A Computational Study of Naïve Bayesian Learning in Anti-spam Management

Zhiwei Fu and Isa Sarac

Virginia International University,
3957 Pender Drive, Fairfax, VA 22030 USA
{zfu,isarac}@viu.edu
http://www.viu.edu

Abstract. It has been argued that Bayesian learning can be used to filter unsolicited junk e-mail ("spam") and outperform other anti-spam methods, e.g., the heuristics approaches. We develop a Bayesian learning system, and conduct a computational study on a corpus of 10,000 emails to evaluate its performance and robustness, particularly the performances with different training-corpus sizes and multi-grams. Based on the computational results, we conclude that the Bayesian anti-spam approach is promising in anti-spam management as compared with other methods at the client side, and may need additional work to be viable at the corporate level in practice.

1 Introduction

As information technology fast advances forward, unsolicited commercial e-mail ("spam") has become an ever-increasing problem. The statistics has shown that 10.4 million spam emails are sent every minute worldwide, and the spam has at least quadrupled in the past two years. The problems caused by unsolicited commercial e-mail ("spam") go well beyond the annoyance spam causes to the public. These problems include the fraudulent and deceptive content of most spam messages, the sheer volume of spam being sent across the Internet, and the security issues raised because spam can be used to disrupt service or as a vehicle for sending viruses [1]. A recent Gartner Group survey revealed that 34% of business email is useless. The same study also revealed that employees spend an average of 49 minutes a day managing email.

President George W. Bush signed a landmark anti-spam bill into law in December 2003 that became effective Jan. 1, 2004, setting into motion the first national standards for sending bulk unsolicited commercial e-mail (UCE). Pre-empting many tougher state anti-spam laws, the Can Spam Act aims to curb the most egregious practices of spammers by targeting e-mail with falsified headers, but allows e-marketers to send UCE as long as the message contains an opt-out mechanism, a functioning return e-mail address, a valid subject line indicating the e-mail is an advertisement and the legitimate physical address of the mailer. However, some critics say that it legitimizes spam by allowing sending unsolicited commercial email as long

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 824–830, 2004.

as it has an opt-out mechanism, a functioning return e-mail address, a valid subject line indicating the email is an advertisement and the legitimate physical address of the mailer. Therefore, improved technological tools will be an essential part of any solution as well. Currently there are three major types of anti-spam approaches, 1) source-based approach of managing sender identity, e.g., Black lists, White lists, Real Time Black lists (RBL), Reverse DNS Lookups. Source-based approaches are perhaps most effective in levering system resources but they are likely to yield poor overall hit rate. 2) Rule-based content analysis, e.g., pattern matching, spam definitions, heuristics, and the approaches of this kind often use scoring techniques. However, rule-based approaches tend to go stale as spammers move on to new tricks. Keeping rule-based up-to-date and effective requires a great amount of human resources to come up with new rules. Rule-based is generally considered more accurate than source-based approaches. In this category, the heuristics approach applies multiple detection tests to provide greater confidence in identifying spam messages [2]. 3) Bayesian learning, a type of statistical approach to identify spam based on their characteristics that have been learned from existing emails categorized by users and then apply the knowledge to new incoming emails. Bayesian analysis has been considered most accurate approach, especially at the client side, to effectively tackle fast-changing spam. But the Bayesian approach consumes heavy computation.

As we emphasize on accuracy than cost, Bayesian analysis became the major focus of this paper, in comparison to the heuristics approach. In this paper, we will first describe naïve Bayesian learning in section 2, and our learning system, followed by our computational study, including experiment design and the computational results; and section 4 provide concluding remarks and some future work.

2 Bayesian Learning

Naïve Bayesian learning is the optimal classification method of supervised learning if the values of the attributes of an example are independent given the class of the example [3]. On many real-world example datasets Bayesian learning gives better test set accuracy than any other known method, including backpropagation [7] and C4.5 decision trees [6].

Let A_1 through A_k be attributes with discrete values used to predict a discrete class C. Given an example with observed attribute values a_1 through a_k, the optimal prediction is class value c such that $P(C = c \mid A_1 = a_1 \cap ... \cap A_k = a_k)$ is maximal. By Bayes' rule this probability equals

$$\frac{P(A_1 = a_1 \cap ... \cap A_k = a_k \mid C = c)}{P(A_1 = a_1 \cap ... \cap A_k = a_k)} \cdot P(C = c).$$

The background probability or base rate $P(C = c)$ can be estimated from training data easily. The example probability $P(C = c \mid A_1 = a_1 \cap ... \cap A_k = a_k)$ is irrelevant for decision-making since it is the same for each class value c. Learning is therefore reduced to the problem of estimating $P(A_1 = a_1 \cap ... \cap A_k = a_k \mid C = c)$ from training

examples. Using Bayes' rule again, this class-conditional probability can be written as

$$P(A_1 = a_1 \mid A_2 = a_2 \cap ... \cap A_k = a_k, C = c) \cdot P(A_2 = a_2 \cap ... \cap A_k = a_k \mid C = c).$$

Recursively, the second factor above can be written as

$$P(A_2 = a_2 \mid A_3 = a_3 \cap ... \cap A_k = a_k, C = c) \cdot P(A_3 = a_3 \cap ... \cap A_k = a_k \mid C = c)$$

and so on. Now suppose we assume for each A_i that its outcome is independent of the outcome of all other A_j, given C. Formally, we assume that

$$P(A_1 = a_1 \mid A_2 = a_2 \cap ... \cap A_k = a_k, C = c) = P(A_1 = a_1 \mid C = c)$$

and so on for A_2 through A_k. Then $P(A_1 = a_1 \cap ... \cap A_k = a_k \mid C = c)$ equals

$$P(A_1 = a_1 \mid C = c) \cdot P(A_2 = a_2 \mid C = c) ... P(A_k = a_k \mid C = c)$$

Now each factor in the product above can be estimated from training data:

$$\hat{P}(A_j = a_j \mid C = c) = \frac{count(A_j = a_j \cap C = c)}{count(C = c)}$$

It can be shown that the above equation gives "maximum likelihood" probability estimates, i.e. probability parameter values that maximize the probability of the training examples. The above induction algorithm is called naïve Bayesian learning [4].

3 Computational Experiments

3.1 Experimental Design

We build our Bayesian learning system based on the above induction algorithm, i.e., the naive Bayesian learning. Our Bayesian learning system is developed to take the whole message into account. Each email is treated as a data record, and the score of the email is collectively determined by the spam characteristics of each and every word in the email. Specifically, the Bayesian learning system not only recognizes keywords that identify spam, but can also recognize words that denote valid mail. For example: not every email that contains the word "free" and "cash" is spam. Our learning system would be able to recognize the name of the business contact that sent the message and thus classify the message as legitimate, and therefore, allows words to "balance" each other. Our learning system also inherits Bayesian's self-adaptation to evolve itself by constantly learning from new spam and new valid outbound mails. For example, when spammers started using "f-r-e-e" instead of "free" they succeeded in evading keyword checking until "f-r-e-e" was also included in the keyword database. But our Bayesian learning system is able to automatically notice such tactics; in fact if the word "f-r-e-e" is found, it is an even better spam indicator. In addition, an initial training database of emails is formed based on inbound emails for our Bayesian learning system and the database is then updated based on the training results. The Bayesian learning will continuously learn the characteristics from the inbound emails and apply the learning to categorize the prospective emails.

We designed and developed our semantic Bayesian learning system to filter spam (junk and unsolicited commercial emails) from hams (the legitimate emails). Our experiments are intended to test the performance and robustness of our Bayesian approach as compared with some anti-spam heuristics scoring approach. We have collected about 10,000 emails to obtain unbiased results. In general, the resampling techniques provide reliable estimates of the true error rate, because nearly all the data points used for training, and all data points are used for testing, and the Bayesian classifier can therefore be reapplied to all data points. In our experiments, we use resampling techniques, i.e., repeated train-and-test partitions, to estimate the classification error rate. In particular, we use 10-fold cross-validation to make our results less prone to random variation and to further compare statistically across different anti-spam approaches [5].

3.2 Computational Results

In our experiments, we apply both rule-based heuristics approach and our Bayesian learning system on a corpus of the 10,000 emails. The overall distribution of spam and ham obtained from the heuristics approach given in Fig.1 indicates a non-trivial set separation of ham and spam for the heuristics approach. The modes of spam and ham in Fig.1 are mainly located in the middle of the scoring spectrum with tails of both "clearly" identified spam and ham extended to both ends. On the contrary, Bayesian learning system provides a desired "clear-cut" set separation of ham from spam (as given in Fig.2). The modes of spam and ham are located far apart at the ends of the entire spectrum of spam scores for all emails, while there are some overlaps (or the area with uncertain emails based on their spam scores) in the middle of the spectrum. Ideally, every email should be correctly classified with no uncertainty and there is no false positive and false negative, and then there would be no overlaps in the middle of the spectrum. Nevertheless, the distribution of spam/ham scores obtained by the Bayesian approach (as in Fig.2) apparently looks more promising in practice than that by the heuristics approach (as in Fig. 1).

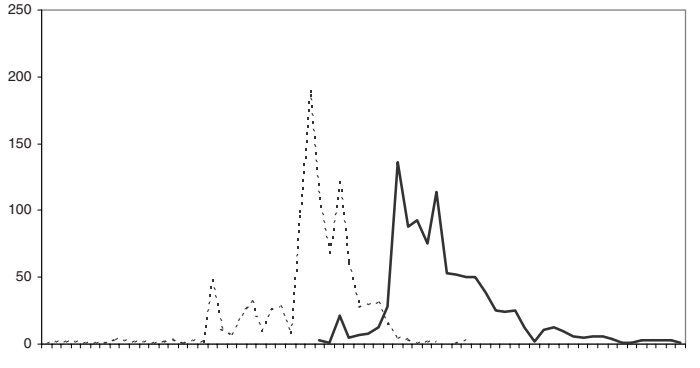

Fig. 1. Overall spam (the solid line) and ham (the dotted line) distribution obtained by the heuristics approach. The horizontal axis is the spectrum of spam scores for all emails, and the vertical axis is the frequency of emails by spam scores.

Anti-spam approaches such as Bayesian learning require some level of training, and almost all anti-spam filtering systems use unigrams (a single 1-word approach) to filter spam. However, the English language becomes more "structured" if we use bigrams and trigrams. Therefore, we study the performance of multi-grams, i.e., *n*-word approaches (n = 1,2,3) combined with the training sets of different sizes in our experiments. As given in Table 1, the heuristics approach provides good error estimates with average classification accuracy above 93%, FP (false positive) and FN (false negative) about 10.5% and 2.5%, respectively. In particular, 1000 training size provides the best performance for all training sizes. It indicates that a better quality training set with reasonable size would be more sufficient and effective than a less quality but larger training set, i.e., "garbage in, garbage out". On the other hand, multi-grams, trigrams and bigrams in our experiments have not demonstrated statistical significance in their performances over unigrams.

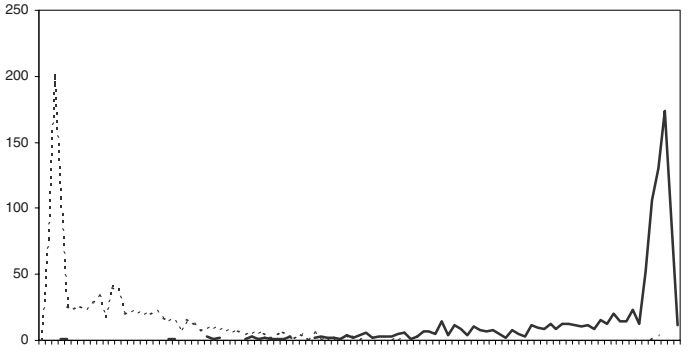

Fig. 2. Overall spam (the solid line) and ham (the dotted line) distribution obtained by our Bayesian learning approach. The horizontal axis is the spectrum of spam scores for all emails, and the vertical axis is the frequency of emails by spam scores.

Bayesian learning has demonstrated significantly better performances in all regards as given in Table 2 than the heuristics approach. The classification accuracy and FP by the Bayesian learning are on average 97% and 2%, respectively. We then conducted paired difference *t*-test, and the *p*-values of paired difference *t*-tests of overall classification accuracy for different combinations are given in Table 3 and Table 4. As shown, the computational results from both the heuristics approaches and our Bayesian approach also find no statistical significance among different training sizes. However, the performance of accuracy for *n*-word approaches has shown some improvements from unigrams and bigrams, to trigrams. As shown, multi-grams outperform unigrams for training size of 1000, but show no significance statistically for other training sizes. Although the experiments here may not be conclusive statistically, the results strongly indicate again that applying Bayesian unigram learning with a good quality training set with reasonable size should be sufficiently effective in anti-spam management at the client side.

Table 1. Computational results of classification accuracy, false positive (FP), and false negative (FN) obtained by the heuristics approach (an experiment with 1000 training set using 1-word approach is denoted by 1000 1-word).

	Accuracy	FP	FN
1000-1w ord	94.06%	7.99%	3.90%
1000-2w ord	94.91%	6.79%	3.40%
1000-3w ord	97.50%	2.20%	2.80%
2000-1w ord	92.85%	12.59%	1.70%
2000-2w ord	92.68%	13.24%	1.40%
2000-3w ord	92.85%	12.64%	1.65%
3000-1w ord	92.93%	12.46%	2.13%
3000-2w ord	92.56%	13.37%	2.00%
3000-3w ord	92.91%	12.54%	2.10%
Mean	93.69%	10.43%	2.34%
StdErr	0.005	0.012	0.003

Table 2. Computational results, classification accuracy, false positive (FP), and false negative (FN) obtained by our Bayesian approach.

	Accuracy	FP	FN
1000-1w ord	96.80%	1.80%	4.60%
1000-2w ord	97.75%	1.70%	2.80%
1000-3w ord	98.75%	0.90%	1.60%
2000-1w ord	96.78%	1.75%	4.70%
2000-2w ord	96.85%	2.10%	4.20%
2000-3w ord	96.90%	2.10%	4.10%
3000-1w ord	96.19%	2.87%	4.67%
3000-2w ord	96.37%	3.02%	4.20%
3000-3w ord	96.37%	2.94%	4.27%
Mean	96.97%	2.13%	3.90%
StdErr	0.003	0.002	0.003

Table 3. *p*-values of paired difference *t*-test of overall classification accuracy obtained by the heuristics approach (the results are computed based on row methods over column methods, and 1000-1word is denoted by 1k-1w).

vs.	1K-1w	1K-2w	1K-3w	2k-1w	2k-2w	2k-3w	3k-1w	3k-2w	3k-3w
1K-1w				0.0199	0.0120	0.0191	0.0074	0.0023	0.0055
1K-2w	0.0942			0.0026	0.0008	0.0023	0.0016	0.0005	0.0009
1K-3w	0.0006	0.0002		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2k-1w					0.1044			0.1010	
2k-2w								0.3184	
2k-3w				0.5000	0.1207			0.1451	
3k-1w				0.3845	0.1735	0.3860		0.0078	0.4158
3k-2w									
3k-3w				0.4128	0.1843	0.4101		0.0126	

Table 4. *p*-values of paired difference *t*-test of overall classification accuracy obtained by the Bayesian approach (the results are computed based on row methods over column methods, and 1000-1word is denoted by 1k-1w).

vs.	1K-1w	1K-2w	1K-3w	2k-1w	2k-2w	2k-3w	3k-1w	3k-2w	3k-3w
1K-1w				0.4734	0.4486	0.3917	0.0488	0.0825	0.1084
1K-2w	0.0137			0.0319	0.0346	0.0455	0.0027	0.0021	0.0046
1K-3w	0.0000	0.0058		0.0003	0.0004	0.0004	0.0000	0.0000	0.0000
2k-1w					0.2796			0.1089	
2k-2w								0.0525	
2k-3w				0.1357	0.2950			0.0437	
3k-1w				0.0246	0.0103	0.0057		0.0934	0.0534
3k-2w									
3k-3w				0.1029	0.0480	0.0399		0.4997	

4 Concluding Remarks

Fighting spam is not a trivial undertaking, especially at the server side or corporate level. For example, the judgment on spam/ham is highly individual, and the unsolicited commercial emails that one recipient would accept/reject could be significantly different from another recipient based on the recipients' interests. As shown in our experiments, the Bayesian learning approach is advantageous as compared to other heuristics approaches, but it works best at client side. In addition, consumes significantly more computational resources, and another major concern resides in maintaining a good database for continuous training. As shown in the study, multi-grams look appealing but need more study for its significance. In our future work, we will research the robustness of implemented at gateway level for corporations and ways to improve the performance of our Bayesian learning.

References

1. Baker, S. (2003). The Taming of the Internet, Business Week Magazine, December 2003.
2. Conry-Murray, A. (2003). Fighting the Spam Monster - and Winning, Network Magazine, April 2003.
3. Elkan, C. (1997). Naïve Bayesian Learning, Technical Report No. CS97-557, University of California, San Diego. Baldonado, M., Chang, C-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121
4. Gelman, A., J. Carlin, H. Stern, and D. Rubin (2000). Bayesian Data Analysis, New York, NY: Chapman & Hall/CRC.
5. Hastie, T., R. Tibshirani, and J. Friedman (2001). The Elements of Statistical Learning – Data Mining, Inference, and Prediction, New York, NY: Springer-Verlag.
6. Quinlan, J. R. (1993). C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann.
7. Riply, B. (1996). Pattern Recognition and Neural Networks, Cambridge, MA: Cambridge University Press.