PSSF: A Novel Statistical Approach for Personalized Service-side Spam Filtering

Khurum Nazir Junejo
Dept. of Computer Science
Lahore University of Management Sciences
Lahore, Pakistan
junejo@lums.edu.pk

Asim Karim
Dept. of Computer Science
Lahore University of Management Sciences
Lahore, Pakistan
akarim@lums.edu.pk

Abstract

The volume of spam e-mails has grown rapidly in the last two years resulting in increasing costs to users, network operators, and e-mail service providers (ESPs). E-mail users demand accurate spam filtering with minimum effort from their side. Since the distribution of spam and non-spam e-mails is often different for different users a single filter trained on a general corpus is not optimal for all users. The question asked by ESPs is: How do you build robust and scalable automatic personalized spam filters?

We address this question by presenting PSSF, a novel statistical approach for personalized service-side spam filtering. PSSF builds a discriminative classifier from a statistical model of spam and non-spam e-mails. A classifier is first built on a general training corpus that is then adapted in one or more passes of soft labeling and classifier rebuilding over each user's unlabeled e-mails. The statistical model captures the distribution of tokens in spam and non-spam e-mails. This model is robust in the sense that its size can be reduced significantly without degrading filtering performance. We evaluate PSSF on two datasets. The results demonstrate the superior performance and scalability of PSSF in comparison with other published results on the same datasets.

1. Introduction

Spam, or unsolicited, e-mails have continued to be a major problem for e-mail users, network administrators, and e-mail service providers (ESPs). Spam e-mails reduce user productivity, clog network links, and waste computing cycles. Although technological and non-technological counter measures have been taken in the past several years, the menace of spam has grown in magnitude. In 1998, spam comprised of 10% of total e-mails sent. Today, this number has risen to over 80% of all e-mails, costing organizations worldwide $75 billion in lost productivity and anti-spam products and services [8, 15]. Moreover, there has been a noticeable surge in spam in the last two years indicating that spammers are moving aggressively against the counter measures [15].

E-mail users spend an increasing amount of time reading messages and deciding whether they are spam or non-spam and categorizing them into folders. Some e-mail clients make users label their received messages for training local spam filters. Similarly, some ESPs ask users to provide feedback on the received messages so that they can build personalized spam filters for them on the service side. Both of these strategies impose additional burden on users. Moreover, because of their volunteer and ad-hoc nature, both these strategies are neither automatic nor robust. E-mail service providers would like to relieve users from this burden by installing service-side spam filters that can classify e-mails as spam automatically and accurately without user feedback.

Automatic personalized service-side spam filtering is challenging along two dimensions. First, it presents a challenging machine learning problem where an accurate personalized spam filter is built for each user without relying on the user's feedback. Specifically, only a general set of labeled e-mails together with a set of unlabeled individual user's e-mails are available for building the personalized filter. Oftentimes the distribution of e-mails of individual users is not identical to that of the general set used for training. As such, a standard semi-supervised classification approach cannot be adopted. Secondly, automatic personalized service-side spam filtering presents several implementation challenges. In particular, such filters have to be space efficient, time efficient, and robust for them to be scalable to thousands and millions of users [13].

In this paper, we present a novel statistical approach for personalized service-side spam filtering (PSSF). PSSF builds a discriminative classifier based on a statistical model of spam and non-spam e-mails. Initially, the classifier is built on the general training set. Subsequently, it is adapted to the individual user's e-mails in one or more passes of soft labeling and classifier rebuilding. This approach allows automatic adaptation of the general classifier to the underlying distribution of e-mails in users' inboxes. The statistical model of spam and non-

0-7695-3026-5/07 $25.00 © 2007 IEEE
DOI 10.1109/WI.2007.47

spam e-mails is built from the tokens in the e-mail content. This is a robust and tunable model that can be scaled up to a large e-mail user base. PSSF is evaluated on two publicly available datasets for personalized spam filtering. The results demonstrate that PSSF's filtering performance is significantly better than other approaches when applied to the same datasets.

The rest of the paper is organized as follows. In section 2, we review content-based spam filters with specific focus on personalized spam filtering. Section 3 formally defines the personalized service-side spam filtering problem. The new approach, PSSF, is described in section 4. Section 5 presents the evaluation of PSSF on two datasets. We conclude in section 6.

2. Related work

Since the inception of e-mail spam, many technological and non-technological measures have been developed to combat it. Among the technological measures, content-based filtering has proven to be a critical anti-spam measure. Content-based spam filters employ machine learning techniques to learn to predict spam e-mails given a corpus of training e-mails. Such filters are typically deployed on the service-side mail server that filters e-mails for all users of the service. Bayesian approaches [9, 16, 17, 18] and support vector machines (SVM) [7, 12] have shown consistently good performances for this (non-personalized) problem setting. Bayesian approaches such as the naïve Bayes classifier is an example of a generative model based classifier while the SVM is an example of a discriminative classifier.

In recent years, there has been significant interest in personalized spam filters. This interest is motivated by the fact that the statistical distribution of e-mails in the training dataset is often different from that of individual users' inboxes, resulting in the poor performance of a single general filter for all users. Classical supervised and semi-supervised machine learning techniques assume that all e-mails are drawn independently from a given distribution. As such, such techniques cannot be directly applied to this problem setting. Furthermore, previous works on personalized spam filtering have relied upon user feedback in the form of e-mail labels from each individual user [10, 19]. This strategy burdens the e-mail user with the additional task of aiding the adaptation of the spam filter. Recently, a workshop was conducted on automatic personalized spam filtering [1]. Consequently, several interesting solutions have been presented for this problem setting ranging from statistical compression model based filters to dirichlet-enhanced generative models to hybrid clustering/ transductive SVM classifiers [2, 3, 4, 5, 11, 14].

Our approach, PSSF, is related to [5, 11] in that it is based on a statistical model of spam and non-spam e-mails and is similar to [14] in that it uses a discriminative classifier on preprocessed datasets. PSSF outperforms all previous solutions, as discussed in detail in section 5.

3. Problem definition

Let L be a set of labeled e-mails and U_1, U_2,…, U_m be sets of unlabeled e-mails. The set L corresponds to the general dataset available for training and the set U_i corresponds to unlabeled e-mails for user i. In general, it is assumed that the distribution of e-mails in L and U_i's are fixed, unknown, and different from one another. The ith e-mail in a set is defined as $\mathbf{x}_i = (x_{i1}, x_{i2},\ldots, x_{id})$, where $x_{ij} \in \{1,0\}$ indicates whether the ith e-mail contains token/word j (1) or not (0). It is assumed that the e-mails in all sets follow a common dictionary containing d tokens. We use the vector notation for simplicity of presentation; a bag-of-words representation is appropriate as well.

The task is to learn the filters $F_i : U_i \rightarrow \{+1,-1\}$($i$ = 1, m) that classifies an e-mail $\mathbf{x}$ for user i as either spam (+1) or non-spam (-1). The filter's performance is evaluated using the area under the receiver operating characteristics curve (AUC) [6]. We desire that the average (for all users) AUC value is as high as possible. To cater for concept drift in individual user's e-mails, we also desire that the filters can be adapted either incrementally or periodically with ease.

Since the personalized spam filters have to be deployed on the service side, we prefer that the filtering system is parsimonious, efficient, and robust for it to be scaled up to thousands and millions of users.

4. PSSF: a statistical approach for personalized service-side spam filtering

This section describes PSSF, a novel approach for personalized service-side spam filtering. PSSF combines the following characteristics: (1) a tunable statistical model of tokens (features) in spam and non-spam e-mails, (2) a generative model of e-mails for enhancing the feature space, (3) a discriminative classifier for the enhanced feature space, and (4) an adaptable procedure for personalization and concept drift tracking. The pseudo-code for PSSF is given in Figure 1; it is described in detail in the subsequent subsections.

```
Procedure: PSSF1/PSSF2
Input: Labeled set $L$, unlabeled sets $U_i$
Output: Filter for each user $i$, labeled sets $U_i$

On training set $L$
1. Build statistical model
2. Build discriminative classifier

On each user's inbox $U_i$
3. Repeat one or more times (requiring one pass over $U_i$)
4.     Label e-mails
5.     Rebuild statistical model
6.     Rebuild discriminative classifier (PSSF2 only)
7. End repeat
8. Label e-mails
```

Figure 1. Key steps in PSSF

4.1 Statistical model of spam and non-spam e-mails

The statistical model is built from a set of labeled e-mails. This set may be L or a soft-labeled U_i (this will become clearer in Section 4.3). Let n_i^S and n_i^N be the count of token i in spam and non-spam e-mails, respectively, in the set. We define the index set of significant spam (Z^S) and non-spam (Z^N) tokens as

$$Z^S = \{j \mid (n_j^S / n^S - n_j^N / n^N) > t\} \text{ and}$$

$$Z^N = \{j \mid (n_j^N / n^N - n_j^S / n^S) > t\}$$

where t is a non-negative real-valued threshold, and n^S and n^N are the total number of spam and non-spam e-mails, respectively, in the set. In other words, a token j is a significant spam (non-spam) token if its estimated probability given spam (non-spam) e-mails is greater than its estimated probability given non-spam (spam) e-mails by the threshold t. Note that $Z^S \cap Z^N = \{\}$ and $|Z^S| + |Z^N| \le d$. This strategy allows us to remove statistically insignificant tokens from our model of spam and non-spam e-mails. And, this model can be tuned by varying the value of t.

A weight is associated with each significant token as follows:

$$w_j = \begin{cases} (n_j^S / n^S)/(n_j^N / n^N) & \text{if } j \in Z^S \\ (n_j^N / n^N)/(n_j^S / n^S) & \text{if } j \in Z^N \end{cases}$$

The set of significant tokens and their weights can be found in a single pass over all e-mails in the labeled set. Notice that the weight of a token is simply the ratio of the estimated probabilities of the token given spam and non-spam e-mails. As such, this represents a generative model of the e-mails somewhat similar to that used by a naïve Bayes classifier. Unlike in a naïve Bayes classifier, however, the size of the statistical model can be tuned by varying the threshold t. It is also worth remarking that our statistical model has similarities to statistical data compression models [3, 5]. We retain high differential probability tokens as descriptors of spam and non-spam e-mails.

4.2 Discriminative classifier

After building the statistical model of e-mails, we construct a discriminative classifier for labeling e-mails in the set. This discriminative model is built in the statistically enhanced e-mail space. Specifically, we learn the following discriminant function:

$$\widetilde{F}(\mathbf{x}) = s \times \sum_{j \in Z^S} w_j x_j - \sum_{k \in Z^N} w_k x_k$$

where s is a positive real valued scaling factor and x_j is the value of the jth element of the vector $\mathbf{x}$. The first summation is over all significant spam tokens in $\mathbf{x}$ and the second summation is over all significant non-spam tokens in $\mathbf{x}$. An e-mail $\mathbf{x}$ is classified as spam if the discriminant function is greater than zero; otherwise, it is classified as non-spam. The weights w_j in this function are defined by the statistical model described in the previous subsection. The only variable is the scaling factor, which is selected such that the resulting discriminant function minimizes the misclassification rate over the labeled set.

The decision hyperplane is defined by the weights and the scaling factor in the feature space. There is no bias term, so the hyperplane passes through the origin. The scaling factor modifies the weights corresponding to the spam tokens only such that the hyperplane becomes an accurate classifier. The feature space is a unit hypercube of dimensionality equal to the number of significant tokens in the statistical model.

4.3 Personalizing the filter

The previous subsections describe how to build a classifier given a set of labeled e-mails (the training set L). The users' e-mails (U_1 to U_m) are unlabeled. We label them using the following procedure. In the first pass over e-mails in U_i, use the discriminant function to soft-label the e-mails and update the statistical model. At the end of the first pass, we rebuild the discriminative classifier using the new statistical model (i.e. find the s that results in maximum classifier performance over the soft-labeled U_i). We repeat the above steps a few more times depending on the trade-off desired between computational efficiency and filtering performance. In the last pass, the user inbox is labeled using the updated filter.

The above procedure permits easy adaptation of the filter. The statistical model can be updated incrementally

Table 1. Evaluation datasets' characteristics

	Dataset A	Dataset B
No. of labeled training e-mails	4000	100
No. of e-mails per user inbox	2500	400
No. of user's inboxes (m)	3	15

as new e-mails are seen by the filter capturing the changing distribution of e-mails received by the user. The scaling factor can be recomputed at periodic intervals (e.g. every week) to cater for significant changes in the distribution of e-mails.

Based on the personalization procedure, we define two variants of PSSF. PSSF1 is the variant in which the discriminative classifier is not rebuilt (i.e. the scale factor *s* is not recomputed) in every initial pass over the user's inbox, while PSSF2 is the variant in which the classifier is rebuilt after every pass over the user's inbox.

5. Experimental evaluation

We evaluate PSSF (PSSF1 and PSSF2) on two publicly available datasets and compare its performance with four previously published results on the same datasets.

5.1 Datasets

We use two datasets available from the ECML/PKDD Discovery Challenge website [1]. Each of these datasets, henceforth identified as dataset A and dataset B, contain a training set (L) and several users' inboxes (U_i) (Table 1). The number of e-mails in the training set and the users' inboxes is much larger in dataset A than in dataset B. Furthermore, the number of training e-mails in dataset B is less than the number of e-mails in users' inboxes. As such, dataset B represents a more challenging personalized spam filtering problem than that captured in dataset A. The composition of the training set in both datasets is: 50% spam e-mails sent by blacklisted servers of the Spamhaus project (http://www.spamhaus.org), 40% non-spam e-mails from the SpamAssassin corpus, and 10% non-spam e-mails from about 100 different subscribed English and German newsletters. The composition of e-mails in users' inboxes is more varied with non-spam e-mails of distinct Enron employees from the Enron corpus and spam e-mails from various sources.

All e-mails are represented by a list of tokens and the corresponding frequency of the token within the e-mail content (including the headers). All datasets follow the same dictionary.

Table 2. Results for dataset A (all values are AUC in %)

	PSSF1/PSSF2	PSSF1	PSSF2	Optimal
Inboxes	Pass 1	Pass 2	Pass 2	
U_1	96.35	98.60	98.99	99.99
U_2	97.37	98.78	99.58	99.96
U_3	94.59	99.43	99.22	99.91
Avg.	96.10	98.94	99.26	99.95

5.2 Results

The performance of PSSF1 and PSSF2 on datasets A and B are shown in Tables 2 and 3, respectively. We report the AUC values after the first, second, and (for dataset B only) third and fourth pass over the users' inboxes. The threshold *t* for these results is set at zero. The results show that the average AUC value of the filters increase significantly after the first pass over the users' inboxes. Soft labeling is done in the first pass by using the filter learned over the training set. When this filter is adapted to the distribution of e-mails in individual users' inboxes after the first pass the classification performance improves significantly. In the last columns of Tables 2 and 3, we also show the 'optimal' filters for each user's inbox. These filters are constructed on each user's inbox by assuming that the labels are known. In general, it is seen that the performance of PSSF for all users increases towards the optimal as the number of passes is increased. For dataset A, PSSF is able to learn a near optimal classifier after the first pass over the users' inboxes.

Comparing the results for the two datasets, it is observed that PSSF performs consistently well for all users in dataset A while its performances varies over a broader spectrum for users in dataset B. For example, the performance of PSSF1 for U_6 actually degrades with the number of passes. This is because of a marked difference in the distributions of e-mails in U_6 and the training set (L). The performance of PSSF2, on the other hand, degrades less significantly after the first and second pass over U_6. PSSF2 compensates for the differences in distributions by rebuilding the classifier (recalculating *s*) after each pass.

Dataset B represents a much more challenging personalized spam filtering problem because of the very small size (100 e-mails) of the training set and the small sizes (400 e-mails) of the users' inboxes. To explore the impact of training set size on performance, we ran PSSF on a modified dataset B. In this dataset, the training set is augmented with the labeled e-mails of users' inboxes 13, 14 and 15 (1300 e-mails). For this dataset, the average AUC value (for the first 12 users' inboxes) produced by PSSF is 97.46%. This is a substantial improvement in performance, highlighting the importance of a sufficiently

Table 3. Results for dataset B (all values are AUC in %)

	PSSF1/PSSF2	PSSF1			PSSF2			Optimal
Inboxes	Pass 1	Pass 2	Pass 3	Pass 4	Pass 2	Pass 3	Pass 4	
U_1	72.16	94.15	96.89	96.72	79.60	92.17	96.97	99.93
U_2	73.52	96.50	96.98	96.61	78.81	95.64	97.40	99.99
U_3	91.24	96.29	96.72	96.75	94.74	96.22	96.30	99.99
U_4	98.14	99.22	99.12	99.12	98.93	99.05	99.05	99.98
U_5	82.11	93.79	94.70	95.05	93.09	94.09	94.48	99.66
U_6	80.71	78.65	74.11	69.90	79.17	78.39	74.96	99.96
U_7	72.42	92.72	91.81	90.79	72.80	87.59	87.39	100.0
U_8	86.78	95.46	95.96	96.16	91.64	94.36	95.73	99.70
U_9	79.62	99.32	99.39	99.24	94.10	99.53	99.54	100.0
U_{10}	75.20	98.12	99.19	98.05	87.10	98.73	98.74	100.0
U_{11}	85.82	94.08	95.88	96.24	89.50	92.92	94.50	99.97
U_{12}	86.69	91.26	92.54	92.36	87.83	89.42	90.38	99.61
U_{13}	91.28	98.85	99.60	99.51	94.75	98.45	99.40	99.88
d14	83.12	88.21	90.23	90.74	84.88	88.41	89.23	99.84
U_{15}	75.49	90.52	95.50	97.92	82.79	89.33	96.01	99.92
Avg.	82.29	93.81	94.57	94.38	87.32	92.95	94.00	99.89

Table 4. Comparison with other techniques (all values are average AUC in %)

Technique	Dataset A	Dataset B
PSSF1	98.94	94.57
PSSF2	99.26	94.00
Junejo et al. [11]	98.75	---
Kyriakopoulou [14]	97.31	95.08
Cormack [5]	93.00	94.90
Cheng and Li [4]	93.33	---

large training set. Obtaining a sufficiently large training set is not difficult as several training corpora are readily available. Moreover, for a service-side implementation, collecting a sufficiently large set of users' e-mails is simply a matter of waiting for the e-mails to accumulate.

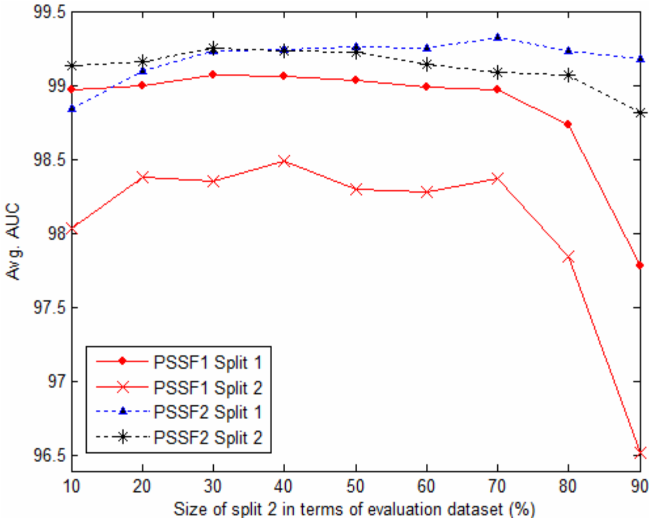

Figure 2. Generalization performance on dataset A

5.3 Comparison with other techniques

We compare PSSF's performance with four recently published results on the same datasets in Table 4. Three of these results [5, 11, 14] are winning performances of the Discovery Challenge [1]. PSSF outperforms all algorithms on dataset A and is on par with the others on dataset B. Junejo et al. has the previous best performance on dataset A (they do not report results for dataset B) [11]. PSSF1 improves on their algorithm by using estimated probabilities rather than occurrence counts for defining the statistical model, and PSSF2 rebuilds the discriminative classifier after each pass over the user's inbox. Kyriakopoulou and Kalamboukis preprocess the dataset by clustering the training set with each user's inbox [14]. The combined set is augmented with additional meta-features derived from the clustering. This combined set is then learned using transductive SVM. This approach is computationally expensive and non-adaptive. Cormack use statistical compression models for predicting spam and non-spam e-mails [5]. His approach is adaptive but the reported performances lag the leaders.

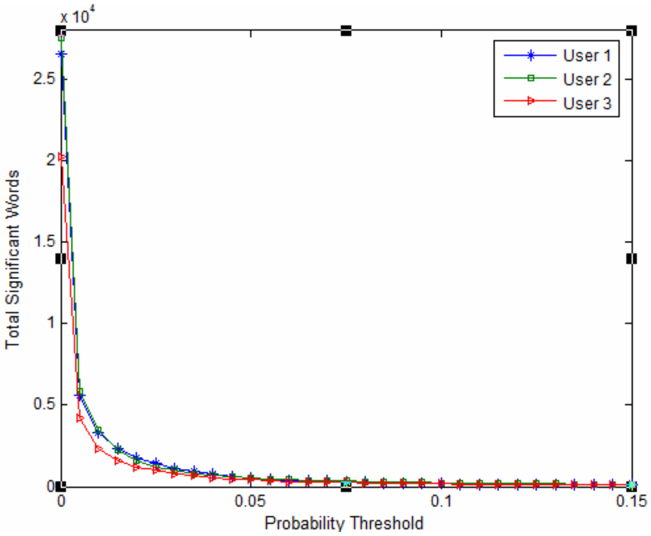

Figure 3. Number of significant tokens vs. threshold for dataset A

Cheng and Li present a semi-supervised classifier ensemble approach for the personalized spam filtering problem [4]. Their approach is also computationally expensive as compared to PSSF, and it lags in performance by more than 5% on dataset A (they do not report results for dataset B).

5.4 Generalization performance

The results presented in the previous subsections assume a transductive learning problem setting where all the unlabeled e-mails in users' inboxes are classified. However, in practice, once a personalized spam filter is learned using labeled and unlabeled e-mails it is applied to unseen e-mails. The performance over these unseen e-mails represents the generalization performance of the filter. We evaluate the generation performance of PSSF by splitting the users' inboxes into two: split 1 is used during learning and split 2 contains the unseen e-mails. The generalization performance of PSSF1 and PSSF2 on dataset A is shown in Figure 2. In general, the average AUC value over split 2 (the unseen e-mails) is less than that over split 1. However, this difference is typically less than 1% for PSSF1 and less than 0.2% for PSSF2. Furthermore, the decrease in average AUC value with increase in size of split 2 (decrease in size of split 1) is graceful. PSSF2, in particular, exhibits excellent generalization performance. It is able to learn the personalized filter for each user from a small number of e-mails for the user. This characteristic of PSSF2 stems from the realignment of the decision hyperplane after each pass over the user's inbox. These results demonstrate the robustness of PSSF.

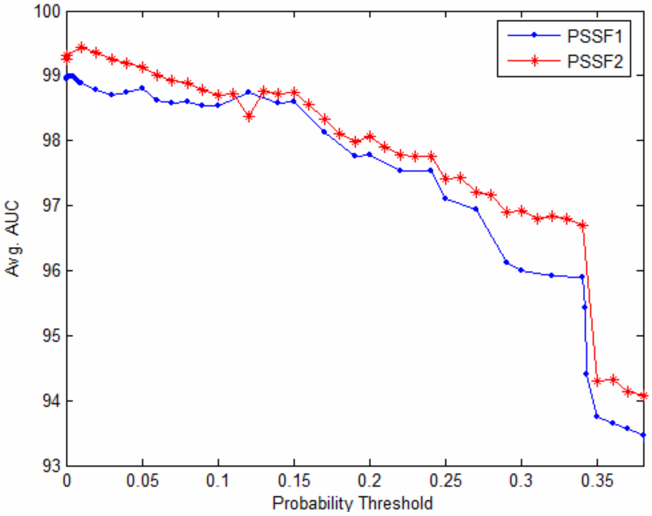

Figure 4. Average AUC value vs. threshold for dataset A

5.5 Scalability analysis

The size of the statistical model can be tuned by varying the value of the threshold *t*. This is evident from Figure 3 which shows the variation of the number of significant tokens (or words) with the threshold *t*. The number of significant tokens drops significantly with only a small increase in *t*. However, the remarkable result is that filtering performance does not drop significantly (and sometimes even increases) when the size of the statistical model is decreased (Figure 4). The performance increases for very small values of *t*. The threshold *t* acts as a feature selection filter which retains tokens that are highly discriminatory for spam classification. The weights of these tokens and the scale factor *s* together define the decision hyperplane for spam classification. As such, the threshold *t* can be used to tune the size and performance of personalized filters.

Robustness and scalability are essential characteristics for personalized service-side spam filtering [13]. To implement PSSF on the service-side for a given user, the statistical model and the scale factor have to be resident in memory. The statistical model comprises of the significant tokens and their weights. Table 5 shows the average personal filter size (as number of tokens in statistical model) and the average AUC value of PSSF for dataset A. It is seen that even when the average filter size is reduced by one-sixth (from 24776 to 4098 tokens) the average AUC value for PSSF1 remains unchanged and that for PSSF2 decreases slightly. Moreover, with an average statistical model size of only 18 tokens PSSF performs admirably with average AUC values greater than 95%. The average filter size is directly related to the scalability of the filter – the smaller the size the greater the number

Table 5. Impact of filter size on average AUC value for dataset A

Threshold	Tokens	PSSF1	PSSF2
0.0	24776	98.94	99.26
0.007	4098	98.94	99.40
0.24	47	97.53	97.76
0.34	18	95.89	96.90

of users that can be served with the same computing resources.

PSSF is built by learning over the training set and the users' inboxes. Learning the general filter from the training set involves a single pass over the set in which the statistical model is built. Subsequently, an iterative procedure is used for finding the scale factor. This step is performed once (or can be done periodically if new training sets become available). The general filter is then adapted in one or more passes over the users' inboxes. This step is also done once (or periodically). The filters can be adapted by continuously updating the statistical model (weights) and periodically recomputing the scale factor.

6. Concluding remarks

In this paper, we present a scalable and robust approach for personalized service-side spam filtering. The approach, named PSSF, uses a tunable statistical model of tokens in spam and non-spam e-mails to build a discriminative classifier. The issue of adapting the filter to the different distributions of unlabeled e-mails in users' inboxes is handled by multiple passes of soft labeling and statistical model rebuilding. PSSF can track concept drift by incremental update to the statistical model and periodic rebuilding of the discriminative classifier. Our experimental evaluations demonstrate the superior filtering performance of PSSF as compared to other published results on the same datasets.

Automatic personalized service-side spam filtering has generated much interest in recent times. E-mail service providers (ESPs) are seeking robust solutions that can relieve their users from providing feedback to achieve improved filtering performance. PSSF is a viable solution for ESPs promising scalable and accurate filtering. In the future, we plan to further develop the theoretical underpinnings of PSSF, investigate efficient data structures for large scale implementations, and incorporate cost-based measures of filtering performance.

7. References

[1] S. Bickell. ECML/PKDD: discovery challenge. http://www.ecmlpkdd2006.org/challenge.html, 2006.

[2] S. Bickel and T. Scheffer. Dirichlet-enhanced spam filtering based on biased samples. *Proc. of Neural Information Processing Systems (NIPS '06)*, 2006.

[3] A. Bratko, G.V. Cormack, B. Filipic, T.R. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, 7, 2673-2698, 2006.

[4] V. Cheng and C.H. Li. Personalized spam filtering with semi-supervised classifier ensemble. *Proc. of International Conf. on Web Intelligence (WI '06)*, 2006.

[5] G.V. Cormack. Harnessing unlabeled examples through application of dynamic Markov modeling. *ECML/PKDD Discovery Challenge Workshop*, 2006.

[6] C. Cortes and M. Mohri. AUC optimization vs. error rate minimization. *Proc. of Neural Information Processing Systems (NIPS '04)*, 2004.

[7] H. Drucker, D. Wu, and V.N. Vapnik. Support vector machine for spam categorization. *IEEE Transactions on Neural Networks,* 10(5), 1048-1054, 1999.

[8] J. Goodman, G.V. Gormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2), 25-33, 2007.

[9] P. Graham. Better bayesian filtering. *Proc. of 2003 Spam Conference*, http://www.paulgraham.com/better.html, 2003.

[10] A. Gray and M. Haahr. Personalised collaborative spam filtering. *Proc. of Conference on Email and Anti-Spam*, 2004.

[11] K.N. Junejo, M.M. Yousaf, and A. Karim. A two-pass statistical approach for automatic personalized spam filtering. *ECML/PKDD Discovery Challenge Workshop*, 2006.

[12] A. Kolcz and J. Alspector. SVM-based filtering of e-mail spam with content-specific misclassification costs. *Proc. of the TextDM Workshop on Text Mining*, 2001.

[13] A. Kolcz, M. Bond, and J. Sargent. The challenge of service-side personalized spam filtering: scalability and beyond. *Proc. of INFOSCALE*, 2006.

[14] A. Kyriakopoulou and T. Kalamboukis. Text classification using clustering. *ECML/PKDD Discovery Challenge Workshop*, 2006.

[15] N. Leavitt. Vendor's fight spam's sudden rise. *IEEE Computer*, 16-19, March 2007.

[16] E. Michelakis, I. Androutsopoulos, G. Paliouras, G. Sakkis, and P. Stamatopoulos. Filtron: a learning-based anti-spam filter. *Proc. of Conf. on Email and Anti-Spam (CEAS 2004)*, 2004.

[17] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian approach to filtering junk e-mail. *Proc. of AAAI Workshop on Learning for Text Categorization*, AAAI Technical Report WS-98-05, 1998.

[18] A.K. Seewald. An evaluation of naive Bayes variants in content-based learning for spam filtering. Kluwer Academic Publsihing, 2005.

[19] R. Segal, J. Crawford, J. Kephart, and B. Leiba. SpamGuru: an enterprise anti-spam filtering system. *Proc. of Conference on Email and Anti-Spam (CEAS '04)*, 2004.