Interactive Spam Filtering with Active Learning and Feature Selection

Masayuki Okabe
Toyohashi University of Technology
Tenpaku 1-1, Toyohashi, Aichi, Japan
okabe@imc.tut.ac.jp

Seiji Yamada
National Institute of Infomatics
Chiyoda, Tokyo, Japan
seiji@nii.ac.jp

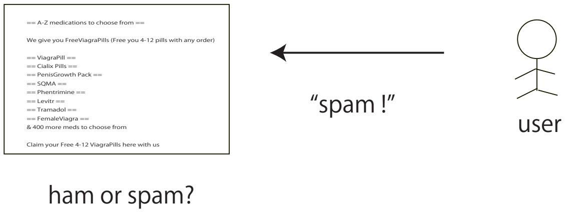
Abstract

This paper proposes an interactive spam filtering method that utilizes active learning and feature selection. Selecting effective features are very important in spam filtering because spam mails include so many meaningless words that are slightly different from each other. Thus selecting effective and ineffective features is promising approach.Although traditional feature selection methods have been done based on some amount of labeled training data, this assumption does not hold in interactive spam filtering. We propose a method to selecting effective features through active learning in spam filtering using naive Bayes approach. Experimental results show that our method outperforms traditional methods that operate with no feature selection.

1. Introduction

There is a report that 96.5% of e-mails flowing in the Internet are spams [1]. They not only make traffic jams in the Internet, but also annoy users. Most of recent e-mail clients adopt some form of spam filtering functionality. However they often fails to remove spams at it's starting point because they do not have enough information about "ham" and "spam" mails early in their usage of the system. Although users can give the information explicitly by pushing a button that indicates a spam mail to the system, this is usually frustrating task if there are a lot of spams. Some kind of efficient feedback framework is necessary.

We propose to apply active learning [2] as such framework. In this framework, systems actively select a mail and urge users to indicate that it is "ham" or "spam" as shown in Figure 1. This learning technique tries to reduce the cost of data labeling, i.e. reducing the number of data to label by finding effective data to make filter. Several active learning techniques have been proposed so far. One of the most popular technique is *uncertainty sampling* [3] that selects the most uncertain data as a candidate to be labeled. A classifier is defined by the filtering method used in a spam filtering system, such as naive Bayes or support vector machines. Uncertainty sampling is a type of active learning technique that tries to reduce the hypothesis space to make classifiers. On the other hand, there is an active learning technique that tries to reduce estimated error that is calculated by the distribution of estimated conditional class probability [4]. The results of the experiments in which this error reduction approach is tested on naive Bayes show good performance compared with uncertainty sampling.



== A-Z medications to choose from ==

We give you FreeViagraPills (Free you 4-12 pills with any order)

== ViagraPill ==
== Cialix Pills ==
== PenisGrowth Pack ==
== SQMA ==
== Phentrimine ==
== Levitr ==
== Tramadol ==
== FemaleViagra ==
& 400 more meds to choose from

Claim your Free 4-12 ViagraPills here with us

Figure 1. Spam Filtering Process

The performance of filtering is influenced by not only the amount of training data, but also the feature set. Feature selection is known to be a good approach [5]. In classification learning, like filtering, feature selection methods that use class information are promising. However a certain amount of labeled data is necessary for those methods to work well; this is a serious problem when applying feature selection to active learning. We propose a method to solve this problem. Our method assumes classess of each unlabeled data can be estimated using conditional class probability during active learning. A system can build a pseudo set of labeled data enough to calculate a feature's score of importance. Once we can calculate the score, we need to determine a threshold to decide the amount of necessary features. Our method set the threshold to a point where score of feature decreases the most in a sorted list.

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.336

2. Spam Filtering with Naive Bayes Classifier

2.1. Naive Bayes Classifier

Naive Bayes is a basic classification technique that is simple and easy to implement but shows relatively good performance [6]. Although there are other classifiers that show better performance such as Support Vector Machines, many spam filter implementations are based on naive Bayes approach because of its simplicity.

The classifier calculates the probability for each mail to belong to "ham" or "spam" based on the Bayes theorem. Suppose we want to classify an e-mail $x \in X$ into a class $c \in C$. The conditional probability $P(c|x)$ that a data belongs to a class is written by $P(c|x) = \frac{P(c)P(x|c)}{P(x)}$ in which Bayes theorem is applied.

Naive Bayes classifiers adopts a generative model that assumes data x are produced by first selecting a class c and then generating an instance x given c. Moreover, it assumes the independence of features of a data x. According to the generative approach, the conditional probability $P(c|x,\theta)$ that a data x belongs to a class c is written as follows.

$$P(c|x,\theta) = \frac{P(c|\theta)\Pi_{w_k \in x} P(w_k|c,\theta)}{\sum_{c\in C} P(c|\theta)\Pi_{w_k \in x} P(w_k|c,\theta)} \quad (1)$$

Here w_k is the kth word that appears in x, $\theta = (\theta_c, \theta_{w|c})$ is a set parameters that includes the class prior and the probability of generating word w class c. Using maximum *a posteriori* estimation, each parameter is calculated as follows.

$$\hat{\theta}_c = \frac{1 + \sum_{x\in X} \delta(x,c)}{|C| + |X|} \quad (2)$$

$$\hat{\theta}_{w_k|c} = \frac{1 + \sum_{x\in X} N(w_k,x)\delta(x,c)}{|V| + \sum_{k=1}^{|V|}\sum_{x\in X} N(w_k,x)\delta(x,c)} \quad (3)$$

where V is all kinds of words that are used to estimate parameters. $N(w_k, x)$ is word frequency of a word w_k appears in x. $\delta(x,c)$ is a function that returns 1 if the class of x equals to c and 0 otherwise.

2.2. Naive Bayes with Active Learning

Until now, we have explained a classification technique with an assumption that we have enough training data (i.e. enough mails categorized either "ham" or "spam"). However such assumption is unrealistic. When we start to use a mail client, there are no classified mails. We need to identify spams to the client one by one.

In this interactive process, active learning tries to reduce labeling cost by selecting data that is expected to improve the classification performance most. We adopt a sampling technique for error reduction approach proposed by Roy [4]. This approach estimates the expected error E using a loss function L that calculates the difference between the true distribution of the class probability $P(c|x)$ and the estimated one using a loss function.

$$E = \int_x L(P(c|x), \hat{P}(c|x))P(x) \quad (4)$$

As a loss function, we use the log loss function described below.

$$L = \sum_{c\in C} -P(c|x)\log \hat{P}(c|x) \quad (5)$$

Since true distribution is unknown, we replace $P(c|x)$ to $\hat{P}(c|x)$. As a result, an expected loss when adding a data is calculated by the following formula.

$$E = \frac{1}{|P|}\sum_{x\in P}\sum_{c\in C} -\hat{P}^*(c|x)\log \hat{P}^*(c|x) \quad (6)$$

where $\hat{P}^*(c|x)$ is conditional class probability when adding a data (x,c) selected by each active learning process. This procedure repeats to consider each unlabeled data in the pool as a feedback candidate. For each data, it considers each possible label c for x, and adds each pair (x,c) to the training data set to make temporary new training data by which a temporal new classifier is created. By using this temporal classifier, resulting expected losses are estimated. Each estimated losse $E(x,c)$ are aggregated by multiplying its present conditional class probability $P(c|x)$.

3. Feature Selection through Active Learning

3.1. Feature selection by Information Gain

Feature selection is a technique for improving classification performance by removing unnecessary features. Although there are several measures to select effective features in text classification, we use information gain since it is one of most effective techniques [5]. Information gain (IG) is calculated by the following formula.

$$\begin{aligned} IG(w) &= -\sum_{c\in C} P(c)\log P(c) \quad (7)\\ &\quad +P(w)\sum_{c\in C} P(c|w)\log P(c|w) \\ &\quad +P(\bar{w})\sum_{c\in C} P(c|\bar{w})\log P(c|\bar{w}) \end{aligned}$$

Where $P(c)$ is the probability that data with class c is selected among all training data, $P(c|w)$ is the probability that data with class c is selected among all data in which a word w appears, and $P(c|\bar{w})$ is the probability that data with class c is selected among all data in which a word w does not appear.

When using IG in active learning, there are mainly two problems. The first one is the lack of training data. Since there are few training data in the early stage of active learning, the confidence of the value of IG is low. The second is how to decide the amount of features to use. Although this problem is not limited to the case of active learning, it is more serious because the confidence of IG is low. We explain the solution of these problems.

3.2. Selection of a feature set though active learning process

We summarize again each problem when applying feature selection in active learning and show the solution for each problem. The first problem is how to prepare enough labeled data. The normal feature selection technique needs some amount of labeled training data. However there are small number of labeled data in the early stage of active learning. We propose a solution for the problem by assuming the class of unlabeled data that have some certain confidence of their class labels. we assume the class label of unlabeled data by the probability $P(c|x)$ with large differences between the two classes, i.e. “ham” and “spam” class. Moreover even if there are enough unlabeled data with high probability of conditional class confidence, it is difficult to discriminate the important features if the amount of data in each class is biased to one class. According to the above consideration, we adopt two thresholds to prepare the source data for selecting important features to classify.

1. T_{conf} : threshold for the conditional probability of class confidence of each unlabeled data.

2. T_{bal} : threshold for the balance of the amount of training data in the classes.

Both thresholds are lower limits that must be satisfied. T_{conf} controls the amount of source data for feature selection. T_{bal} is a parameter to prevent from making an unbalanced source data set where data with only one class occupies the set. This is the solution for the first problem. The second problem is how to define the amount of useful features. To cope with this problem, we propose to define the amount of features by finding the largest gap point. Gap points can be found by firstly sorting the score of $IG(w)$s, and secondly checking the difference of the score between rank k and rank $k+1$. If the largest gap point is K, we select features until the Kth feature in the sorted list of IG.

Algorithm 1 summarizes the procedure of active learning accompanied with feature selection.

Algorithm 1 Procedure for active learning with feature selection

```
$V$ : vocabulary set
$w_i$ : $i$th word in $V$
Conduct sample selection and update probabilities
Calculate conditional probability $P(c|x)$ for each $x$
Select $x$ that has the probability more than $T_{conf}$
Make a pseudo training data set $D_{pse}$ that consists of true and pseudo training data.
for $i = 1$ to $|V|$ do
    calculate IG($w_i$) using $D_{pse}$
end for
Sort $V$ in descending order
Find the largest gap point and create new vocabulary set $V^*$
return $V^*$
```

4. Experiments

In the experiments, we used an e-mail collection that is used in the TREC Spam Filtering Track [7]. This data set contains about 90,000 e-mails that are sorted by date. Since this data is too large to simulate active learning, we used first 1000 mails listed in the *index* file in this data set and divide them two data sets. They are labeled **(a) 1-500**, **(b) 501-1000**. As shown in Table 1, their composition differ especially in the pool set. Each data set consists of 500 data elements in which first 100 data are used for the pool of unlabeled training; the remaining 400 are used for testing. The process of active learning starts from no labeled training data, and then add labeled data one by one until all 100 data elements have been processed. E-mails are preprocessed by removing only symbols. Each data are represented as a bag of words.

We compared our method with other two ones.

1. **ELOSS-FS** : This is our proposed method. Parameters T_{conf} and T_{bal} are set to 0.9 and 0.1, respectively.

2. **ELOSS** : naive Bayes based active learning with sample error reduction

3. **UNCERT** : naive Bayes based active learning with uncertainty sampling

The system **ELOSS** is naive Bayes based active learning with sample error reduction approach that is the basis of our proposed method. Our method merge feature selection procedure to this method. This base method uses all features. The system **UNCERT** is also naive Bayes based active learning with uncertainty sampling. This technique selects a data that is the most difficult to classify, namely the difference between class probability.

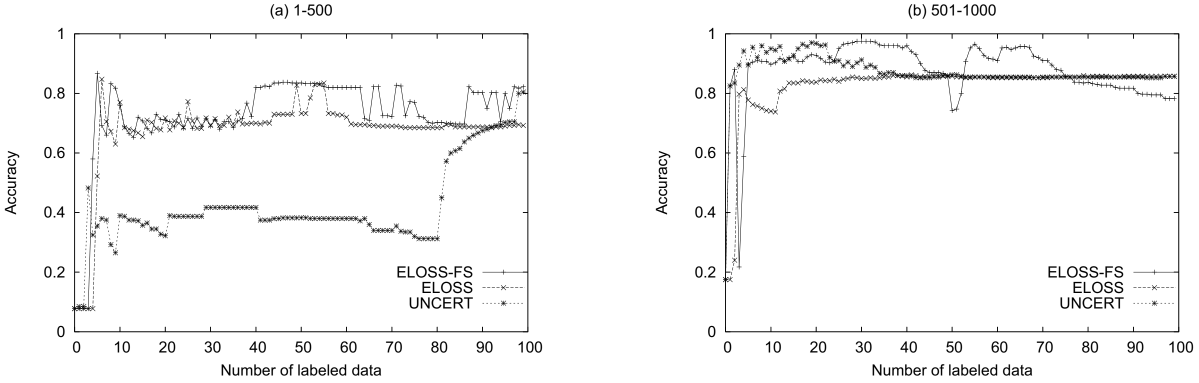


Figure 2. Results of experiments

Table 1. Composition in two data sets

	1-500		501-1000	
	ham	spam	ham	spam
pool	77	23	15	85
test	31	369	70	330

Figure 2 shows the results on two data sets. On all data sets, our method shows better or comparative performance to other two methods. For Data 1-500, **ELOSS-FS** and **ELOSS** are comparative during the early stage (until about 40 data are added). In the following stage, **ELOSS-FS** mostly outperforms **ELOSS**. **UNCERT** shows poor performance in this data set. We observed that **UNCERT** selects mostly ham mails in the early stage of active learning though **ELOSS** and **ELOSS-FS** select ham and spam alternately. The difference between their selection patterns may be the reason of the results in Data 1-500.

Also, for Data 501-1000, **UNCERT** slightly outperforms **ELOSS-FS** until about 20 data are added. However from the point, **ELOSS-FS** gets better, and **UNCERT** decreasing to the same performance of **ELOSS**.

5. Conclusions

In this paper, we propose a feature selection approach with active learning. Our approach iteratively applies information gain based feature selection through active learning. To cope with the lack of labeled training data, we attached pseudo labels estimated by temporal conditional class probabilities. Also, in order to determine the amount of features, we set a threshold by finding the largest gap among information gain scores. Experimental results showed that our proposed method outperformed a conventional approach without feature selection and an approach based on uncertainty sampling.

References

[1] Sophos's report: http://www.sophos.com/pressoffice /news/articles/2008/07/dirtydozjul08.html,2008.

[2] S. Tong, D. Koller, Support vector machine active learning with applications to text classification. In *Proceedings of the Seventeenth International Conference on Machine Learning* , 2001.

[3] D. Lewis, W. Gale, A sequential algorithm for training text classifiers, In *Proceedings of the International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 3-12, 1994.

[4] N. Roy and A. McCallum, Toward Optimal Active Learning through Sampling Estimation of Error Reduction, In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp.441-448, 2001.

[5] Y. Yang and J. O. Pederson, Feature selection in statistical learning of text categorization, In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp.412-420, 1997.

[6] A. McCallum, K. Nigam, A comparisonof event models for naive Bayes text classification, In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[7] G. Cormack and T. Lynam, TREC 2005 Spam Track Overview, In *Proceedings of the Fourteenth Text Retrieval Conference*, 2005.