necessary because the sections concerned with the first three modules require understanding of the classification process and the parameters used to evaluate its improvement.

V. Performance Evolution

Precision and recall a well employed metric for performance measurement in information retrieval is precision and recall. These measures have been diligently used in the context of spam classification (Sahami et al.1998). Recall is the proportion of relevant items that are retrieved, which in this case is the proportion of spam messages that are actually recognized. For example if 9 out of 10 spam messages are correctly identified as spam, the recall rate is 0.9. Precision is defined as the proportion of items retrieved that are relevant. In the spam classification context, precision is the proportion of the spam messages classified as spam over the total number of messages classified as spam. Thus if only spam messages are classified as spam then the precision is 1. As soon as a good legitimate message is classified as spam, the precision will drop below 1. Formally: Let gg n be the number of good messages classified as good (also known as false negatives). Let gs n be the number of good messages classified as spam (also known as false positives).(9). Let ss n be the number of spam messages classified as spam (also known as true

positives). Let sg n be the number of spam messages classified as good (also known as true negatives). The precision calculates the occurrence of false positives which are good messages classified as spam. When this happens p drops below 1. Such misclassification could be a disaster for the user whereas the only impact of a low recall rate is to receive spam messages in the inbox. Hence it is more important for the precision to be at a high level than the recall rate. The precision and recall reveal little unless used together. Commercial spam filters sometimes claim that they have an incredibly high precision value of 0.9999% without mentioning the related recall rate. This can appear to be very good to the untrained eye. A reasonably good spam classifier should have precision very close to 1 and a recall rate > 0.8. A problem when evaluating classifiers is to find a good balance between the precision and recall rates. Therefore it is necessary to use a strategy to obtain a combined score. One way to achieve this is to use weighted accuracy.

VI. Cross Validation

There are several means of estimating how well the classifier works after training. The easiest and most straightforward means is by splitting the corpus into two parts and using one part for training and the other for testing. This is called the holdout method. The disadvantage is that the evaluation depends heavily on which samples end up in which set. Another method that reduces the variance of the holdout method is k - fold cross-validation. In k -fold cross-validation (Kohavi 1995) the corpus, M , is split into k mutually exclusive parts, k M ,M ,...M 1 2 . The inducer is trained on i M \ M and tested against i M . This is repeated k times with different i such that iÎ{1,2,...,k}. Finally the performance is estimated as the mean of the total number of tests.

VII. Conclusion

Optimal search algorithm called SFFS was applied to find a subset of delimiters for the tokenizer. Then a filter and a wrapper algorithm were proposed to determine how beneficial a group of delimiters is to the classification task. The filter approach ran about ten times faster than the wrapper, but did not produce significantly better subsets than the base-lines. The wrapper did improve the performance on all corpuses by finding small subsets of delimiters. This suggested an idea concerning how to select delimiters for a near-optimal solution, namely to start with space and then add a few more. Since the wrapper generated subsets had nothing in common apart from space, the recommendation is to only use space as a delimiter. The wrapper was far too slow to use in spam filter.

References Références Referencias

1. Almuallim, H. and T. Dietterich. (1991), Learning with many irrelevant features. In Proceedings of the Ninth National Conference on Artificial Intelligence, pp. 547-552. Menlo Park, CA: AAAI Press/The MIT Press.
2. Androutsopoulos I., Paliouras G., Karkaletsis V., Sakkis G., Spyropoulos C. and Stamatopoulos, P. (2000a) Learning to filter spam email: A comparison of a naive bayesian and a memory-based approach. In Workshop on Machine Learning and Textual Information Access, 4th European
3. Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000). Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, George and Spyropoulos, C.D. (2000b),
4. An Evaluation of Naive Bayesian Anti-Spam Filtering. In Potamias, G., Moustakis, V. and van Someren, M. (Eds.), Proceedings of the Workshop on Machine Learning in the New
5. Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain, pp. 9-17.
6. Androutsopoulos, I., Paliouras, G., Michelakis, E. (2004), Learning to Filter Unsolicited Commercial E-Mail. Athens University of Economics and Business and National Centre for Scientific Research "Demokritos" Bevilacqua-Linn M. (2003),
7. Machine Learning for Naive Bayesian Spam Filter Tokenization Breiman, L., and Spector, P. (1992), Submodel selection and evaluation in regression: The Xrandom case. International Statistical Review, 60, 291-319.
8. Androutsopoulos, G. Paliouras, and E. Michelakis. Learning to filter unsolicited commercial e-mail. Technical Report 2004/2, NCSR "Demokritos", October 2004.
9. F. Assis, W. Yerazunis, C. Siefkes, and S. Chhabra. CRM114 versus Mr. X: CRM114 notes for the TREC 2005 spam track. In *Proc. 14th Text REtrieval Conference (TREC 2005)*, Gaithersburg, MD, November 2005.
10. A.R. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998. D. Benedetto, E. Caglioti, and Loreto V. Language trees and zipping. *Physical Review Letters*, 88 (4), 2002.
11. A.Bratko and B. Filipi Spam filtering using character-level markov models: Experiments for the TREC 2005 Spam Track.

This page is intentionally left blank