An Anti-Spam Filter Based on One-Class IB Method in Small Training Sets

Chen Yang[1, 2], Shaofeng Zhao[3], Dan Zhang[4], and Junxia Ma[2]
[1]School of Information, Renmin University of China, China
[2]School of Software Engineering, Zhengzhou University of Light Industry, China
[3]College of Computer and Information Engineering, Henan University of Economics and Law, China
[4]Geophysical Exploration Center, China Earthquake Administration, China

Abstract: *We present an approach to email filtering based on one-class Information Bottleneck (IB) method in small training sets. When themes of emails are changing continually, the available training set which is high-relevant to the current theme will be small. Hence, we further show how to estimate the learning algorithm and how to filter the spam in the small training sets. First, In order to preserve classification accuracy and avoid over-fitting while substantially reducing training set size, we consider the learning framework as the solution of one-class centroid only averaged by highly positive emails, and second, we design a simple binary classification model to filters spam by the comparison of similarity between emails and centroids. Experimental results show that in small training sets our method can significantly improve classification accuracy compared with the currently popular methods, such as: Naive Bayes, AdaBoost and SVM.*

Keywords: *IB method, one-class IB, anti-spam filter, Small training sets.*

Received September 5, 2014; accepted November 25, 2014; Published online December 23, 2015

1. Introduction

The increasing popularity and low cost of electronic mail have intrigued marketers to flood the mailboxes of hundreds of users with unsolicited messages. These messages are usually referred to as spam and may advertise anything from the drug to vacations, so people have to waste much time to view and delete spam messages. As the development of the Internet, Spam messages are more and more harmful to network users. A 2009 study shows that 75% emails which are received by all the Internet users are spam messages. Besides, a report of Kaspersky also indicates that spam messages consume 84.4% of the whole e-mail bandwidth in 2010 and now, the percentage had a slight decrease, such that: A 2012 research from Symantec shows that it had got 67.6% of all emails. All in all, the situation still seems to be unoptimistic and without appropriate counter-measures, spam messages could eventually undermine the usability of email.

Spam filtering technology is also called anti-spam filter [2], as a particular instance of the Text Categorization problem (TC). Some researches consider the email dataset contains multiclass, but every class cannot cover explicit semantic information [19], so we see the problem that only two classes are possible: Spam and legitimate [7]. In recent years, several machine learning algorithms have been applied to anti-spam filter [1, 4]. However, there are some special hypotheses in anti-spam filter being different from the general TC. Firstly, with the development of Internet, thousands of emails came out every minute, the theme of which was always changing continually.

For example, a mail server received many emails about business in some short period, but plenty of emails about drugs might be received in the next short period. The new theme was so different from the old one that the learned filter was weak for upcoming emails. Therefore, users wished that the filter could have a decent performance to filter emails of the new theme, but at the beginning the training emails were very rare. Secondly, spam was easy to be obtained because of its publicity and accessibility, but legitimate being generally private letters, which may involve personal privacy and commercial confidentiality are difficult to obtain. Therefore, the dataset may be encrypted or spam samples are larger than legitimate. Previous researches on anti-spam filter consider the number of training samples are very sufficient and also, ignore the rarity of legitimate [7, 16].

The existing classifiers for anti-spam filter mainly are Naive Bayes classifier [2, 16], boosting [7] and Support Vector Machine (SVM) [20]. These algorithms are content-based filters, which associate spam filtering with the binary document classification. Naive Bayes classifier, applied to text, can be considered an improved learning-based variant of keyword filtering. It depends on the so-called naive independence assumption, namely that all the features are statistically independent [18]. Boosting is a general name for the algorithms based on the idea of finding a highly accurate classification rule by combining many weak rules. For filtering spam AdaBoost algorithm was proposed by Carreras [8]. Another classifier proposed for spam filtering is SVM [9]. Given the training

samples and a predefined transformation, which maps the features to a transformed feature space, the classifier separates the emails of the two classes with a hyper plane in the transformed feature space [9]. Although, the above algorithms have a good performance over abundant training samples, they never consider the insufficient of training samples might weaken the performance at the beginning of spam filtering.

To address this issue, we propose an Anti-Spam Filter model based on Binary TC (SFBTC) and the corresponding algorithm as Anti-Spam Filter algorithm based on One-Class Information Bottleneck (SFOC-IB). The Information Bottleneck (IB) method is proposed for dimensionality reduction by constructing some compact representations of given data set [22]. The IB method copes with the difficulty of specifying an "appropriate" distortion measure in the rate distortion approach by defining a relative variable with respect to the original data X [15]. The one-class IB model proposed for retrieving a small set of relevant samples similar to a few seed samples [12]. When the training set is small, the set retrieved by one-class IB is highly relevant to the original training set.

In this paper, we consider anti-spam filter as a binary text classification problem, in which the training sets are small. Hence, to avoid over-fitting SFOC-IB algorithm tries to construct centroids to protect the uppermost feature of training sets. We optimize one-class IB objective function by Blahut-Arimoto (BA) method [13] to extract highly positive emails for the construction of centroids. In SFBTC, it is not necessary for users to be concern on the number of categories of spam, but only focus on whether spam is filtered or not. Therefore, we consider anti-spam filter as a binary classification problem. Lastly, we measure similarity between testing samples and centroids in a certain distance, such as JS divergence [14]. Due to less training samples, classification methods depending on learning hyper plane, such as SVM, are poor generalization ability in anti-spam filter, which is confirmed in the experiment chapter. The main contributions in this paper can be summarized as follows:

1. We use BA method for optimizing one-class IB objective function and propose SFOC-IB algorithm to learn spam and legitimate centroids in small training sets, respectively.
2. To filter spam, we design a binary classification model referred to as SFBTC and use JS divergence for measuring the similarity between testing emails and centroids.

The rest of the paper can be organized as follows: In section 2, we describe the background. In section 3, we introduce SFBTC model and SFOC-IB algorithm. In section 4, we present experiment results and compare the performance in abundant training emails and with training emails reducing, respectively. Finally, in section 5, we summarize this paper with discussions on some further works.

2. Background

In this section, we will recap the IB method and one-class IB. Throughout this section, we use the following notations: capital letters (X, Y, T, …) denote the random variables; lowercase letters (x, y, t, …) denote the corresponding realizations; the notation $p(x)$ denotes $p(X=x)$, namely the probability of the random variable X taking the value x; and $|X|$, $|Y|$, $|T|$ denote the cardinality of X, Y, T respectively.

2.1. The IB Method

The IB method [22] originated from Shannon's rate distortion approach. Consider an encoding scheme for a random variable X that follows a distribution $p(X)$. The encoding scheme involves representing the random variable X by a compressed variable T. Given a distortion function $d(x, t)$, we want to encode the values of X such that the average error is less than a given distortion threshold D. Shannon's rate distortion theorem states that the infimum of all rates under a given constraint on the average distortion D is given by the following function:

$$R(D) \equiv \min_{\{p(t|x):E(d(x,t))\le D\}} I(T;X) \quad (1)$$

Where the $I(X; T)$: Is the mutual information between X and T, and the function $E(d(x,t)) = \sum_{x,t} p(x,t)d(x,t)$ is the expected distortion induced by $p(t|x)$.

Unlike the rate distortion approach, the IB method avoids the arbitrary choice of the distortion function. The motivation comes from the fact that in many cases, defining the "target" variable Y with respect to X is a much easier task than defining a distortion function. Given the joint probability distribution $p(X, Y)$ on variables X and Y, the IB method tries to define a compact representation T with respect to X, which could minimize the compression-information $I(X; T)$, while maximize the relevant-information $I(X; T)$. In a sense, the compact representation T implements a "bottleneck" for the dependency between X and Y is given by the following function:

$$F(p(t \mid x)) = \min_{p(t|x),p(t),p(y|t)} I(T;X) - I(T;Y) \quad (2)$$

Where β: Is a Lagrange multiplier. As $\beta \to 0$, the compression is optimal; on the other extreme, as $\beta \to \infty$ the compression is minimum.

2.2. The One-Class IB

The one-class IB model [12], which is used to seek a rule to find a coherent subset of instances similar to a few positive examples in a large pool of instances, is proposed by Crammer. Given a set of instances

indexed by the integer random variable X and each instance is described by a point x. In particular, the learning task is to find a centroid w in the space where there are many seed instances x close to it.

For formalizing the task as a source coding problem, an instance x is either coded as the one-class, with distortion $D(x||w)$ and assigned the code 0, or coded as itself with zero distortion. The random variable T represents the code for an instance: If $T=0$, the instance was coded with the one-class, while if $T=x>0$, the instance is coded as itself. The coding process is summarized by the conditional probability $p(t|x)$ of encoding x as t. If $t \notin \{x,0\}$, $p(t|x)=0$; $t=x$, $p(t|x)=1$; $t=0$, $0<p(t|x)<1$. We use IB method to describe the above process by the following function:

$$F(p(t|x)) = I(T;X) + D(w,\{p(0|x)\}) \quad (3)$$

For one-class learning, the distortion term measures how well on average the centroid w serves as a proxy to each of the instances x:

$$D(w,\{p(0|x)\}) = \sum_x p(x)p(0|x)D(x\|w)$$

In contrast with standard rate distortion and IB formulations, the average distortion is computed only for $T=0$, because the distortion is zero for $T>0$.

3. Spam Filtering Based on One-Class IB

In this section, we formally define the problem as follows. Let M_S be the feature matrix of labeled spam using training, M_L be the feature matrix of labeled legitimate using training and M_T be the feature matrix of unlabeled emails to be predicted, where every row vector represents an email. As we have discussed that the number of training data is much less than the number of testing data, $|M_S|+|M_T|<<|M_T|$. We denote M as the matrix of all the labeled and unlabeled emails, so that $M=M_S \cup M_L \cup M_T$. The feature word set of M is denoted by Y. For each email $x \in M_T$, there is a class-label set $C=\{R_S, R_L\}$, which makes $x \in R_S$ or $x \in R_L$. Our objective is to estimate a hypothesis h: $M_T \rightarrow C$ which predicts the labels of spam/legitimate in M_T as accurately as possible.

3.1. The Anti-Spam Filter Algorithm Based on One-Class IB

In this paper, the M_S is identical-distribution to the spam in M_T, but M_T contains more hidden pattern information. If we learn these information from M_S as much as possible, the result would be easy over-fitting. Hence, it is necessary to learn the most relevant information instead of the whole information. SFOC-IB algorithm is used for learning spam and legitimate pattern information respectively, so the pseudo code of SFOC-IB is only provided to learn spam pattern.

Denote $t \in T_S=\{C_o, x\}$ as a class set, where C_o represents the one-class set and x represents the non-one-class email, and w_S as the spam centroid. The value of w_S is solved by highly positive emails instead of the whole emails. Assume if an email x in M_S is a highly positive email, we classify it into the one-class as $x \in C_o$. Through strengthening the constraint, we should remove low positive samples from C_o. Hence, the centroid w_S averaged by highly positive samples is the better representation of training samples. The conditional distribution, which is formulated as:

$$p(t|x) = \begin{cases} p(C_o|x), & t = C_o \\ 1-p(C_o|x), & t = x \end{cases} \quad (4)$$

Is the probability of assigning x to t. First of all, the mutual information $I(T; X)$ is expanded as the following form:

$$I(T;X) = \sum_{x,t} p(x)p(t|x)\log\left(\frac{p(t|x)}{p(t)}\right)$$

Secondly, we rewrite the mutual information term by Equation 4:

$$I(T;X) = \sum_x p(x)\left[p(C_o|x)\log\frac{p(C_o|x)}{p(C_o)} + (1-p(C_o|x))\log\left(\frac{p(x|x)}{q(x)}\right)\right]$$

Where $q(x)=p(x|x)p(x)$ is the probability that x appears as the non-one-class email, and bring $q(x)$ into $I(T;X)$ as:

$$I(T;X) = \sum_x p(x)\left[p(C_o|x)\log\left(\frac{p(C_o|x)}{p(C_o)}\right) + (1-p(C_o|x))\log\frac{1}{p(x)}\right]$$

Lastly, we use the above equation for rewriting Equation 3:

$$\begin{aligned} F(p(t|x)) = {} & \sum_x p(x)[p(C_o|x)\log\frac{p(x)p(C_o|x)}{p(C_o)} - \log(p(x))] \\ & + \sum_x p(x)p(C_o|x)D(x\|w_S) \end{aligned} \quad (5)$$

Equation 5 is our objective function associated with three sets of $p(C_o|x)$, $p(C_o)$ and w_S. The distribution $p(C_o)$ is the marginal of $p(C_o, x)$ and the marginal is the probability of assigning any email to the one-class C_o. The centroid w_S is the average of all the points x weighted by their probability of membership in the one-class C_o.

$$p(C_o) = \sum_x p(x)p(C_o|x) \quad (6)$$

$$w_S = \sum_x p(x|C_o)x = \frac{\sum_x p(x)p(C_o|x)x}{p(C_o)} \quad (7)$$

Hence, the optimizing of Equation 5 completely depends on the $p(C_o|x)$, so we use KKT conditions to solve $p(C_o|x)$:

$$p(C_o|x) = p(C_o)\frac{e^{-D(x\|w_S)}}{p(x)} \quad (8)$$

Because the solution of Equation 5 has three self-consistent non-convex Equations 6, 7 and 8 we apply BA method to optimize these equations. Nevertheless, BA method is always a convex optimization scheme used for solving the optimum on the convex sets, but

many researches utilized it to gain the satisfying solution [22] on the non-convex sets. Therefore, SFOC-IB algorithm can be described in Algorithm 1.

Algorithm 1: The pseudo code of SFOC-IB.

Input: *Joint distribution matrix* $M_S, \forall x \in M_S$, *Lagrange multiplier* β, *Convergence parameter.*
Output: *A centroid* w_S *of* M_S, *and iterations m.*
Initialization: *Initialize* $p(C_o \mid x) = \frac{1}{|X|}$, *and find the corresponding* $p(C_o)$, w_S *through Equations 6 and 8*

While True

$$p^{(m+1)}(C_o \mid x) = p(C_o)^{(m)} \frac{e^{-D(x \| w_S^{(m)})}}{p(x)}, \forall x \in M_S$$

$$p(C_o) = \sum_x p(x) p(C_o \mid x)^{(m+1)}, \forall x \in M_S$$

$$w_S^{(m+1)} = \frac{\sum x \; p(x) p(C_o \mid x)^{(m+1)} x}{p(C_o)^{(m+1)}}, \forall x \in M_S$$

If

$$| D(w_S^{(m)}, \{p(C_o \mid x)^{(m)}\}) - D(w_S^{(m+1)}, \{p(C_o \mid x)^{(m+1)}\}) | \leq \pounds, \forall x \in M_S$$

Break

.

After initializing $p(C_o|x)$, $p(C_o)$ and w_S, the algorithm fixes $p(C_o)$ and w_S and find their closest point in $p(C_o|x)$ by Equation 8; fix $p(C_o|x)$ and find its closest point in $p(C_o)$ by Equation 6; then fix $p(C_o|x)$ and $p(C_o)$ and find their closest point in w_S by Equation 7. Repeating this process must converge since clearly the distance between w_S and x in C_o deceases with each iteration, and the more positive the email is, the closer to 1 the condition probability $p(C_o|x)$ is. In this paper, $p(C_o|x)$ and $p(C_o)$ are non-convex, so SFOC-IB only can converge to a satisfying local minimum.

Although, many kinds of emails have been mushrooming during the last few years, the text form occupied the dominated advantage in daily communication. Therefore, we use KL divergence [11], widely measuring text discrimination, for describing the difference of emails, defined as:

$$D_{KL}[p_1 \| p_2] = \sum_x p_1(x) log \frac{p_1(x)}{p_2(x)} \quad (9)$$

Where D_{KL} is not symmetric and dissatisfies the triangle inequality. The distance $D(x||w_S)$, that is used for the difference between x and w_S is equal to $D_{KL}[x||w_S]$ in this paper.

3.2. The Anti-Spam Filter Model Based on Binary TC

In Figure 1, we will divide the SFBTC model into three subdivisions including data preprocessing, training process and filtering process. The preprocessing transforms the original email dataset into three feature matrixes M_S, M_L, and M_T, which represent the spam training data, legitimate training data, and testing data, respectively. In training process, SFOC-IB algorithm learns the spam centroid w_S and legitimate centroid w_L after inputting M_S and M_L. The filtering process classifies the preprocessed testing matrix M_T into the spam set R_S and legitimate set R_L.

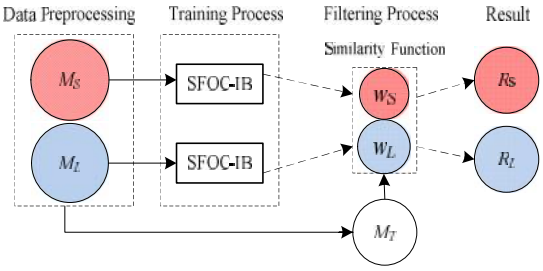


Figure 1. SFBTC model.

The preprocessing transforms the original email dataset containing the training and testing datasets into the feature matrixes M_S, M_L, and M_T, where each value means the joint probability about emails and feature words. For decreasing the interference of the noisy words, we process the original email dataset by the following steps described as:

1. Deleting HTML marks in email such as attachments, head and pictures, but retaining the subject and content.
2. Transforming capital letters into lowercase letters.
3. Uniting all digits into a single "digit" symbol.
4. Ignoring the non-alpha-numeric characters and the single letter, while reserving "$".
5. Using an English lemmatizer, named as Morph Adorner [6], for converting each word into its base form (e.g., "is" becomes "be").
6. Deleting stop-list words [2] and 100 neutral phrases whose frequency are roughly equal in spams and legitimates.
7. Retaining the n highest information gain IG score [3] words as features.

In the dataset, we consider every email is a vector $<y_1, y_2, \ldots, y_n>$, $y \in Y$, where $y_1, y_2, \ldots, y_n$ are the features words in this email. Therefore, we can transform the dataset into a sparse count matrix N, where the matrix value $N(x, y)$ denotes the occurrence times of the word y in the email x and $N(x, y)=0$ if this email does not contain y. After that, we construct the feature matrix $M(X, Y)$, where the matrix value $M(x, y)$ denotes the joint probability $p(x, y)$ measured as follows:

$$M(x, y) = \frac{N(x, y)}{\sum_x \sum_y N(x, y)} \quad (10)$$

Because the spam training data, legitimate training data and testing data are independent to one another, we divide N into three count matrixes N_S, N_L, and N_T, used for the calculation of the joint probability distributions M_S, M_L and M_T, by Equation 10, respectively.

We input the joint probability distributions M_S and M_L into SFOC-IB algorithm and solve the centroids w_S and w_L in the training process. In the filtering process, we measure the similarity between the testing email and the solved centroids by some similarity function f_S

to make sure if one email in M_T is spam. In Algorithm 2, when one email x is more similar to W_S ($f_S(x,w_S)<f_S(x,w_L)$), we will mark it as spam, or mark it as legitimate.

Algorithm 2: The filtering process.

Input: *Centroids* w_S *and* w_L*; Joint distribution matrix* M_T.
Output: *Result* R_S *and* R_L *by filtering* M_T.

While True
If $f_S(x, w_S)<f_S(x, w_L)$, $\forall x \in M_T$.
$x \to R_S$.
Else
$x \to R_L$.
If $M_T = \varnothing$.
Break.

For measuring the similarity between a testing email and centroids, meanwhile ensuring the symmetry of $f_S(x, w)< f_S(w, x)$, we use *JS* divergence [14] defined as:

$$JS(p_1 \| p_2) = ®_1 KL(p_1 \| \bar{p}) + ®_2 KL(p_2 \| \bar{p}) \quad (11)$$

Where $0 < ®_1, ®_2 < 1$, $®_1 + ®_2 = 1$ and $\bar{p} = ®_1 p_1 + ®_2 p_2$. The smaller value of $JS(x||w)$ shows x is more similar to w.

4. Experiment

This section mainly includes three parts about the introduction of data sets, the evaluation method and experiment design.

4.1. Data Sets

In Table 1, there are four datasets in our experiment: Ling-spam [5], spam base [5], PU3 [5], trec07p [10], where the percentage of spam was mushrooming year by year, resulting in the weak proportion of legitimates.

Table 1. The list of datasets.

	The Number of Emails	Spam%
Ling-Spam	2893	17%
Spam base	4601	39%
PU3	4139	44%
Trec07p	75419	67%

These datasets are widely used in many spam-filter experiments, and their forms are completely different. Ling-spam and PU3 were tested by Androutsopoulos and Michelakis [2, 17]. Ling-spam is the mailing list dataset and PU3 from private users' mailboxes is encrypted. Trec07p and spam base were tested by Sculley and Wachman [21]. Trec07p contains original emails from public mail servers and spam proportion is close to the real environment. Spam base from the UCI database had been preprocessed.

Ling-spam and trec07p need the whole data preprocessing steps. The encrypted PU3 hid the detailed semantic information, so we can only process it by step 7. Emails in the preprocessed spam base only contain 58 feature words, so we do nothing with it.

4.2. The Evaluation Method

Following [20] the average recall, precision and F1-measure are also used as our evaluation measures. Assume that we define four variables *A*, *B*, *C*, *D*, as shown in Table 2.

$$Recall = \frac{A}{A + C} \quad (12)$$

Table 2. The evaluation matrix.

	Classified as Spam	Classified as Legitimate
Spam	A	C
Legitimate	B	D

- *Recall*: Measures the fraction of spam correctly predicted by the classifier. Classifiers with large recall have few spams misclassified as legitimates.

$$Precision = \frac{A}{A + B} \quad (13)$$

- *Precision*: Determines the fraction of emails that actually turns out to be spam in the group where the classifier has declared as the spam class. The higher the precision is, the lower the number of false spam errors committed by the classifier will be.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (14)$$

Precision and *recall* can be summarized into another metric known as the *F*1-measure (*F*1). In principle, *F*1 represents a harmonic mean between recall and precision. A high value of *F*1 ensures that both precision and recall are reasonably high. Hence, we depend on *F*1 to evaluate how good the algorithm is.

4.3. Experiment Design and Performance Analysis

In this section, we compare SFOC-IB with Naive Bayes filter, AdaBoost and SVM. Generally speaking, the number of feature words affects the performance of classifiers. The small number cannot express enough information of the original email; likewise, the large number contains too much noise and slows the algorithm speed. Hence, we need to find the appropriate number of feature words by an experiment to make the final result better. In the comparison experiment, we compare SFOC-IB with all benchmark algorithms using abundant training emails, and then, with training emails reducing the trend of *F*1 will be described in detail.

4.3.1. The Appropriate Number of Feature Words

This experiment is done on feature matrixes where feature words were chosen by *IG* score with six different dimensions. In the experiment, we wish to obtain the most appropriate range of the number of feature words, in which the accuracy of SFOC-IB is high and the result is not over-fitting.

The trend of the recall and precision on the ling-

spam corpora is presented in Figure 2, where the x-axis represents the number of feature words stepping by 500 words. We can conclude that, with the increasing of feature words, the trends of the recall and precision are ascending gradually. However, after this number reaches a certain level, continuously increasing feature words could not affect the values of the recall and precision. The reason of this phenomenon is that, irrelevant feature will be imported when feature words surpass a certain threshold, and more words could not sufficiently reflect the characteristic of emails. In our experiment, when the range of the number is from 1500 to 2500, the result is outstanding and over-fitting is restrained effectively. Hence, we will retain the 2000 highest information gain *IG* score words as features in the following experiment.

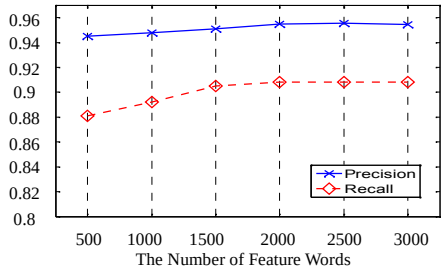


Figure 2. The influence of the number of feature words to SFOC-IB.

4.3.2. Results in Abundant Training Emails

We compare SFOC-IB with Naive Bayes, AdaBoost and SVM under the normal condition, where the majority of emails are used for training and the rest are used for testing. The detailed results are illustrated in Table 3, where "P%", "R%" and "*F*1%" represent the percentage of *precision*, *recall* and *F*1, respectively, and the bold values are the top percentages of some measurement in every dataset. In order to improve SVM algorithm, we choose the C-SVC algorithm and RBF kernel function in libsvm library, gain the optimal parameters by the ten-fold cross-validation and normalize all emails [9].

To reduce random variation, ten-fold cross-validation is employed. That is, each message collection was divided into ten equally large parts, maintaining the original distribution of the two classes in each part. Each experiment was repeated ten times, reserving a different part for testing at each iteration, and using the remaining parts for training. The results were then averaged over the ten iterations, producing more reliable results.

Table 1. Precision, recall and F1 results for all algorithms.

P%/ R%/ F1%	Ling-Spam	Spam Base
SFOC-IB	**98.1**/93.2/95.6	**94.5**/79.3/88.1
SVM	97.2/**95.0**/**96.1**	86.2/**94.8**/**90.3**
Naive Bayes	93.2/82.9/87.7	90.4/82.5/86.3
AdaBoost	94.2/85.2/89.5	81.7/88.1/84.7
P%/R%/F1%	**PU3**	**trec07p**
SFOC-IB	**96.0**/94.2/95.1	92.3/96.8/95.2
SVM	95.1/95.9/95.5	**96.6**/**98.6**/**97.6**
Naive Bayes	95.3/96.7/**96.0**	92.4/96.2/94.3
AdaBoost	89.6/**97.1**/93.2	91.2/98.4/94.7

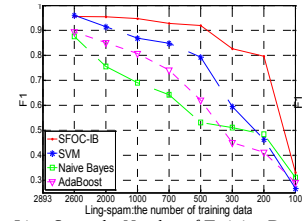
Evidently, the precision of SFOC-IB is more excellent than the benchmark algorithms in Ling-spam, spam base and PU3. In contrast, the recall of SVM is better than others in Ling-spam, spam base and trec07p. Thus, we conclude that SFOC-IB tends to achieve high precision by sacrificing *recall*. In other words, spam recognized by SFOC-IB tends to be purer but more incomplete. This is a good character because users could tolerate that spam came into mailbox occasionally but could not tolerate that legitimate was recognized as spam. Therefore, the next point is how to improve the *precision* under the condition of acceptable recall in future research.

In terms of *F*1, SFOC-IB is inferior to SVM in three datasets, but superior to others. In PU3 SFOC-IB is close to Naive Bayes. Presumably, when SFOC-IB performs a high *precision*, the sacrificed *recall* is low. Because the training emails are abundant, SVM can predict a good hyper plane. In contrast, superfluous emails cannot provide more effective information to the centroid learned by SFOC-IB. It should be the reason that SFOC-IB is slightly inferior. All in all, the performance of SFOC-IB is comparable to the best one in four dataset. In addition, it is important to bear in mind that SVM relies on the normalization. In fact, preliminary results of SVM with no normalization were substantially inferior and normalization took much time on large dataset, such as: Trec07p. In contrast, the SFOC-IB requires no normalization, which is a practically important advantage.

4.3.3. The Trend with Training Emails Reducing

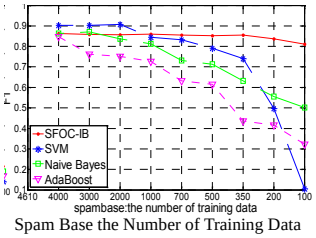
In this experiment, we focus on comprehensive performance of all algorithms in small training sets, and present the trend of *F*1 with training emails reducing. In order to observe the trend, we use many emails for training, and then gradually reduce training emails. First, we train all algorithms by the majority of emails, use the rest for testing and report the *F*1. Second, we reduce training emails, test the performance of all algorithms and report the *F*1. Repeat this experiment until the scale of training sets are 100, except that the scale are 500 in trec07p for it is too large. To gain the reliable results, in every experiment we randomize part of dataset as the training set, and run this procedure 10 times. Thus, the scales of ten training sets are the same as each other, but they contain different emails and then, we run all algorithms in the ten training sets, respectively and average the ten *F*1 as the final result.

In Figure 3, the first scale in the x-axis represents the total number of datasets and the second scale is 90%. In order to emphasize the performance in small training sets, the majority of our experiments are done in the training sets, whose scales are less than 50% of the total. Hence, we set a large decrease of training emails at the beginning and with training emails reducing; the decrease also has been reduced.



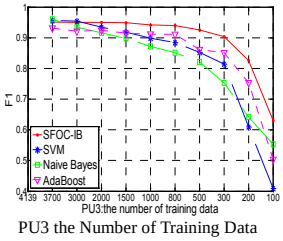
Ling-Spam the Number of Training Data

a) Result of Ling-spam.



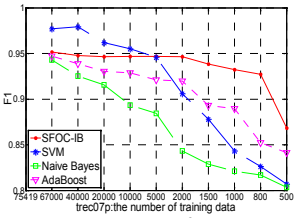
Spam Base the Number of Training Data

b) Result of spam-based.



PU3 the Number of Training Data

c) Result of PU3.



TREC07P: The Number of Training Data

d) Result of tre07p.

Figure 3. The trend of F1 with training emails reducing.

Figure 3 shows that the *F*1 of all algorithms has different degrees of descending gradually with the number of training emails decreasing. The results also make clear that the large scales of training emails are beneficial for the production of a good prediction model. However, when training emails are insufficient, SFOC-IB will be more excellent than the benchmark algorithms. In addition, the less the training emails are, the clearer the advantage is. Especially, although the last scale of x-axis represents 2.16%, 2.41% and 0.66% of spam base, PU3 and trec07p, respectively, SFOC-IB still produces the decent *F*1 against the benchmark algorithms. Presumably, because training emails are insufficient, the feature of training sets cannot contain the comprehensive description of testing emails. Hence, as the benchmark algorithms try to predict the feature of testing emails as much as possible by leaning some rules or hyper planes of training sets, such as: AdaBoost and SVM, the over-fitting has appeared. In contrast, SFOC-IB algorithm only uses highly positive emails for constructing the centroid, so the over-fitting can be avoided. Based on the above, we can draw a conclusion that SFOC-IB can learn a good centroid in the small training sets. Noting that the *F*1 trend of SFOC-IB is more stable than the benchmark algorithms in spam base and trec07p datasets, we conclude that the small training sets can contain the majority of main feature information of testing emails. Therefore, at the beginning of spam filtering SFOC-IB algorithm which only learns the uppermost feature is effective.

5. Conclusions

In this paper, we focus on how to learn a good centroid to filter spam in small training sets and propose SFBTC model and the corresponding algorithm as SFOC-IB. This new algorithm extracts highly positive training data by one-class IB method and constructs centroids that can describe the uppermost feature of original datasets and then, SFBTC model is used for filtering spam by the comparison of similarity between new emails and centroids. The experiments on four email datasets show that the performance of SFOC-IB can be comparable to the popular spam filtering algorithms in abundant training emails and in small training sets, the performance of SFOC-IB is more outstanding and stable than the benchmark algorithms.

Our future research will extend the method to the classification of documents in other domains and consider how to learn multiclass simultaneously by one-class IB method in the field of anti-spam filter.

Acknowledgements

This work is supported by the Science and Technology Foundation of Education Department of Henan Province (No.14A520062), the Science and Technology Foundation of Zhengzhou University of Light Industry (No. 2013XJJ018).

References

[1] Allias N., Noor M., Ismail M., and Silva K., "A Hybrid Gini PSO-SVM Feature Selection Based on Taguchi Method: An Evaluation on Email Filtering," *in Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, Siem Reap, pp. 94-97, 2014.

[2] Androutsopoulos I., Koutsias J., Chandrinos K., George Paliouras G., and Spyropoulos C., "An Evaluation of Naive Bayesian Anti-spam Filtering," available at: http://arxiv.org/pdf/cs/0006013.pdf, last visited 2000.

[3] Androutsopoulos I., "Learning to Filter Unsolicited Commercial E-Mail," *Technical Report*, National Center for Scientific Research, 2004.

[4] Barigou F.,a, Beldjilali B., and Atmani B., "Using Cellular Automata for Improving KNN Based Spam Filtering," *The International Arab Journal of Information Technology*, vol. 11, no. 4, pp. 345-353, 2014.

[5] Blanzieri E. and Bryl A., "A Survey of Learning-based Techniques of Email Spam Filtering," *Artificial Intelligence Review*, vol. 29, no. 1, pp. 63-92, 2008.

[6] Burns R., "MorphAdorner: Morphological Adorner for English Text," 2006.

[7] Carreras X. and Marquez L., "Boosting Trees for Anti-Spam Email Filtering," available at: http://web.cs.ucla.edu/~miodrag/cs259-security/carreras01boosting.pdf, last visited 2001.

[8] Carreras X., Màrquez L., and Padró L., "A Simple Named Entity Extractor using AdaBoost," *in Proceedings of the 7th Conference on Natural Language Learning*, Reykjavik, pp. 152-155, 2003.

[9] Chang C. and Lin C., "LIBSVM: A Library for Support Vector Machines," available at: http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf, last visited 2011.

[10] Cormack G., "TREC 2007 Spam Track Overview," *in Proceedings of the 6th Text REtrieval Conference*, Maryland, USA, pp. 1-16, 2007.

[11] Cover M. and Thomas J., *Thomas. Elements of Information Theory*, Wiley Press, New York, 1991.

[12] Crammer K., Talukdar P., and Pereira F., "A Rate-distortion One-class Model and its Applications to Clustering," *in Proceedings of the 25th International Conference on Machine learning*, Helsinki, pp. 184-191, 2008.

[13] Csisz I and Tusnády G., "Information Geometry and Alternating Minimization Procedures," *Statistics and Decisions*, pp. 205-237, 1984.

[14] El-Yaniv R., Fine S., and Tishby N., "Agnostic Classification of Markovian Sequences," *in Proceedings of the 10th Annual Conference on Neural Information Processing Systems*, pp. 465-471, 1997.

[15] Harremoes P. and Tishby N., "The Information Bottleneck Revisited or How to Choose a Good Distortion Measure," *in Proceedings of the 29th IEEE International Symposium on Information Theory*, Nice, France, pp. 566-570, 2007.

[16] Kosmopoulos A., Paliouras G., and Androutsopoulos I., "Adaptive Spam Filtering using only Naive Bayes Text Classifiers," *in Proceedings of the 5th Conference on Email and Anti-Spam*, Mountain View, pp. 1-3, 2008.

[17] Michelakis E., "Filtron: A Learning-based Anti-Spam Filter," *in Proceedings of the 1st Conference on Email and Anti-Spam*, California, pp. 1-8, 2004.

[18] Rish I., "An Empirical Study of the Naive Bayes Classifier," *in Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Washington State, pp. 41-46, 2001.

[19] Sahami M., Dumais S., Heckerman D., and Horvitz E., "A Bayesian Approach to Filtering Junk E-Mail," available at: http://research.microsoft.com/en-us/um/people/horvitz/spam.pdf, last visited 1998.

[20] Sculley D. and Wachman G., "Relaxed Online SVMs for Spam Filtering," *in Proceedings of the 30th Annual International Conference on Research and Development in Information Retrieval*, Amsterdam, pp. 415-422, 2007.

[21] Sculley D. and Wachman G., "Relaxed Online SVMs in the TREC Spam Filtering Track," available at: http://trec.nist.gov/pubs/trec16/papers/tuftsu.spam.final.pdf, last visited 2007.

[22] Tishby N., Pereira F., and Bialek W., "The Information Bottleneck Method," available at: http://arxiv.org/pdf/physics/0004057.pdf, last visited 1999.

Chen Yang received his BE and ME degrees from the School of Information Engineering, Zhengzhou University. Currently, he isa PhD candidate in School of Information, Renmin University of China, China and is also an assistant in School of Software Engineering at Zhengzhou University of Light Industry, China. His research interests include machine learning and BigData system.

Shaofeng Zhao received his BE and ME degrees from the School of Information Engineering, Zhengzhou University. Currently, he is an Assistant in library at Henan University of Economics and Law, China. His research interests include cloud computing and cloud storage.

Dan Zhang received her BE degree from the school of computer, Henan University of Economics and Law, and ME degree from the School of Information Engineering, Zhengzhou University. Currently, she is an Engineer in Geophysical Exploration Center of China Earthquake Administration. Her research interests include complex system and machine learning.

Junxia Ma received her ME degree from Zhengzhou University. Currently, she is a lecturer in the School of Software Engineering at Zhengzhou University of Light Industry, China. Her research interests include artificial intelligence, data mining