An Approach to Spam Detection by Naive Bayes Ensemble Based on Decision Induction

Zhen Yang, Xiangfei Nie, Weiran Xu, and Jun Guo
PRIS Lab, School of Information Engineering,
Beijing University of Posts and Telecommunications, 100876 Beijing, China
yangzhen@pris.edu.cn

Abstract

Spam has been a serious problem to global email users. In this paper, a two-layered spam detection flow was used, which showed the trade-off between accuracy and efficiency. Then we discussed Naive Bayes classifiers ensemble based on Bagging. By casting spam detection in a decision theoretic framework, a Naive Bayes Bagging spam detection model based on embedded decision tree is proposed. Then this model was reduced by strict likelihood score bound limitation of the Naive Bayes classifiers. Finally, an improved method based on classifier error weighted is presented. The experiment results show that the modification is effective.

1. Introduction

Spam, also called unsolicited bulk and unsolicited commercial email, have become a global threat against email users [6]. Spammers continue to devise more aggressive and elaborate techniques. Correspondingly, varied solutions arise. Traditional techniques against spam mainly are rule-based methods, including email header/subject/address analysis and tracking, keyword/ keyphrase matching and filtering [5]. On the other hand, statistical filtering based on content tends to automatically reject e-mail that is classified as spam relying on user's interests. The states of art include rule based learning, Bayesian methods, decision trees, support vector machines, text compression models and combinations of different classifiers [3]~ [7] [15].

There is no unique solution to spam detection, which a multi-faceted approach to fighting spam is necessary. Rule-based method is fast and simple but not very accurate, while statistical filtering based on content is more valid but time-consuming. In this paper, a two-layered spam detection model was used, which showed the trade-off between accuracy and efficiency. Then we discussed Naive Bayes classifiers ensemble based on Bagging for its desirable properties, which can transform good predictors into nearly optimal ones in classification task [10]. Though it is widely known that combining multiple classification models usually provides superior performances compared to using a single, well-tuned model. However, the ways of combining multiple classifiers still are the important factor affecting the aggregated predictor performance. In this paper, by casting spam detection in a decision theoretic framework, a Naive Bayes Bagging spam detection model based on embedded decision tree is proposed. Then this model was reduced by strict likelihood score bound limitation of the Naive Bayes classifiers. Finally, the model was further improved by classifier error weighted based on mutual information.

2. Two-Layered Spam Detection Flow

As discussed above, rule-based method is fast and simple but not very accurate, while statistical filtering based on content is more valid but time-consuming. For some real time massive data processing, accurate and efficiency are both important. So in this work, a two-layered spam detection model was used, which showed the trade-off between accuracy and efficiency. Firstly, mail header was subtracted from incoming datagram. If the mail can be classified according to the email header by rules, there is no need to decode the email body and processing time is saved. Otherwise, the email would be classified based on content. By using different rule sets, the trade-off between accuracy and efficiency can be arrived.

2.1. Rule-based Spam Detection

Rule-based spam detection is very important for real time massive data processing, which can reduce the lo ad of later content-based filtering. But unsuitably select ed rules sets lead to the accuracy degradation dramatically. In our work, four rule sets were used, including white list, porn keyword, drug key word, and Chinese key word. The rule sets was set up by using SpamAssassin[1] as reference. With using regular expression, the rule-based header analysis and matching become flexible, convenient, and robust.

System processing velocity can be effectively accelerated by rule-based spam detection. But the key technology is content-based spam filtering, which determines the performance of system. Among existing techniques, Bayesian methods [9] and their modifications [11] [12] [14] are particularly attractive, because they more formally model the relationship between the content of the email and the reading favors of the user. In next section, we discuss the Naive Bayes spam detection model.

2.2. Naive Bayes Spam Detection Model

The Bayesian classifier is a probability based approach, which is often applied to text categorizations tasks [7] [8] [14]. For spam detection, suppose each email instance M is described by a conjunction of word attribute values $< w_1, w_2, ..., w_n >$. And L is the number of target classes ($C_i, i = 1, ..., L$). The basic concept of Bayesian classifier is to find whether an e-mail is spam or not by looking at which words are found and which words are absent from the message [11] [12]. In the literature, the Bayesian approach to the new email is to assign the most probable target label:

$$C_{NB} = \arg\max_{i \in L} P(C_i) P(w_1, w_2, ..., w_n \mid C_i) . \quad (1)$$

To makes the estimation of parameters tractable, the Naive Bayes assumption is used, which suppose that the attribute values are conditionally independently, then:

$$C_{NB} = \arg\max_{i \in L} P(C_i) \prod_k P(w_k \mid C_i) . \quad (2)$$

For the situation of spam detection, basic attribute units $< w_1, w_2, ..., w_n >$ are the words in one email message (For Chinese corpus, word segmentation is needed), where L is the number of target classes C_i (e.g. C_+ spam/C_- ham). 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-variants Bernoulli model, Poisson Naive Bayes model, and the multinomial model [18]. The difference between these models is the ways of calculating $P(w_k|C_i)$. In this work, $P(w_k|C_i)$ is calculated using multinomial model for its superior performance [11] [12] [18]. In practice, the score of an input e-mail M calculated as follow, and the logarithm formula of (2) is used:

$$\begin{aligned} \text{score}(M) = & \log P(C_+) + \sum_k \log P(w_k \mid C_+) - \\ & (\log P(C_-) + \sum_k \log P(w_k \mid C_-)). \end{aligned} \quad (3)$$

Therefore, if score(M) > 0, the email will be assigned to C+, and C- otherwise. The Naive Bayes filtering model is shown in Fig. 1 and it can work on the supervised model [3] [6]. Supervised model is an online feedback model. In supervised model, 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 [6].

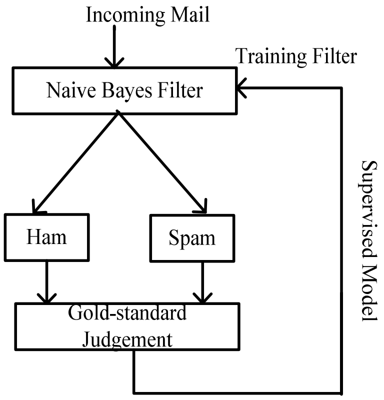

Fig. 1. Spam Detection Filter Model

2.3 Feature Representation Pre-Processing

For text classifying, the feature is a word or word union and following pre-processing measures are often considered: word stemming, stop word elimination, feature selection. But the performance of these measures varied from application to application. And it would be far better if the learning machine itself either made the pre-processing or used all the features. Furthermore, Bayesian method is not sensitive to text pre-processing [3] [5]. Based on the literature, in this paper, all the features were used rather than a subset. But our filter does not use HTML tags as tokens. Such tags and other information such as images, links and attachment are simply eliminated.

3. Naive Bayes Bagging Aggregate

[1] http://spamassassin.apache.org

From above discussion, Naive Bayes classifier is simplicity, low time and memory requirements. It is known that Naive Bayes show every good performance under zero-one loss [9] [10]. But the simple Bayesian classifier is limited in expressiveness in that it can only create linear frontiers. In this section, the improvement of Naive Bayes classifier is discussed. A natural improvement of Naive Bayes is Bagging ensemble [10].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 good predictors into nearly optimal ones [10].

Suppose we are given a sequence of learning email corpus $\{L_k\}$ each consisting of n independent observations data $\{(M_n^k, C_n^k), n = 1,..., N\}$) from the same email data set, where C is the class labels and *M* is the email. Using these learning email data sets $\{L_k\}$, the classifier $S(M, L_k)$ can be given. The aggregating predictor $S_{Bagging}$ shows as follow:

$$S_{Bagging} = Combination(S(M, L_k)) \qquad (4)$$

It denotes that the predictor $S_{Bagging}$ is the combination of this single classifier $S(M, L_k)$. The very natural method of classifiers combination is simple vote and it work well in most application. But the simple vote didn't take the correlation of each classifier in consideration [16] [17]. In this section, we discuss the methods of combining multiple predictors.

3.1. Naive Bayes Bagging Based on Embedded Decision Tree

Though it is widely known that combining multiple classification or regression models typically provides superior results compared to using a single, well-tuned model. The ways of combining multiple classifiers are still on studying. Decision tree induction [1] [2] is a highly practical method for generalizing from examples whose class membership is well known. In this work, voting over C4.5 [2] is generated.

Spam detection model based on decision tree aggregate (see Fig. 2) can be naturally induced from the basic model (see Fig. 1). In every run, the binary decisions (e.g. 0 for spam and 1 for ham) made by filter group can assemble by C4.5. Because classification trees are nonlinear, which can build disconnected decision regions. Therefore the modification can achieve better performance though it needs more overload. Unfortunately, the empirical results show that this method does not have the superiority compared to simply voting. This is partly because the number of filters is few and most of them give the same decision for the special e-mail.

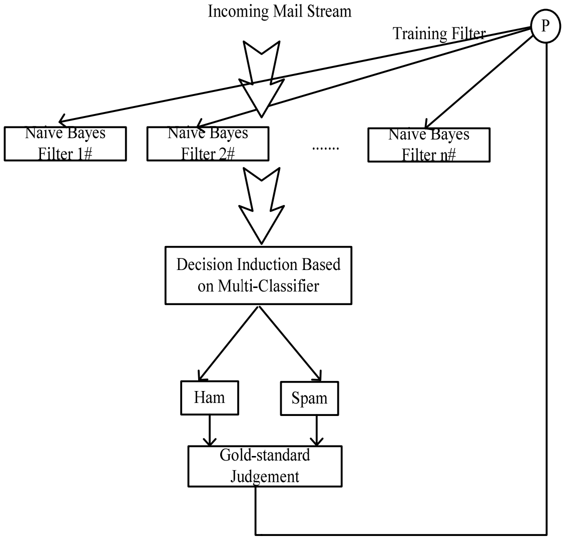

Fig.2. Naive Bayes Bagging Spam Filtering Based on Decision Tree

The performance can be further improved by generating the ensemble predictor from scores made by filter group instead of the binary decisions. Originally, every filter makes binary decision 0/1 for the incoming e-mail *M*, i.e., if score(*M*) > 0, the email will be assigned to spam, and ham otherwise. But now the decision tree is generated using the score(*M*) itself. It means that we are not only combining a number of weak learners to form an ensemble, but also adjusting the classified frontier of the Naive Bayes automatically. Then C4.5 can generate the numerical decision N according to the e-mail sequence adaptively, i.e., if score(*M*) > N, the email will be assigned to spam, and ham otherwise. It is so-called Naive Bayes Bagging based on embedded decision tree method. But there is still no well known solution for tasks with numeric variables that have large value sets in decision tree induction. This method increase the computational complexity because the score(*M*) varied in a larger range. Indeed, the scores that take effect at the decision tree induction are these around zero. Because the larger score show this email is more like a spam, which doesn't impact the classification result. If score can be limited within a minimal bound, the amount of calculation for decision tree generation would be decrease. The general method is score normalization, but the dynamic range of score is unknown. In this paper, if the score is lager than the threshold, it is intercepted by threshold. This method is shown as algorithm I.

Algorithm I: Naive Bayes Bagging Based on embedded Decision Tree for Spam Detection

```
Initial: constant score_threshold
Input: M form incoming mail stream

  If M labeled by rule-based filter
      Output the label
  Else
     Put M to the Naive Bayes filter group
     For i=1 to n
         Let $score_i(M)$ be classification score predicted
            by Naive Bayes filter #i
        If  $score_i(M)$ > score_threshold
          $score_i(M)$ = score_threshold
        End
     End
     Generate decision tree T based on the $score_i(M)$
     Label M by using decision Tree T
  End

Output: the label of the input Email→ham or spam
```

3.1. Error Weighted Naive Bayes Bagging

From above discussion, these mutual information/ information gain ways of combining multiple classifiers are valid and widely used [16] ~ [18]. However, there is some problem with reconstructing the whole tree for every run. If the decision tree can be generated incremental [1], vast overload and time can be reduced. But the synthesis performance of incremental decision tree induction is quite interior compared with original algorithm, and more study is needed.

In this section, we discuss the Bagging-based selective classifiers ensemble method. Selective classifiers ensemble is thought an improved method for Bagging aggregate, in which mutual information weighted method was widely used [16] ~ [18].

Suppose system had already processed n piece of email $\{M_i\}_{i=1}^{n}$, and the correct classification were labeled by an ideal predictor S^o . We are given a sequence of predictors $\{S^i\}_{i=1}^{k}$, which learning from different email corpus $\{L_k\}$ and $S^i(M)$ is the email *M*'s predictor value of classifier S^i. Let $\Psi(S^i, S^o)$ be the mutual information between classifier S^o and S^i . Because the optimal combined predictor should share the most information with the ideal classifier S^o , the aggregating predictor (4) can be rewritten as follow:

$$S_{Error} = \sum_i \Psi(S^i, S^o) \cdot S^i(M) \ . \qquad (5)$$

Considered the specificity of our spam detection model (it's an online feedback system), some simple index which can quantify the statistical information shared among predictors can be used. By using *MMR* (Mail Misclassification Rate) was used, then:

$$S_{Error} = \sum_i I(MMR^i) \cdot S^i(M) \, . \qquad (7)$$

Where MMR^i is the real-time misclassification of classifier S^i , and *I* is the weighted function. In this work, the indicator function *I* is used:

$$I(MMR^i) = \begin{cases} 1, \ if \ \ MMR^i \in \min k(MMR^i), i = 1,...,L \\ 0, \ else \end{cases} \qquad (8)$$

Where min*k* denote the first *k* minimum *MMR* in $\{MMR^i\}_{i=1}^{L}$. In this condition, this method equal to choose the first *k* classifiers with lower *MMR*. Then this method is shown as algorithm II:

Algorithm II: Error Weighted Naive Bayes Bagging for Spam Detection

```
Initial: constant score_threshold=0
Input: M form incoming mail stream

  If M labeled by rule-based filter
     Output the label
  Else
     Put M to the Naive Bayes filter group
     For i=1 to n
         Let $score_i(M)$ be classification score predicted
             by Naive Bayes filter #i
        Calculate the  $MMR^i$  for each filter #i
     End

     If $\sum_{i=1}^{n} I(MMR^i) \cdot \mathrm{score}_i(M) > score\_threshold$
        Label M as Spam
     Else
        Label M as Ham
     End
  End

Output: the label of the input Email→ham or spam
```

4. Experimental Results

To test the performance of proposed methods, the automated test jig will run the target filter against the email sequence providing by NIST[2]. Evaluation will be based on following measures:

[2] http://plg.uwaterloo.ca/~gvcormac/spam/

Table I. The Experiment Results

		HMR(%)	*SMR*(%)	*MMR*(%)	*1-ROCA*(%)
C1	Naive	1.83 (1.33-2.44)	0.21 (0.01-1.16)	1.56 (1.14-2.08)	0.145505 (0.071776-0.294747)
	Simple Vote	1.58↘ (1.12-2.16)	0.21 -- (0.01-1.16)	1.35 ↘ (0.96-1.84)	-------
	Bagging C4.5	1.58↘ (1.12-2.16)	0.21 -- (0.01-1.16)	1.35 ↘ (0.96-1.84)	-------
	Embedded C4.5	1.49↘ (1.05-2.06)	0.21 -- (0.01-1.16)	1.28 ↘ (0.90-1.76)	-------
	Error Weighted	1.45↘ (1.01-2.01)	0.42 ↗ (0.05-1.50)	1.28 -- (0.90-1.76)	-------
C2	Naive	0.80 (0.55-1.12)	12.37 (10.91-13.94)	4.41 (3.90-4.96)	1.59932 (1.21684 - 2.09945)
	Simple vote	0.94 ↗ (0.67-1.28)	11.69 ↘ (10.27-13.23)	4.29 ↘ (3.80-4.84)	-------
	Bagging C4.5	4.70 ↗ (4.08-5.39)	3.45 ↘ (2.67-4.37)	4.31 ↘ (3.81-4.85)	-------
	Embedded C4.5	3.54↗ (3.00-4.15)	3.51 ↘ (2.72-4.44)	3.53 ↘ (3.08-4.03)	-------
	Error Weighted	1.01 ↗ (0.73-1.37)	12.00 ↘ (10.57-13.56)	4.44 ↗ (3.94-4.99)	-------

1) *HMR*: Ham misclassification rate, the fraction of ham messages labeled as spam.
2) *SMR*: Spam misclassification rate, the fraction of spam messages labeled as ham.
3) *MMR*: Mail misclassification rate, the fraction of mail messages is misclassified.
4) *1-ROCA*: Area above the Receiver Operating Characteristic (ROC) curve.

We use two email datasets. The Mail corpus 1 (C1) was Ling-spam: A mixture of 2888 messages (479 spam and 2409 ham) sent via the Linguist list [8]. The SpamAssassin[3] (C2) dataset provided by NIST spam track and consists of 6034 messages (1885 spam and 4149 ham). The purpose is to examine the impact of modifications that the filter is work with no memory files left over from a previous run. And simple voting with 7 filters is used to form the aggregate result. Each corpus was compared the performance with Naive Bayes, Naive Bayes Bagging aggregate with simple vote, Naive Bayes Bagging based on C4.5 using binary decision, embedded C4.5 Naive Bayes Bagging and error weighted Naive Bayes Bagging. Empirical results are shown in Table I. It can therefore be seen that the modified methods are able to improve the performance of Naive Bayes for spam detection significantly. The performance of error weighted-based method varied in different test corpora. This probably is generated by the unstable estimation for the *MMR* from the real-time *MMR*. And it is also noticeable that the input sequence of emails has great influences on the classifying properties.

5. Discussions and Further Work

Technology can help to reduce the volumes of spam 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. In this work, by casting spam detection in decision theoretic framework, some improvements are proposed. The experiment results show that the modifications are effective. But there is still no well known solution for more valid incremental decision tree induction algorithm. Here needs be paid attention to some new algorithms such as C5.0 and ITI [1]. Future work may be directed towards developing better method for mutual information calculation [13] [16] [17] instead of using the whole decision tree framework. The performance of error weighted-based method varied in different test corpora and different training email sequence. In TREC 2006 SPAM track, the active

[3] http://spamassassin.apache.org/publiccorpus/

learning task [2] selects a sequence of messages from the first 90% of the corpus as "teach me" examples, which can alleviate the effect of training sequence of emails.

6. Acknowledgements

This research is partially supported by NSFC (National Natural Science Foundation of China) under Grant No.60475007, Key Project of Chinese Ministry of Education under Grant No.02029 and the Foundation of Chinese Ministry of Education for Century Spanning Talent.

7. References

[1] E. U. Paul, C. B. Neil, and A. C. Jeffery, "Decision Tree Induction Based on Efficient Tree Restructuring," *Machine Learning*, Vol. 29, 1997, pp. 5-44

[2] J. R. Quinlan, *C4.5: Programs for machine learning*, San Mateo, CA: Morgan Kaufmann, 1993

[3] A. Bratko, B. Filipic, "Spam Filtering Using Character-Level Markov Models: Experiments for the TREC 2005 Spam Track," *Proc. of the Fourteenth Text REtrieval Conference*, 2005

[4] E. Frank, C. Chui, I. Witten, "Text Categorization Using Compression Models", *Proc. of IEEE Data Compression Conference*, 2000, pp. 200-209

[5] W. Yerazunis, S. Chhabra, C. Siefkes, F. Assis, D. Gunopulos, "A Unified Model of Spam Filtration", *MIT Spam Conference*, 2005

[6] 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*, 2004

[7] C. Lai, M. 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

[8] I. Androutsopoulos, J. Koutsias, K Chandrinos, G. Paliouras, C. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering", *Proc. of the Workshop on Machine Learning in the New Information Age*, 2000, pp. 9-17

[9] P. Domingos, M. Pazzani, "On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss", *Machine Learning,* 1997, pp. 103-130

[10] L. Breiman, "Bagging Predictors", *Machine Learning*, 1996, pp. 123-140

[11] H. Zhang, "The Optimality of Naive Bayes", *Proc. of the 7th International Florida Artificial Intelligence Research Society Conference*, 2004, pp. 562-567

[12] S. Ma, H. Shi, "Tree-Augmented Naive Bayes Ensembles", *Proc. of 2004 International Conference on Machine Learning and Cybernetics*, 2004, 1497-1502

[13] V. Zorkadis, D.A. Karras, M. Panayotou, "Efficient Information Theoretic Strategies for Classifier Combination, Feature Extraction and Performance Evaluation in Improving False Positives and False Negatives for Spam E-mail Filtering", *Neural Networks*, Vol. 18, 2005, pp. 799-807

[14] Y. Wang, J. Hodges, B. Tang, "Classification of Web Documents Using a Naive Bayes Method", *Proc. of the International Conference on Tools with Artificial Intelligence*, 2003, pp. 560–564,.

[15] H. Drucker, D. H.Wu, V. N. Vapnik, "Support Vector Machines for Spam Categorization", *IEEE Trans. on Neural Networks*, vol. 10, no. 5, 1999, pp. 1048-1054

[16] H. Peng, F. Long, C. Ding, "Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, 2005, pp. 1226-1237

[17] A. Strehl, J. Ghosh, "Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions", *Journal on Machine Learning Research (JMLR)*, 2002, pp.583-617

[18] A. McCallum, K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI'98 Workshop on Learning for Text Categorization*, 1998