Adaptive Email Spam Filtering Based on Information Theory

Xin Zhang, Wenyuan Dai, Gui-Rong Xue, and Yong Yu

Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200240, China
{zhangxin, dwyak, grxue, yyu}@apex.sjtu.edu.cn

Abstract. Most previous email spam filtering techniques rely on traditional classification learning which assumes the data from training and test sets are drawn from the same underlying distribution. However, in practice, this *identical-distribution* assumption often violates. In general, email service providers collect training data from various public available resources, while the tasks focus on users' individual inboxes. Topics in the mail-boxes vary among different users, and distributions shift as a result. In this paper, we propose an *adaptive* email spam filtering algorithm based on information theory which relaxes the identical-distribution assumption and adapts the knowledge learned from one distribution to another. Our work focuses on the content analysis which minimizes the *loss in mutual information* between email instances and word features, before and after classification. We present theoretical and empirical analyses to show that our algorithm is able to solve the adaptive email spam filtering problem well. The experimental results show that our algorithm greatly improves the accuracy of email filtering, against the traditional classification algorithms, while scaling very well.

Keywords: adaptive, email spam, information theory.

1 Introduction

Email becomes increasingly popular in people's daily life while spam is more and more severe. As a result, people spend increasing amount of time for reading emails and deciding whether they are spam or non-spam. In this situation, a robust junk mail filter is highly needed. Usually, email service provides some kinds of spam filters to help users detect spam. In general, it is not able to train such filters based on labeled messages from individual users, but the available labeled data are often public, e.g. newsgroups messages or emails received through "spam traps"[1]. Although the spam emails do not much differ among various users' inboxes since spammers often send spam emails non-targetedly, the non-spam emails in the training set and users' inboxes could be rather diversity due to topics drifting among different users. In the case, a basic assumption of most machine learning techniques violates, which is the training examples should be drawn from the same underlying distribution as the test ones. If

[1] Spam traps are email addresses published visually invisible for humans but get collected by the web crawlers of spammers.

B. Benatallah et al. (Eds.): WISE 2007, LNCS 4831, pp. 159–170, 2007.

we use the public data to train a traditional classifier for predicting the class labels of the emails from individual users, the performance could be rather limited. Therefore, there is a critical email spam filtering problem how to classify emails under different distributions, which is known as *transfer* or *adaptive learning*.

In this paper, we focus on the problem of identifying spam emails. Different from traditional machine learning tasks, in this work, the training and test data come from different distributions. An adaptive email spam filtering algorithm based on information theory is proposed in this paper. We define an objective function (or evaluation criterion) for the categorization focusing on both training and test data. The objective function is formulated by the *loss in mutual information* between email instances and word features, before and after prediction. By optimizing the objective function, test emails should be well self-organized, while the labels in the training set could be considered as a kind of constraints to help the predictions. This classification method does not make the *identical-distribution* assumption, and hence is capable for the adaptive email spam filtering problem. The theoretical analysis shows that our algorithm is able to monotonically optimize the value of objective function. The experimental results support our theory and demonstrate that our algorithm is effective in adaptively filtering spam emails with a high rate of convergence.

The rest of paper is organized as follows. In Section 2, we introduce some preliminary concepts from information theory. The problem is presented in Section 3. Our AdaFilter algorithm based on information theory is proposed in Section 4. Section 5 presents the experiments and empirical analysis. In Section 6, we review some related works. We conclude the whole paper and give the future work in Section 7.

2 Preliminaries

We briefly introduce some preliminary concepts in information theory which will be used frequently in this paper. For more details, please refer to [7]. Let $\mathcal{X}$ and $\mathcal{Y}$ be two random variable sets with a joint distribution $p(x, y)$ and marginal distributions $p(\mathcal{X})$ and $p(\mathcal{Y})$. The *mutual information* $I(\mathcal{X}; \mathcal{Y})$ is defined as

$$I(\mathcal{X}; \mathcal{Y}) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \tag{1}$$

Mutual information indicates a kind of dependency between two variables. Larger mutual information value means more certainty that random variables depend on each other. *Kullback-Leibler* (KL) *divergence* [15] or *relative entropy* measures the distance between the two probability distributions. Let $p(x)$ and $q(x)$ be two probability mass functions where $x \in \mathcal{X}$. The KL-divergence is defined as

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \tag{2}$$

KL-divergence is a kind of distance between two different distributions, although it is not a real distance measure because it is not symmetric that $D(p||q)$ is usually unequal to $D(q||p)$. Besides, KL-divergence is always non-negative.

3 Problem Formulation

We formally define the problem as follows. Let $\mathcal{E}_{tr}$ be the (training) set of *labeled emails* using for training, $\mathcal{E}_{ad}$ be the (adaptive or test) set of *unlabeled emails* to be predicted. As we have discussed in Section 1, $\mathcal{E}_{tr}$ and $\mathcal{E}_{ad}$ are about the different topics under different distributions. We denote $\mathcal{E}$ as the set of all the labeled and unlabeled emails, so that $\mathcal{E} = \mathcal{E}_{tr} \cup \mathcal{E}_{ad}$. The word feature set of $\mathcal{E}$ is denoted by $\mathcal{W}$. For each email $e \in \mathcal{E}$, there is a class-label $c \in \mathcal{C} = \{spam, ham\}$ associated with it. Our objective is to estimate a hypothesis $h: \mathcal{E} \to \mathcal{C}$ which predicts the labels of spam/ham emails in $\mathcal{E}$ as accurately as possible.

4 The Adaptive Email Spam Filtering Algorithm

4.1 Objective Function

Let $\hat{\mathcal{E}} = \{\hat{e} | \hat{e} = spam \vee \hat{e} = nonspam\}$ be a prediction to $\mathcal{E}$ given by some hypothesis h. We consider $\mathcal{E}$, $\mathcal{W}$ and $\hat{\mathcal{E}}$ as the random variable sets which take emails, words, and predictions (*spam* or *nonspam*) as random variables. The *mutual information* $I(\mathcal{E}; \mathcal{W})$ measures the amount of information email instances $\mathcal{E}$ contain about their word features $\mathcal{W}$ [7]. Likewise, $I(\hat{\mathcal{E}}; \mathcal{W})$ measures the amount of information predictions $\hat{\mathcal{E}}$ contain about the word features $\mathcal{W}$. Since the emails are presented by word features, the mutual information $I(\mathcal{E}; \mathcal{W})$ could be considered as an information theoretic property about the data set $\mathcal{E}$. A good prediction $\hat{\mathcal{E}}$ should retrain this property. That is, $I(\hat{\mathcal{E}}; \mathcal{W})$ should close to $I(\mathcal{E}; \mathcal{W})$. In this paper, we use the *loss in mutual information* before and after prediction as the objective function which is formulated as

$$I(\mathcal{E}; \mathcal{W}) - I(\hat{\mathcal{E}}; \mathcal{W}). \tag{3}$$

The objective function is to be minimized. As we will show later in Lemma 1, the objective function is always non-negative.

In order to calculate Equation (3), let us first define two probability distributions, and then the approach for calculation will be presented in Lemma 1. Let $p(\mathcal{E}, \mathcal{W})$ denote the joint probability distribution of the emails $\mathcal{E}$ and the word features $\mathcal{W}$. The joint distribution under a prediction $\hat{\mathcal{E}}$ is denoted by $\hat{p}(\mathcal{E}, \mathcal{W})$ which is defined as

$$\hat{p}(e, w) = p(\hat{e}, w) p(e|\hat{e}) = p(\hat{e}, w) \frac{p(e)}{p(\hat{e})}, \tag{4}$$

where $e \in \hat{e}$. Note that $p(e|\hat{e}) = \frac{p(e)}{p(\hat{e})}$ since e totally depends on $\hat{e}$. Lemma 1 gives an alternative calculation approach for Equation (3). Furthermore, it also builds a connection between *loss in mutual information* and *KL-divergence*.

Lemma 1. *For a fixed prediction $\hat{\mathcal{E}}$, we can express the objective function in Equation (3) as*

$$I(\mathcal{E};\mathcal{W}) - I(\hat{\mathcal{E}};\mathcal{W}) = D(p(\mathcal{E},\mathcal{W})||\hat{p}(\mathcal{E},\mathcal{W})), \tag{5}$$

where $D(\cdot\,||\,\cdot)$ is the KL-divergence defined in Equation (2).

The proof of Lemma 1 is omitted due to the limitation of space. It can be easily derived by the definitions of the mutual information and KL-divergence. From Equation (5), we can find that the loss in mutual information before and after prediction equals in value to the KL-divergence between $p(\mathcal{E},\mathcal{W})$ and $\hat{p}(\mathcal{E},\mathcal{W})$. As a consequence, we can minimize $D(p(\mathcal{E},\mathcal{W})||\hat{p}(\mathcal{E},\mathcal{W}))$ instead of minimizing the objective function in Equation (3).

4.2 Optimization

Equation (5) is a function based on joint probability, which is difficult to optimize. In Lemma 2, we will convert the calculation of the objective function into the form of conditional probability. Then, the objective function can be optimized by minimizing the KL-divergence between two conditional probability distributions.

Lemma 2. *The objective function in Equation (3) can be rewritten into a form of conditional probability*

$$D\big(p(\mathcal{E},\mathcal{W})||\hat{p}(\mathcal{E},\mathcal{W})\big) = \sum_{\hat{e}\in\hat{\mathcal{E}}}\sum_{e\in\hat{e}} p(e)D(p(\mathcal{W}|e)||\hat{p}(\mathcal{W}|\hat{e})). \tag{6}$$

Proof

$$D\big(p(\mathcal{E},\mathcal{W})||\hat{p}(\mathcal{E},\mathcal{W})\big) = \sum_{\hat{e}\in\hat{\mathcal{E}}}\sum_{w\in\mathcal{W}}\sum_{e\in\hat{e}} p(e,w)\log\frac{p(e,w)}{\hat{p}(e,w)}.$$

Since

$$\hat{p}(e,w) = p(\hat{e},w)\frac{p(e)}{p(\hat{e})} = p(e)\frac{p(\hat{e},w)}{p(\hat{e})} = p(e)p(w|\hat{e}),$$

and

$$\hat{p}(w|e) = \frac{\hat{p}(\hat{e},w)}{\hat{p}(\hat{e})} = \frac{\sum_{e\in\hat{e}} p(\hat{e},w)\frac{p(e)}{p(\hat{e})}}{\sum_{e\in\hat{e}} p(\hat{e})\frac{p(e)}{p(\hat{e})}} = \frac{p(\hat{e},w)\sum_{e\in\hat{e}}\frac{p(e)}{p(\hat{e})}}{p(\hat{e})\sum_{e\in\hat{e}}\frac{p(e)}{p(\hat{e})}} = \frac{p(\hat{e},w)}{p(\hat{e})} = p(w|\hat{e}),$$

$D\big(p(\mathcal{E},\mathcal{W})||\hat{p}(\mathcal{E},\mathcal{W})\big)$ can be expressed by

$$\begin{aligned} D\big(p(\mathcal{E},\mathcal{W})||\hat{p}(\mathcal{E},\mathcal{W})\big) &= \sum_{\hat{e}\in\hat{\mathcal{E}}}\sum_{w\in\mathcal{W}}\sum_{e\in\hat{e}} p(e)p(w|e)\log\frac{p(e)p(w|e)}{p(e)\hat{p}(w|\hat{e})} \\ &= \sum_{\hat{e}\in\hat{\mathcal{E}}}\sum_{e\in\hat{e}} p(e)\sum_{w\in\mathcal{W}} p(w|e)\log\frac{p(w|e)}{\hat{p}(w|\hat{e})} \\ &= \sum_{\hat{e}\in\hat{\mathcal{E}}}\sum_{e\in\hat{e}} p(e)D(p(\mathcal{W}|e)||\hat{p}(\mathcal{W}|\hat{e})). \end{aligned}$$

□

From Lemma 2, we can find that optimizing $D\big(p(\mathcal{E},\mathcal{W})||\hat{p}(\mathcal{E},\mathcal{W})\big)$ is equivalent to optimizing $\sum_{\hat{e}\in\hat{\mathcal{E}}}\sum_{e\in\hat{e}} p(e)D(p(\mathcal{W}|e)||\hat{p}(\mathcal{W}|\hat{e}))$. Thus, for a single email instance e, if we want to decrease $D(p(\mathcal{W}|e)||\hat{p}(\mathcal{W}|\hat{e}))$, an alternative approach is to assign each e to a better $\hat{e}$ which is able to reduce the value of $D(p(\mathcal{W}|e)||\hat{p}(\mathcal{W}|\hat{e}))$. As a result, the objective function can be reduced through assigning better $\hat{e}$ to e. Then, the algorithm is derived as follows. For each email instance e, the algorithm chooses the best $\hat{e}$ to minimize the value of $D(p(\mathcal{W}|e)||\hat{p}(\mathcal{W}|\hat{e}))$, and then assigns e to $\hat{e}$. Based on Lemma 2, we know that this process can reduce the value of the objective function. The detailed description of the algorithm is presented in Fig. 1.

The Adaptive Email Spam Filtering Algorithm

Input: a labeled training set $\mathcal{E}_{tr}$; an unlabeled test (or adaptive) set $\mathcal{E}_{ad}$; an initial prediction $h^{(0)}$; the number of iterations T.

Output: the final prediction $h_f: \mathcal{E} \to \mathcal{C}$

1. Estimate the probability distribution p based on $\mathcal{E} = \mathcal{E}_{tr} \cup \mathcal{E}_{ad}$.
2. Initialize the probability distribution $p^{(0)}$ based on $h^{(0)}$ and Equation (4).
3. For $t = 1, \ldots, T$
4. Update the adaptive emails $\mathcal{E}_{ad}$ based on
$$h^{(t)} = \operatorname{argmin}_{c\in\mathcal{C}} D(p(\mathcal{W}|e)||\hat{p}^{(t-1)}(\mathcal{W}|c))$$
where $\hat{p}^{(t-1)}(\mathcal{W}|c) = \hat{p}^{(t-1)}(\mathcal{W}|\hat{e})$, and $\forall e' \in \hat{e}, h^{(t-1)}(e) = c$
5. For $e \in \mathcal{E}_{tr}$, $h^{(t)}(e) = h^{(t-1)}(e)$.
6. Update $\hat{p}^{(t)}$ based on $h^{(t)}$ and Equation (4).
7. End For
8. Return $h^{(T)}$ as the final prediction h_f.

Fig. 1. The description of the Adaptive Email Spam Filtering (AdaFilter) algorithm

In Fig. 1, in each iteration, the algorithm AdaFilter chooses the best label c for each email instance e in the adaptive set $\mathcal{E}_{ad}$ to minimize the function $D(p(\mathcal{W}|e)||\hat{p}^{(t-1)}(\mathcal{W}|\hat{e}))$. It has been proven that this process is able to reduce the value of the objective function. Note that, the predictions for all the $e \in \mathcal{E}_{tr}$ stay unchanged throughout the whole procedure, since these emails are already labeled.

4.3 Convergence

We have already presented the algorithm details. Since our algorithm AdaFilter is iterative, an important issue is to prove its convergent property. In the following theorem, we will show that AdaFilter could monotonically decrease the objective function, and then prove the termination property of AdaFilter.

Theorem 1. *The algorithm* AdaFilter *monotonically decreases the objective function in Equation (6). That is,*

$$D\left(p(\mathcal{E},\mathcal{W})||\hat{p}^{(t)}(\mathcal{E},\mathcal{W})\right) \geq D\left(p(\mathcal{E},\mathcal{W})||\hat{p}^{(t+1)}(\mathcal{E},\mathcal{W})\right) \tag{7}$$

Proof

Following Lemma 2, we have

$$D\left(p(\mathcal{E},\mathcal{W})||\hat{p}^{(t)}(\mathcal{E},\mathcal{W})\right) = \sum_{\hat{e}:h^{(t)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{p(w|e)}{\hat{p}^{(t)}(w|\hat{e})}.$$

Based on the Steps 4 and 5 in Fig. 1,

$$\begin{aligned}
&\sum_{\hat{e}:h^{(t)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{p(w|e)}{\hat{p}^{(t)}(w|\hat{e})} \\
&\quad\geq \sum_{\hat{e}:h^{(t)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{p(w|e)}{\hat{p}^{(t)}\left(w\middle|h^{(t+1)}(e)\right)} \\
&\quad= \sum_{\hat{e}:h^{(t+1)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{p(w|e)}{\hat{p}^{(t)}(w|\hat{e})} \\
&\quad\geq \sum_{\hat{e}:h^{(t+1)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{p(w|e)}{\hat{p}^{(t+1)}(w|\hat{e})} \\
&\quad= D\left(p(\mathcal{E},\mathcal{W})||\hat{p}^{(t+1)}(\mathcal{E},\mathcal{W})\right).
\end{aligned}$$

Note that, the third inequality follows by

$$\begin{aligned}
&\sum_{\hat{e}:h^{(t+1)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{p(w|e)}{\hat{p}^{(t)}(w|\hat{e})} \\
&\quad= \sum_{\hat{e}:h^{(t+1)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log p(w|e) \\
&\quad+ \sum_{\hat{e}:h^{(t+1)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{1}{\hat{p}^{(t)}(w|\hat{e})},
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{\hat{e}:h^{(t+1)}} \sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\log\frac{1}{\hat{p}^{(t)}(w|\hat{e})} \\
&\quad= \sum_{\hat{e}:h^{(t+1)}} \left(\sum_{e\in\hat{e}} p(e) \sum_{w\in\mathcal{W}} p(w|e)\right)\log\frac{1}{\hat{p}^{(t)}(w|\hat{e})} \\
&\quad= \sum_{\hat{e}:h^{(t+1)}} \hat{p}^{(t+1)}(\hat{e}) \sum_{w\in\mathcal{W}} \hat{p}^{(t+1)}(w|\hat{e})\log\frac{1}{\hat{p}^{(t)}(w|\hat{e})} \\
&\quad\geq \sum_{\hat{e}:h^{(t+1)}} \hat{p}^{(t+1)}(\hat{e}) \sum_{w\in\mathcal{W}} \hat{p}^{(t+1)}(w|\hat{e})\log\frac{1}{\hat{p}^{(t+1)}(w|\hat{e})}
\end{aligned}$$

Note that, the last inequality follows by the non-negativity of the Kullback-Leibler divergence. □

Theorem 1 proves that AdaFilter decreases the objective function monotonically. From Theorem 1, we know that AdaFilter is guaranteed to converge in a finite number of iterations. Note that, AdaFilter can only find a locally optimal solution, and finding the global optimal solution is NP-hard.

Regarding the computational complexity of AdaFilter, suppose the non-zeros in $p(\mathcal{E}, \mathcal{W})$ is N. In each iteration, AdaFilter calculates $h^{(t)}$ in $O(N)$ and update $\hat{p}^{(t)}(\mathcal{W}|\hat{e})$ in $O(|\mathcal{W}|)$. In general, $|\mathcal{W}|$ is not larger than N. Thus, the time complexity of AdaFilter is $O(N)$, which can be considered as good scalability.

5 Experiments

In this section, we empirically evaluate our algorithm AdaFilter. Our focus is the superior of the algorithm, so we did not pay much attention on optimization by feature selection or considering some structure features.

5.1 Data Set

In order to evaluate the properties of our algorithm, we conducted our experiments on the three data sets from ECML/PKDD discovery challenge 2006[2].

In this corpus, for each training data set, 50% of the labeled training data are assigned to spam which were sent by blacklisted servers of the Spamhaus project[3]. 40% of the labeled data are non-spam from the SpamAssassin corpus and the other 10% are non-spam which were sent by about 100 different subscribed English and German newsletters. The composition of the labeled training is summarized in Table 1.

Table 1. Composition of labeled train data

	#documents
emails sent from blacklisted severs	2000
SpamAssassin emails	1600
newsletters	400
total	4000

Three users' inboxes with the size of 2500 are tested in the experiments. Table 2 summarizes the composition of the evaluation inboxes consisting of 50% spam and 50% non-spam emails. For more details about the data sets, please refer to [3].

The test (or adaptive) data were collected from different users' inboxes using real but open messages. The non-spam part of the inboxes consists of ham-messages received by distinct Enron employees from the Enron corpus [14] cleaned from spam. The spam part of the inboxes consists of spam-messages from distinct spam sources. Since users' topic interests differ from each other, the distributions of emails are different among all the inboxes.

[2] http://www.ecmlpkdd2006.org/challenge.html
[3] http://www.spamhaus.org

Table 2. Composition of adaptive evaluation inboxes for data sets

Inbox ID	Non-spam	Spam source
0	Beck	Spam trap of Bruce Gunter (www.em.ca/~bruceg/spam)
1	Kaminski	Spam collection of SpamArchive.org
2	Kitchen	Personal spam of Tobias Scheffer (www.em.ca/~bruceg/spam)

Fig. 2 shows the instance-term co-occurrence distribution on the first data set. In this figure, instances 1 to 4000 are from the training data $\mathcal{E}_{tr}$ while instances 4001 to 6500 are from the user's inbox $\mathcal{E}_{ad}$. The instances are ordered first by their source ($\mathcal{E}_{tr}$or $\mathcal{E}_{ad}$, and second by their categories (*spam* or *nonspam*). The terms (or words) are sorted by $n_{+}(t)$ / $n_{-}(t)$, where $n_{+}(t)$ and $n_{-}(t)$ represent the number of occurrence the feature term t appears in spam and non-spam emails, respectively. From Fig. 2, it can be found that the distributions of training and test data are somewhat different, whereas the figure also shows amount of commonness exists between the two data. The common part ensures that feasibility for training data $\mathcal{E}_{tr}$ from public available source to help categorization on test data $\mathcal{E}_{ad}$ from users' inboxes.

5.2 Evaluation Metrics

The performance of the proposed methods was evaluated using two metrics. The first metric is accuracy, which expresses the proportion of email instances that are predicted correctly. The second metric is AUC value [16]. The AUC value is the area under the ROC curve (Receive Operating Characteristic curve). The AUC value specifies the confidence that the decision function assigns higher values to positive instances than to negative ones.

5.3 Comparison Methods

In order to show the superiority of our algorithm AdaFilter, we compare it with several traditional classification algorithms. We take Naïve Bayes classifier (NBC) [17] and Support Vector Machines (SVM) [5] as the baseline methods. Transductive Support Vector Machines (TSVM) is also introduced as a comparison semi-supervised

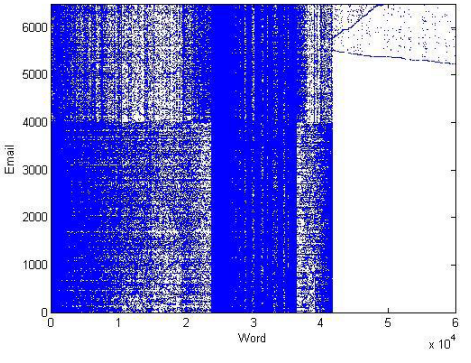

Fig. 2. Instance-term co-occurrence distribution on the first data set

learning methods. SVM and TSVM were implemented by SVM^{light} [12] with default parameters (linear kernel). The initial hypothesis $h^{(0)}$ of AdaFilter is given by NBC. All the comparison methods are implemented in the way that the classifier is trained using $\mathcal{E}_{tr}$ as labeled data and $\mathcal{E}_{ad}$ as unlabeled data, and then predict the labels of the emails in $\mathcal{E}_{ad}$.

5.4 Performance

Table 3 presents the performance in accuracy on each data set given by NBC, SVM, TSVM and our algorithm AdaFilter. We can see from the table that AdaFilter always provides the best results, compared with traditional classification methods. Besides, NBC gives better performance than SVM. In our opinion, although SVM is known as a much stronger classifier than NBC, NBC is more general for adaptive learning. Moreover, TSVM outperforms SVM, because it utilizes the information given by unlabeled data from the adaptive data set.

Table 4 presents the AUC values by NBC, SVM, TSVM and AdaFilter for each data set (each user's inbox). The AUC value specifies the confidence that the decision function assigns higher values to positive instances than negative ones. Thus, the decisions made by AdaFilter are more confident than NBC, SVM and TSVM.

Fig. 3 presents the accuracy on different sizes of labeled training data. We randomly choose the labeled data from $\mathcal{E}_{tr}$ of the first data set by different proportion. It can be seen that our AdaFilter algorithm always gives the best performance, while the proportion of the training data set changes from 1% to 64%. Furthermore, when the size of the labeled data decreases, the performance of AdaFilter drops much slower than SVM and NBC. We believe that the performance of AdaFilter is not much sensitive to the size of the training data set. NBC and SVM quickly get worse when the proportion of training data set is less than 16%. Meanwhile, our AdaFilter algorithm stays relatively more stably.

Table 3. Accuracy for each classifier on each data set

Data Set	NBC	SVM	TSVM	AdaFilter
0	0.764	0.713	0.752	**0.830**
1	0.774	0.726	0.770	**0.835**
2	0.861	0.856	0.929	**0.955**

Table 4. AUC for each classifier on each data set

Data Set	NBC	SVM	TSVM	AdaFilter
0	0.765	0.738	0.799	**0.875**
1	0.769	0.779	0.851	**0.854**
2	0.926	0.925	0.979	**0.992**

5.5 Convergence

Since our algorithm AdaFilter is iterative, the convergence property becomes an important issue. Theorem 3 has already proven the convergence of AdaFilter theoretically. Now, we empirically show the convergence property of AdaFilter. Fig. 4 gives the accuracy curves as functions measure the performances after each iteration on all the

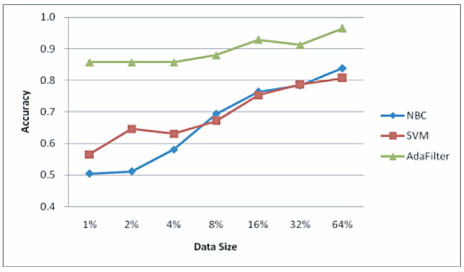

Fig. 3. Accuracy curves on different size of training data on the first data set

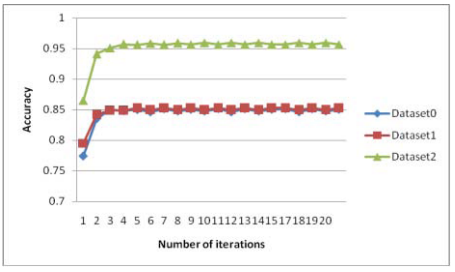

Fig. 4. Accuracy curves after each iteration on all the data sets

data sets. It can be seen in the figure clearly that AdaFilter always reaches almost convergent points within 5 iterations. As a result, we believe that AdaFilter converges very fast, and 5 iterations are enough in practice.

6 Related Works

In this section, we review some prior works mostly related to our work. Addressing on the email spam filtering problem, [22] proposed a Bayesian method to classify spam/non-spam emails. Three feature regimes were considered in their works, i.e. words only, words + phrases, words + phrases + domain specific features. [1] presented a throughout evaluation of the Bayesian methods for email spam filtering on an email spam corpus *Ling-Spam*[4]. [11] evaluated several machine learning techniques for detection spam emails, including C4.5 [20], Naïve Bayes Classifier [17], PART [10], Support Vector Machines [5], Rocchio [21]etc. All the above works assumes the training examples are under the same distribution from which the test data are drawn. However, as we have discussed in the previous sections, this assumption does not hold in general. Several other researches addressing on the same or similar work are [2, 6, 25, 19, 18] etc., to be mentioned.

Recently, email spam filtering has been recognized as an adaptive learning problem that the underlying distributions of the training and test data are different. Several heuristic approaches have been proposed during the last year's ECML/PKDD Discovery Challenge[5]. For example, the first place algorithm [13] proposed to use a simple statistical classifier which detects spam emails based on *strong* spam/non-spam words. Self-training using the statistical classifier as the basic learner is applied to improving the performance further. [4] addressed the same problem using a non-parametric hierarchical Bayesian model to learn a common prior and impose on a new email account. In contrast, we proposed a new method under the information theoretic framework. This framework, with well theoretic supporting, does not make any assumptions on the specific underlying distributions, while Bayesian model usually requires the distribution be Gaussian.

[4] http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz
[5] http://www.ecmlpkdd2006.org/challenge.html

Another related research is information theory based learning. Information theory is widely used in machine learning, e.g. Decision Tree [20], feature selection [24], etc. Recently, mutual information has been applied to improving clustering [23]. [8] proposed a word clustering method which minimizes the loss in mutual information between words and class-labels before and after clustering. Using the similar strategy, mutual information based co-clustering was proposed [9]. In contrast to these works, we try to design an information theoretic approach to solve the adaptive email spam filtering problem.

7 Conclusions and Future Work

Detecting email spam is challenging in both research and practice. In this paper, we focus on the email spam filtering problem that labeled and unlabeled data are under different distributions. Usually, this situation comes true, because email service providers gather training data from public available sources, but the test data are from users' individual inboxes. Due to the topic drift among different users, the underlying distributions from which the training and test data are drawn should be different as a consequence. In our work, an adaptive classification algorithm, called AdaFilter, is proposed based on information theory. The algorithm is motivated by minimizing the *loss in mutual information* between email instances and word features, before and after prediction. An iterative approach was designed to achieve the goal. Our theoretical analysis demonstrates that AdaFilter monotonically optimizes the value of the objective function. The experimental results support our theory and present the superior performance and scalability of AdaFilter, comparing with several traditional classification algorithms.

In our work, we focus on filtering email spam in one individual user's inbox by using global training data. In the future, we want to modify our algorithm in order to deal with multiple users' inboxes simultaneously. We believe the relations between different users' inboxes are able to help the prediction. Moreover, we also want to deal with the case that there are some but not sufficient labeled data in each user's inbox. Collaboratively using these labeled data could be a challenging and exciting task.

References

1. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G., Spyropoulos, C.D.: An Evaluation of Naive Bayesian Anti-Spam Filtering. In: Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (2000)
2. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Spyropoulos, C.D.: An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Encrypted Person-al E-mail Messages. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2000)
3. Bickel, S.: ECML-PKDD Discovery Challenge 2006 Overview. In: Proceedings of the ECML/PKDD Discovery Challenge Workshop (2006)

4. Bickel, S., Scheffer, T.: Dirichlet-Enhanced Spam Filtering based on Biased Samples. Advances in Neural Information Processing Systems (2006)
5. Boser, B.E., Guyon, I., Vapnik, V.: A Training Algorithm for Optimal Margin Classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory (1992)
6. Carreras, X., Mrquez, L.: Boosting Trees for Anti-spam Email Filtering. In: Proceedings of the 2001 International Conference on Recent Advances in Natural Language Processing (2001)
7. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience, New York, NY, USA (1991)
8. Dhillon, I.S., Mallela, S., Kumar, R.: Enhanced Word Clustering for Hierarchical Text Classifi-cation. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Dis-covery and Data Mining (2002)
9. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-Theoretic Co-clustering. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2003)
10. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning (1998)
11. Hidalgo, J.G.: Evaluating cost-sensitive unsolicited bulk email categorization. In: Proceedings of 17th ACM Symposium on Applied Computing (2002)
12. Joachims, T.: Learning to classify text using support vector machines. Dissertation, Kluwer (2002)
13. Junejo, K., Yousaf, M., Karim, A.: A Two-Pass Statistical Approach for Automatic Persona-lized Spam Filtering. In: Proceedings of the ECML/PKDD Discovery Challenge Workshop (2006)
14. Klimt, F., Yang, Y.: The Enron corpus: A new dataset for email classification research. In: Proceedings of the European Conference on Machine Learning (2004)
15. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statis-tics 22(1), 79–86 (1951)
16. Hanley, J., McNeil, B.: A Method of Comparing the Areas under Receiver Operating Characteristic Curves Derived from the Same Cases. Radiology 148, 839–843 (1983)
17. Lewis, D.D.: Representation and Learning in Information Retrieval. Doctoral dissertation, Amherst, MA, USA (1992)
18. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam Filtering with Naive Bayes? Which Naive Bayes? In: Proceedings of the 3rd Conference on Email and Anti-Spam (2006)
19. Michelakis, E., Androutsopoulos, I., Paliouras, G., Sakkis, G., Stamatopoulos, P.: Filtron: A Learning-Based Anti-Spam Filter. In: Proceedings of the 1st Conference on Email and Anti-Spam (2004)
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
21. Rocchio, J.J.: Relevance Feedback in Information Retrieval. In: The SMART Retrieval System: Experiments in Automatics Document Processing (1971)
22. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian Approach to Filtering Junk E-mail. In: AAAI 1998 Workshop on Learning for Text Categorization (1998)
23. Slonim, N., Tishby, N.: Document Clustering using Word Clusters via the Information Bottle-neck Method. In: Proceedings of the Twenty-Third Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2000)
24. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of Fourteenth International Conference on Machine Learning (1997)
25. Zhang, L., Yao, T.: Filtering Junk Mail with a Maximum Entropy Model. In: Proceedings of the 20th International Conference on Computer Processing of Oriental Languages (2003)