PRIS Kidult Anti-SPAM Solution at the TREC 2005 Spam Track: Improving the Performance of Naive Bayes for Spam Detection

Yang Zhen, Xu Weiran, Chen Bo, Hu Jiani, Guo Jun

PRIS Lab, School of Information Engineering,
Beijing University of Posts and Telecommunications,
Beijing, 100876, China
yangzhen@pris.edu.cn

Technical Report KidSPAM xxxx

Abstract. Recently, the spam already constituted a serious problem for both e-mail users and Internet Service Providers (ISP). Solutions to the abuse of spam would be both technical and legal regulatory. This paper reports our solution for the TREC 2005 spam track, in which we consider the use of Naive Bayes spam filter for its desirable properties (simplicity, low time and memory requirements, etc.). Then the approaches to modify the Naive Bayes by simply introducing weight and classifier assemble based on dynamic threshold are proposed, which can help to improve the accuracy of a Naive Bayes spam classifier dramatically. Additionally, we discuss some steps that must be adopted naturally thought before, such as stop list, word stemming, feature selection, class prior probabilities. The theory analysis implies these steps are not necessarily the best way to extend the Bayesian classifier, and these were also verified empirically. Many of these techniques appear to be counterintuitive but can be explained by the statistical properties of e-mail itself. Experiment results of TREC 2005 spam track demonstrate the effectiveness of the proposed method.

1 Introduction

Recently, the spam already became a serious actual problem, which merely was a latent threat several years ago [8] [12] [14]. Though the purpose of e-mail is to make communication more convenient, e-mail does not always provide the increased efficiency desired. World widely, spam is estimated to comprise 69% of global e-mail and the percentage of spam has risen steadily during the past couple of years [14]. The growing volumes of spam causes huge losses to both e-mail users and ISP due to bandwidth consumption, storage space, mail server processing load, user's efficiency (time spent for responding, deleting or forwarding etc[12] [14]).

Though there is no standardized definition for spam, the TREC 2005 track's definition of spam is "*Unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship* [24]." Generally speaking, the spam detection belongs to problem of the information security domain. People stressed in formal securities, such as the confidentiality, integrality and availability. So in the first time, we doubt whether the spam detection can be treated as a special problem in text categorization or not. But the TREC 2005 spam track let us see the successful application of statistical method based on content. More importantly, we realize that the modern information security should not only be the formal security but also be the content security, which users can enjoy information sharing, in the same time avoid the information abusing in the greatest degree.

Traditional techniques cope with spam include header analysis and tracking based on sender address or header content, digital signatures, keyword/keyphrase matching and analogous rule-based predicate [14]. The problem with traditional techniques is that sometimes a valid message may be blocked. Furthermore, in realistic terms, this is really about 1 to 3 truly new spams or spam methods per month. Instead of blocking spam simply, this work aims at deciding whether or not the latent subject matter is consistent with the user's interests.

Statistical filtering tends to automatically reject e-mail that is classified as spam relying on user's intent. Therefore, the statistical learning techniques are more suitable for spam detection. The state of art include rule learning, Naive Bayes, memory based learning, decision trees, support vector machines or combinations of different learners [3] [6] [15] [16] [17]] [18] [19]. Among these techniques, Bayesian methods [1] [2] [4] [5] [9] and their improvements [10] [11] are particularly attractive, because they more formally model the relationship between the content of the spam and the reading favors of the user.

Naive Bayes uses a simple probabilistic model that makes strong assumptions about the data: it assumes that words in a document are independent. Clearly, this assumption is violated in most natural language text; therefore, some techniques including augmented Naive Bayesian network and augmented Naive Bayes (ANB) [20] are proposed to relax independent assumption. Nonetheless, Naive Bayes performs quite well in practice even when attributes are not independent [1] [2], often comparable to more sophisticated learning methods.

In this report, a spam detection filter framework is proposed based on Naive Bayes for its desirable properties. And then the approaches to modify the Naive Bayes by simply introduce weight and classifier assemble based on dynamic threshold are proposed. Additionally, we discuss some steps that must be adopted naturally thought before, such as stop list, word stemming, feature selection, class prior probabilities. The theory analysis implies these steps are not necessarily the best way to extend the Bayesian classifier, and these were also verified empirically. Many of these techniques appear to be counterintuitive but can be explained by the statistical properties of e-mail itself. Experiment results of TREC 2005 spam track demonstrate the effectiveness of the proposed method.

The resulting technology has been successfully released in TREC 2005 spam track system 'Kidult Anti-SPAM Solution'. Another rich on-line resources and information about anti-spam include http://plg.uwaterloo.ca/~gvcormac/spam/, http://www.ceas.cc/, and http://www.spam.com.cn/.

2 Naive Bayes Spam Detection Filter Framework

2.1 Naive Bayes

Naive Bayes is often used in text classification applications and experiments for its simplicity and effectiveness [1] [2] [5] [9]. And spam detection poses a special problem in text categorization. The Naive Bayes classifier is a probability based approach. The basic concept of it is to find whether an e-mail is spam or not by looking at which words are found in the message and which words are absent from it [7] [13].

In the literature, the Naive Bayes classifier is defined as follows:

$$C_{NB} = \arg\max_{i \in L} P(C_i) \prod_k P(w_k \mid C_i) \tag{1}$$

The e-mail composes of w_k words, where L is the set of target classes. There are several Naive Bayes models that make different assumptions about how documents are composed from the basic units. The most common models are: multi-variate Bernoulli model, Poisson Naive Bayes model, and the multinomial model [22]. The most apparent difference between these models is ways of $P(w_k \mid C_i)$ calculation. In this work, $P(w_k \mid C_i)$ is calculated using multinomial model for its superior performance [22].

For spam detection, there are only two classes (C_+ spam/C_- ham), the score of an input e-mail M calculated as follow, and the logarithm formula of (1) is used:

$$\text{score(M)}=\log P(C_+)+\sum_k \log P(w_k \mid C_+)-(\log P(C_-)+\sum_k \log P(w_k \mid C_-)) \tag{2}$$

Therefore, if score(M) > 0, the email will be assigned to C_+, and C_- otherwise.

2.2 Spam Detection Filter Framework

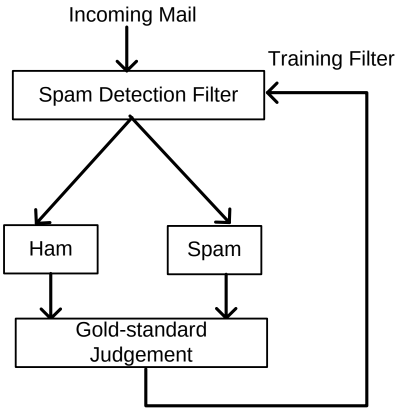

Fig. 1. Spam Detection Filter Framework

In TREC 2005 spam track, spam filtering was defined as a continuously and most practical applications based on online user feedback with fast, incremental and robust learning algorithms. The framework show in Fig. 1 [24], which supervised filtering involves the filter and recipient in a closed loop. The recipient regularly examines the ham and spam files and reports misclassifications according to gold-standard back to the filter, which updates its memory accordingly. Obviously, most of mail process terminal software (Microsoft^{TM} Outlook^{TM} etc.) adopts this model, and the only difference is that the performance of the filter is measured by users [24].

2.3 Performance Criteria

In TREC 2005 spam track, Evaluation will be based on these measures:

a) HMR: Ham Misclassification Rate, the fraction of ham messages labeled as spam.

b) SMR: Spam Misclassification Rate, the fraction of spam messages labeled as ham.

c) LAM: Logistic average misclassification.

d) 1-ROCA: Area above the ROC curve.

3 Feature Representation and Selection

Feature representation and selection is one of the important links of text categorization. Many complex techniques, such as natural language understanding (NLU) and named entity recognition, can help us to achieve better textual understanding, and enable machine work more intelligently. But if our goal merely is simple level processing of text (e.g. text categorization, spam detection), not involving the higher level application (e.g. natural language's meaning's understanding and the ambiguous natural language's meaning of the computer's understanding). Perhaps the usage of these complex techniques without fully understanding led to performance degradation. In this section, we discuss some steps that must be adopted naturally thought before, such as stop list, word stemming, feature selection, class prior probabilities. The theory analysis implies these steps are not necessarily the best way to extend the Bayesian classifier.

3.1 Feature Representation Pretreatment

For text classifying, the feature is a word. Our filter does not use HTML tags as tokens. Such tags, as well as other information such as images, links and attachment are simply eliminated. In typical text classifying tasks, some measures were often considered:

3.1.1 Word Stemming

Word stemming is similar to cluster, which can cluster the words with same stem together. Use of word stemming can lowers the size of the feature vector, but it may be the case that certain forms of a word (such as sex and sexy) were important for classification.

Consider a mail M, described by two attributes *A*, *B*. Assume that the two classes, denoted by + and -, are equiprobable $P(+) = P(-) = 1/2$. The optimal classification procedure for test mail is to assign it to class + if

$$P(A|+)P(B|+) - P(A|-)P(B|-) > 0 \tag{3}$$

and to class - if the inequality has the opposite sign, and to an arbitrary class if the two sides are equal.

Applying Bayes' theorem, $P(A|+)$ can be rewritten as $P(A)P(+|A)/P(+)$, and similarly for the other probabilities. Since $P(+) = P(-)$, after canceling like terms this leads to the equivalent expressions

$$P(+|A)P(+|B) - P(-|A)P(-|B) > 0 \tag{4}$$

for the optimal decision. Let $P(+|A) = p$ and $P(+|B) = q$. Then class + should be selected when $pq - (1 - p)(1 - q) > 0$, which is equivalent to $q + p > 1$.

And if A, B have the same stem C, then the mail have only one attribute C after word stemming. The optimal decision function became (using multinomial model):

$$P(+|C) – P(- |C)>0 \tag{5}$$

$P(+|C) = P(+)P(C|+)/P(\mathrm{C})=(p+q)/2P(\mathrm{C})$. And the $P(\mathrm{C}) = P(A)+P(B)$, then class + should be selected when $p+q > P(C)$, which is equivalent to $q + p > P(A)+P(B)$. So the classified frontier always moves toward one direction for $P(C) < 1$. Fortunately, with the increase of words in a special mail, the influence of word stemming gradually became small. For an n-words mail (A, B, $\{w_i\}_{i=3}^{n}$), A, B have the same stem. The classification scores are dominated by the words probabilities, and the *A*, *B* hardly affect the classification for longer documents $\{w_i\}_{i=3}^{n}$.

3.1.2 Stop List

Words like "of," "and," "the," etc., are used to form a stop list. Words on the stop list are not used in forming a feature vector. The rationale for this is that common words are less useful in classification. The argument against using a stop list is that it is not obvious which words, beyond the trivial, should be on the stop list. The choice of words to put on a stop list is probably a function of the classification task and it would be better if learning algorithm itself determined whether a particular word is important or not. For example, financial articles may be distinguished by a prevalence of numeric dollar figures, which may well be discarded wholesale by a preprocessor.

The most important things are that the Naive Bayes is not sensitive to stop list. Similarly, for an n-words mail $(\{w_i\}_{i=1}^{m}, \{w_j\}_{j=m+1}^{n})$, $\{w_i\}_{i=1}^{m}$ are the words in the stop list. Let $P(+|w_i)=k_i$,i=3,…,n, then the optimal decision function without using stop list is (logarithm formula of (1) is used):

$$\log\frac{\prod_{i=1}^{n}k_i}{\prod_{i=1}^{n}(1-k_i)}=\log\frac{\prod_{i=1}^{m}k_i}{\prod_{i=1}^{m}(1-k_i)}+\log\frac{\prod_{i=m+1}^{n}k_i}{\prod_{i=m+1}^{n}(1-k_i)}>0 \tag{6}$$

The optimal decision function using stop list is:

$$\log\frac{\prod_{i=m+1}^{n}k_i}{\prod_{i=m+1}^{n}(1-k_i)}>0 \tag{7}$$

For most application, these word in the stop list appear evenly in ham and spam mail, therefore, the influence of word stemming is small. For a typical mail text, the influence can be omitted.

3.2 Feature Selection

In typical text categorization solutions, we suppose that there have possible advantage of using a finite number of features rather than all. Some mechanisms designed to find the optimum number of features are document frequency ratios, information gain, mutual information, term strength, and χ^2 [16] [19].

One important reason for feature selection is the attributes dependent. Feature subset selection is hoped to improve accuracy on some data sets, but the effection varied from application to application depending on the learning algorithm. Especially, [1] [2] show Naive Bayes is optimal even when attributes are not independent. And the main disadvantage of searching for the best features is that it requires additional time in the training algorithm. It would be far better if the learning machine itself either made the feature selection automatically or used all the features. As implied in [1] [2], feature selection is not necessarily the best way to extend the Bayesian classifier.

3.3 Class Prior Probabilities

In typical text classification, ignoring prior probabilities altogether (or equivalently, assuming uniform priors) was widely employed. Because in practice, the classification scores are dominated by the word probabilities, and the prior probabilities hardly affect the classification for longer documents. However, in situations where documents are usually very short, especially in our spam detection filter framework (see figure 1), because the system comes in with an empty memory and learns what spam is, from the user. In such online feedback cases, the prior probabilities may be oscillatory and skewed and affect the classification obviously. In this work, we found the real class prior probabilities calculated from actual input stream can improve the accuracy of classifier, especially for these samples located around classifying hyperplane.

3.4 Conclusion

Preprocessing requires language dependent mechanisms like word stemming, stop list, feature selection that may not be readily available for the language now. Based on a review of the literature, in our 'Kidult Anti-SPAM Solution' for TREC 2005 spam track, we use all word as feature without stemming and discarding the words in stop list. And the actual class prior probability calculated from actual input stream was used. But our filter does not use HTML tags as tokens. Such tags, as well as other information such as images, links and attachment are simply eliminated.

4 Improving the Performance of Naive Bayes for Spam Detection

4.1 Weighted Naive Bayes

There are two possible approaches that can improve performance of Naive Bayes: (1) modify the data, (2) modify the classifier (or the probabilistic model) [11] [23]. Many researchers have proposed modifications to the way documents are presented, to better fit the independent assumptions made by Naive Bayes. This includes extracting more complex features, n-Gram, and word clustering. These methods did show some improvement of classification accuracy, but have been largely unsuccessful. And based on the above discuss, Naive Bayes is not sensitive to data and the effection varied from application to application. On the other hand, researchers have tried to improve the performance by using more complex probabilistic model alleviating the effection of independent assumption. This includes TAN (Tree augmented Naive Bayes), TAN assemble, and Bayes network [20]. These methods were very difficult to optimize even were the NP-hard problem.

In this section, an approaches to modify the Naive Bayes by simply introducing weight was proposed, which can help to improve the accuracy of a Naive Bayes spam classifier dramatically.

4.1.1 Framework

In text classification, the feature is word, and we always ignore syntax information altogether. Many researchers think this is the key making classifier degraded, and tried to employ more complex techniques to improve this model, such as Nature Language Processing (NLP). But these attempts have been largely unsuccessful, that did not provide a significant benefit on any natural data sets only for complex and invalid.

In this section, an improvement using simple syntax information is proposed. The basement of our scheme is that some parts of the document may have stronger dependence on the label than other parts. If every input e-mail can be divided into *S* components, and every component is composed of N_d, *d*=1,…,*S*, words($\{w_k^d\}$,*k*=1,…, N_d, *d*=1,…,*S*). The natural extend to the Naive Bayes is to introduce weight to every component, then the (1) can rewrite as:

$$C_{NB} = \arg\max_{i \in L} P(C_i)\{\prod_{d=1}^{S} \alpha_d \cdot \prod_{k=1}^{N_d} P(w_k^d \mid C_i)\} \qquad \textbf{(8)}$$

And the weight α_d , d=1,…,*S* is introduced. (9) is the logarithm formula of (8) is used):

$$C_{NB} = \arg\max_{i \in L}\{\log P(C_i) + \sum_{d=1}^{S}(\log \alpha_d + \sum_{k=1}^{N_d} \log P(w_k^d \mid C_i))\} \qquad \textbf{(9)}$$

(10) is the normalized formula using N_d that can denote the effection of text length:

$$C_{NB}^{norm} = \arg\max_{i \in L} P(C_i)\{\prod_{d=1}^{S}\prod_{k=1}^{N_d} {\alpha_d}^{N_d} \cdot P(w_k^d \mid C_i)\} \qquad \textbf{(10)}$$

Where α is chosen as follow, given a training set of *m* labeled e-mail ($(w^i, C^i)_{i=1}^m$), and a very natural criterion is to choose to maximize the log likelihood of the labeled training data:

$$\alpha = \arg\max_{\alpha} \sum_{i=1}^{m} \sum_{d=1}^{S} \{\log \alpha_d + \sum_{k=1}^{N_d} \log P((w_k^d)^i \mid C^i)\} \tag{11}$$

$\alpha = (\alpha_1 \alpha_2 ...... \alpha_S)$. It remains to specify how e-mail can be divided in different components.

a) The first natural extense was that dividing the text according the syntax structure (e.g. Title, header, paragraph, accessory). [11] employ this to USENET posting that can be considered to consist of a subject line and a body component. [11] achieve good performance, but it is not suitable for e-mail because the header and accessory were ignored and too short to use.

b) In this work, we divide the words of e-mail into different components by its probability. Consider a mail, described by n words. For a special word *w*, if *P*(*w*|+)>*P*(*w*|-), is to assign it to I^+ component, and assign to I^- component otherwise. This I^+ were though to be help for a e-mail assigned to spam, and I^- are help for a e-mail assigned to ham. So introducing weight were help to improve the performance of Naive Bayes under zero-one loss function [1] [2]. Weighted Naïve Bayes can modify the classified frontier adaptively according the characteristic of sample, which can improve the classify accuracy for these sample located around classifying hyperplane. And this method was used in our system.

4.2 Classifier Assemble

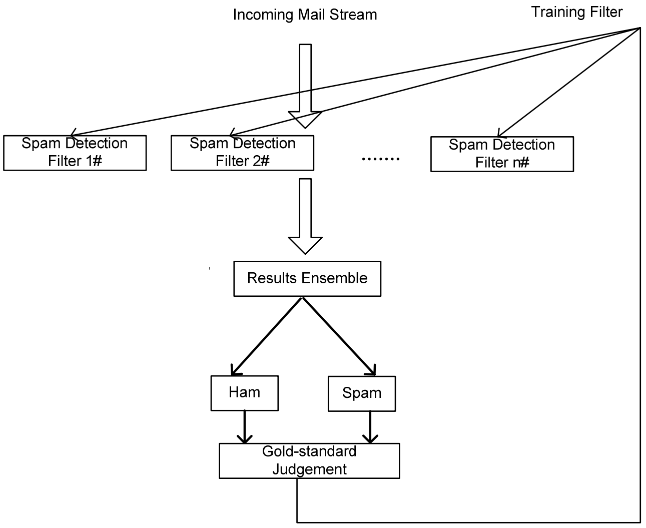

Fig. 2. Spam Detection Filter Framework Based on Naive Bayes Bagging Aggregate

The Bagging [3] predictor is a technique to combine a number of weak learners to form an ensemble. The Bagging predictor is a PAC (probably approximately correct) method to combine a number of weak learners with error rate slightly better than 50% to form an ensemble. In classification task, aggregating can transform predictors into nearly optimal ones [3]. Spam detection filter framework based on Naive Bayes bagging aggregate (see Fig. 2) can be naturally induced form the basic model (see Fig. 1). This version of Naive Bayes bagging works as following: the filtering involves the filter and recipient in a closed loop. The recipient regularly examines the ham and spam files and reports misclassifications according to gold-standard back to the random filter (Filter #1~ Filter #n), which updates its memory accordingly.

4.2.1 Naive Bayes Spam Filter Bagging Based on Dynamic Threshold

Unfortunately, the empirical results show that simply voting does not have the significant superiority compared to single classifier. This is partly because the number of filters is few and most of them give the same decision for a special e-mail.

Though it is widely known that combining multiple classification or regression models typically provides superior results compared to using a single, well-tuned model. However, the ways of combining multiple classifiers still is the important factor affecting the aggregated predictor performance. By using incremental decision tree induction (ITI) [26], the performance can be further improved by generating the assemble predictor from scores made by filter group instead of the binary decisions. Originally, every filter makes binary decision spam/ham for the incoming e-mail M, i.e., if score (M) > 0, the email will be assigned to spam, and ham otherwise. And then the assemble prediction was generated based on these binary decisions. But now the decision tree was generated using the score (M) itself. It means that we were not only to combine a number of weak learners to form an ensemble, but also to adjust the classified frontier of the Naive Bayes automatically by generating dynamic threshold for every filter. The assemble predictor based on dynamic threshold more correctly represent the difference of every classifier. The typical C4.5 [27] can work well, and ITI algorithm performs incremental decision tree induction on symbolic or numeric variables, and handles noise and missing values. Thus ITI can reduce the computational complexity significantly.

5 KidSPAM Filter Results of TREC 2005 Spam Track

In this section, we report the test results on eight email datasets provided by TREC 2005 spam track. The basic statistics for all eight datasets are given in Table 1. Further details about these corpora were described in [24] [25]. And the performance of Kidult Anti-SPAM Solution is given in Table 2 -Table 3. Any detail results and the comparison of all participant filters were published in [24] [25].

Table 1. Corpus Statisitic

		Ham	Spam	Total
Public Corpora	Trec05p-1/full	39399	52790	92189
	Trec05p-1/ham25	9751	52790	62541
	Trec05p-1/ham50	19586	52790	72376
	Trec05p-1/spam25	39399	13179	52578
	Trec05p-1/spam50	39399	26283	65682
Private Corpora	Mr X	9038	40048	49086
	SB	6231	775	7006
	TM	150685	19516	170201

6 Discussions and Conclusion

Spammers continue to devise aggressive and devious techniques, and we are taking a multi-faceted approach to fighting spam. Statistical filter is considered as a possible solution to protecting consumers from spam. Though technology can help to reduce the attacks but it is never complete and by itself cannot meet all the realistic needs.

Solutions to the abuse of spam would be both technical and legal regulatory [8]. In this report, we introduce the work of PRIS lab for TREC 2005 spam track. Future work may be directed towards developing better algorithms for our system, including more valid algorithm for weight selection, and more valid assemble algorithm based on incremental decision tree induction for numeric variables.

Table 2. kidSPAM1- kidSPAM2 Results

Corpus	Ham Misc%		Spam Misc%		Logit av Misc%		(1-ROCA)%	
	kidSPAM1	kidSPAM2	kidSPAM1	kidSPAM2	kidSPAM1	kidSPAM2	kidSPAM1	kidSPAM2
Trec05p	0.91	0.87	9.40	10.53	2.99	3.11	1.463	4.544
	(0.81-1.00)	(0.78-0.97)	(9.15-9.65)	(10.27-10.79)	(2.83-3.15)	(2.95-3.28)	(1.399 - 1.529)	(4.408 - 4.685)
Mr X	4.02	2.71	9.10	9.89	6.08	5.24	1.274	2.738
	(3.62-4.44)	(2.39-3.07)	(8.82-9.39)	(9.60-10.19)	(5.77-6.40)	(4.92-5.58)	(1.150 - 1.412)	(2.506 - 2.992)
SB	3.37	3.40	13.57	16.15	6.89	7.62	3.553	7.020
	(2.94-3.85)	(2.97-3.88)	(11.23-16.18)	(13.63-18.93)	(6.14-7.73)	(6.83-8.49)	(2.826 - 4.460)	(5.919 - 8.308)
TM	0.65	0.61	5.24	6.85	1.86	2.08	0.530	2.749
	(0.61-0.69)	(0.57-0.65)	(4.93-5.56)	(6.50-7.21)	(1.78-1.95)	(1.99-2.16)	(0.477 - 0.589)	(2.601 - 2.906)
Aggregate Runs	0.93	0.84	8.60	9.71	2.88	2.92	0.768	3.003
	(0.89-0.97)	(0.80-0.88)	(8.44-8.77)	(9.53-9.88)	(2.82-2.96)	(2.85-2.99)	(0.737 - 0.799)	(2.921 - 3.088)

Table 3. kidSPAM3- kidSPAM4 Results

Corpus	Ham Misc%		Spam Misc%		Logit av Misc%		(1-ROCA)%	
	kidSPAM3	kidSPAM4	kidSPAM3	kidSPAM4	kidSPAM3	kidSPAM4	kidSPAM3	kidSPAM4
Trec05p	0.82	9.74	12.49	6.57	3.33	8.01	4.167	3.990
	(0.74-0.92)	(9.45-10.03)	(12.20-12.77)	(6.36-6.79)	(3.15-3.51)	(7.84-8.19)	(4.031 - 4.308)	(3.840 - 4.145)
Mr X	3.03	5.31	10.27	2.39	5.64	3.57	2.822	2.326
	(2.69-3.41)	(4.86-5.79)	(9.97-10.57)	(2.24-2.54)	(5.32-5.99)	(3.39-3.77)	(2.612 - 3.049)	(2.122 - 2.548)
SB	2.86	5.75	24.42	18.09	8.89	10.40	6.360	8.042
	(2.46-3.30)	(5.18-6.35)	(21.43-27.60)	(15.44-20.98)	(8.03-9.83)	(9.46-11.43)	(5.247 - 7.690)	(6.843 - 9.430)
TM	0.51	0.91	8.58	5.88	2.15	2.34	2.653	2.473
	(0.48-0.55)	(0.86-0.96)	(8.19-8.98)	(5.55-6.22)	(2.06-2.24)	(2.25-2.43)	(2.482 - 2.836)	(2.320 - 2.636)
Aggregate Runs	0.75	2.94	11.11	5.05	2.99	3.86	2.741	2.606
	(0.72-0.79)	(2.87-3.02)	(10.93-11.29)	(4.92-5.18)	(2.91-3.07)	(3.79-3.93)	(2.672 - 2.812)	(2.537 - 2.676)

Acknowledgements

This work was supported by National Natural Science Foundation of China, Beijing, China (Grant No. 60475007), Key Project of Foundation of Ministry of Education of China (Grant No. 02029), and Cross - Century Talents Foundation.

REFERENCES

1. P. Domingos, and M. Pazzani, “On the optimality of the simple Bayesian classifier under zero-one loss,” Machine Learning, pp. 103-130, 1997.
2. P. Domingos, “A unified bias-variance decomposition for zero-one and squared loss,” in Proc. of the 17th National Conference on Artificial Intelligence, AAAI Press, 2000.
3. L. Breiman, “Bias, variance and arcing classifiers (Technical Report 460) ,” Statistics Department, University of California at Berkeley, Berkeley, CA. ftp://ftp.stat.berkeley.edu/users/breiman/arcall.ps.Z.
4. H. Zhang, “The optimality of Naive Bayes,” in Proc. of the Seventeenth International Florida Artificial Intelligence Research Society Conference, 2004
5. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian approach to filtering junk e-mail,” in AAAI’98 Wkshp. Learning for Text Categorization, Madison, WI, July 27, 1998.
6. J.R. Bellegarda, D. Naik, K.E.A Silverman, “Automatic junk e-mail filtering based on latent content,” in ASRU '03 Wkshp. Automatic Speech Recognition and Understanding, 2003

7. G. Cormack and T. Lynam, “A study of supervised spam detection applied to eight months of personal email,” http://plg.uwaterloo.ca/~gvcormac/spamcormack.htm
8. D, Geer, “Will new standards help curb spam?” Computer, pp. 14-16, 2004.
9. P. Graham, “A plan for spam,” http://www.paulgraham.com/spam.html
10. P. Graham, “Better Baysian filtering,” In Proc. of Spam Conference, http://spamconference.org/proceedings2003.html
11. Yirong Shen and Jing Jiang, “Improving the preformance of Naive Bayes for text classification,” http://nlp.stanford.edu/courses/cs224n/2003/fp/yirong99/report.pdf
12. J. R. Jeffrey, “Fighting spam on multiple fronts,” http://www.spam.com.cn/ppt/Yahoo!.ppt
13. W. S. Yerazunis, S. Chhabra, C. Siefkes, F. Assis, and D. Gunopulos, “A unified model of spam filtration,” http://crm114.sourceforge.net/UnifiedFilters.pdf
14. Hong Kong Anti-Spam Coalition, “Legislation: One of the key pillars in the fight against spam,” http://www.hkispa.org.hk/spam/20040113-coalition-paper.pdf
15. S. Chhabra, Y. William, S. Christian, “Spam filtering using a Markov random field model with variable weighting schemas,” Proc. of Fourth IEEE International Conference on Data Mining, ICDM 2004, 2004
16. H. Drucker, Donghui Wu, and V. N. Vapnik, “Support vector machines for spam categorization,” IEEE Trans. Neural Networks, vol. 10, pp. 1048-1054,1999
17. T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” Machine Learning: ECML-98, Tenth European Conference on Machine Learning, 1998
18. L. Pelletier, J. Almhana, V. Choulakian, “Adaptive filtering of SPAM,” Proc. of the Second Annual Conference on Communication Networks and Services Research (CNSR’04), 2004
19. Chih-Chin Lai, Ming-Chi Tsai, “An empirical performance comparison of machine learning methods for spam e-mail categorization,” Proc. of the Fourth International Conference on Hybrid Intelligent Systems (HIS’04), 2004
20. Shang-Cai Ma, Hong-Bo Shi, “Tree-augmented Naive Bayes ensembles,” Proc. of 2004 International Conference on Machine Learning and Cybernetics, 2004
21. I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, George Paliouras, and C.D. Spyropoulos, “An evaluation of Naive Bayesian anti-Spam filtering,” Proc. of Workshop on Machine Learning in the New Information Age, Spain, 2000.
22. Yong Wang, Julia Hodges, Bo Tang, “Classification of web documents using a Naive Bayes method,” Proc. of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’03), 2003
23. Karl-Michael Schneider, “Techniques for improving the performance of Naive Bayes for text classification,” Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005), LNCS 3406, pp. 682-693, 2005.
24. Gordon Cormack and Thomas Lynam, “TREC 2005 spam track overview,” http://plg.uwaterloo.ca/~gvcormac/trecspamtrack05/trecspam05paper.pdf
25．TREC 2005 Conference Notebook Appendix - Spam Track，http://plg.uwaterloo.ca/~gvcormac/trecspamtrack05/trecspam05appendix.pdf
26. 1. E. U. Paul, C. B. Neil and A. C. Jeffery, “Decision Tree Induction Based on Efficient Tree Restructuring,” Machine Learning, pp. 1-42, 1997
27. J. R. Quinlan, “C4.5: Programs for machine learning,” San Mateo, CA: Morgan Kaufmann.