Personalized Spam Filtering for Gray Mail

Ming-wei Chang[*]
Computer Science Dept.
University of Illinois
Urbana, IL, USA
mchang21@uiuc.edu

Wen-tau Yih
Microsoft Research
One Microsoft Way
Redmond, WA, USA
scottyih@microsoft.com

Robert McCann
Microsoft
One Microsoft Way
Redmond, WA, USA
robert.mccann@microsoft.com

Abstract

Gray mail, messages that could reasonably be considered either spam or good by different email users, is a commonly observed issue in production spam filtering systems. In this paper we study this class of mail using a large real-world email corpus and signature-based campaign detection techniques. Our analysis shows that even an optimal filter will inevitably perform unsatisfactorily on gray mail, unless user preferences are taken into account. To overcome this difficulty we design a light-weight user model that is highly scalable and can be easily combined with a traditional global spam filter. Our approach is able to incorporate both partial and complete user feedback on message labels and catches up to 40% more spam from gray mail in the low false-positive region.

1 Introduction

Publicly available email corpora for spam filtering often implicitly or explicitly assume that the label of a message does not depend on who receives the mail (Cormack & Lynam, 2005; Cormack, 2006). Although this assumption is somewhat necessary as a clear annotation guideline for creating benchmark corpora, unfortunately it does not always hold in practice. For example, a particular company may send monthly advertisements to past customers. Even though the email content is the same, some users consider this good mail while others treat it as spam (Fallows, 2003). As another example, it is common for users to begin reporting newsletters as spam rather than unsubscribing them, even if they had previously signed up to receive those newsletters (Email Sender and Provider Coalition, 2007). In these cases, nearly identical messages sent to multiple recipients have no globally correct label and can be reasonably treated as either good or spam. Such messages are called *gray mail*, which is first addressed in (Yih et al., 2007).

Not surprisingly, the seemingly inconsistent labels of gray mail messages present a difficult challenge to spam filtering. When learning a filter, the learner is faced with the problem of how to handle gray mail appropriately. One possible strategy is to treat it as a label noise issue, where the labels of some gray messages can be "corrected" before used for training. Perhaps more seriously, because identical messages will have the same predicted label at run time, the decision is a classification error to some users, even if the filter is globally "optimal".

To overcome these difficulties, in this paper, we first analyze the properties of gray mail using a large corpus obtained via campaign detection techniques. By clustering near-duplicate email in a collection of more than 2.6 million messages and examining their labels, we managed to obtain a large corpus of gray mail. Our study confirms that a big portion of email does belong to gray mail. Moreover, we show that a globally trained content-based filter performs poorly on this special category of mail and even a perfect filter will inevitably produce some classification errors. To solve the gray mail problem, certain degree of personalization is thus necessary.

For this purpose, we design an approach that incorporates user preferences into the classification model to avoid the limitations of global filtering. Unlike previous personalized filtering schemes which incur significant storage and processing costs per user, our user models are highly scalable and practical for large Web-based mail systems. We find that, with very little additional cost beyond current global filtering systems, we are able to incorporate both partial and complete user feedback on message labels and catch up to 40%

[*] This work was done while the author was an intern at Microsoft Research.

more spam from gray mail in the low false-positive region.

The rest of the paper is structured as follows. We first revisit the gray mail problem by measuring its pervasiveness and quantifying the limitations of global filters in Section 2. We then discuss the need for personalized filtering and propose various user models in Section 3, followed by the experimental evaluation in Section 4. Finally, we introduce other related work in Section 5 and conclude the paper in Section 6.

2 The Gray Mail Problem

In this section we study the effects of gray mail on spam filtering. We first describe how we obtain a gray mail corpus using signature-based campaign detection techniques and then proceed to quantify the prevalence of gray mail and the limitations it places on global filters.

2.1 Obtaining a Gray Mail Corpus

To obtain a gray mail corpus we mine a large email dataset for campaigns that have been labeled inconsistently by different recipients. The labeled messages come from the Hotmail Feedback Loop and the campaigns are detected with a recently developed near-duplicate detection technology. We describe each of these as follows.

The Hotmail Feedback Loop: The gray mail problem has been overlooked in the research community and can only be observed in a more realistic environment. Fortunately, having access to the Hotmail Feedback Loop data provides us the opportunity to examine this problem closely. The Feedback Loop data consists of messages labeled as spam or good by polling over 200,000 Hotmail volunteers daily. In this data collection mechanism each user's incoming mail is randomly selected, regardless of whether it is headed for the inbox, junk folder, or deletion. A special copy of the selected message is then sent to the user, asking him to annotate the original message as *good* or *spam*. Notice that this is not a truly random sample of mail sent to these users since each user receives at most 1 labeling request per day, and there is a significant fraction of mail that is immediately deleted and never enters this process (e.g., from block lists of clearly known spammers). Nonetheless, internal studies have shown that this set provides a reasonable approximation to mail received by Hotmail users. Most importantly, unlike in traditional research corpora, these messages are labeled by their intended recipients in real time. We believe we get the true, up-to-date personal judgements that only the mail recipients can make, which is crucial to a study on gray mail. In this analysis and the remainder of this paper, we use Feedback Loop data on messages received from January through May 2007.

Email Campaign Detection: Because gray mail is essentially messages that could be labeled either good or spam by different users, the straightforward method to find gray mail is to identify *identical* or *near-duplicate* messages in the dataset that have been labeled differently by different users. Messages with almost identical content and sent roughly in the same short period are usually called an email *campaign*. Although detecting email campaigns is an important anti-spam technique, not all of the campaigns are spam messages. Newsletters or commercial messages are often sent as email campaigns and can often be detected using the same method.

The campaign detection method we use in this paper is a variation of I-Match (Chowdhury et al., 2002), which has been shown very effective in finding near-duplicate email messages (Kolcz & Chowdhury, 2007). Briefly speaking, I-Match is one type of *fingerprinting* method that generates a signature for an email message. This method first pre-compiles a list of important words, or *lexicon*, from a large document collection. The signature is simply a hashed representation of the terms in the email that also occur in the lexicon. This method is further enhanced by Kolcz and Chowdhury (2007) to use not only uni-grams (i.e., words in the messages) but also some short n-grams based on a language model, which tends to be more robust to good-word attacks (Lowd & Meek, 2005) from spammers.

Applying the near-duplicate detection method on the Hotmail Feedback Loop mail collection, we are able to find several email campaigns or clusters of identical messages. If the messages in the same campaign are labeled differently, then we consider it as a gray mail campaign. Although the precision of this gray mail detection approach is fairly high, its recall is unfortunately limited by the sample size of the email collection. Remember that the Feedback Loop data is only collected from a small portion of Hotmail users. Despite the fact that it contains millions of messages, the dataset is still just a small sample of messages sent to Hotmail accounts. Therefore, small email campaigns may not always be detected by this method.

Notice that the campaign detection technique is mainly used for offline analysis of gray mail. For a real-time spam filter that needs to detect gray mail from large incoming mail streams, this is likely to be expensive. One alternative is to use sender reputation. For example, if mail from a given sender IP is consistently labeled as both spam and good by different users, then all the messages it sends in the future may be treated as gray mail. While this is not as precise as the campaign detection technique because some senders send a mix of clearly good and clearly bad mail (e.g., forwarders), as we will discuss later in Section 4.1, it is still a good and efficient alternative in practice.

2.2 Limitations of Global Filtering

Given a gray mail corpus, we can now quantitatively study the problem. Because a gray mail message can be labeled as either spam or good, a conventional global spam filter will be faced with the challenge of learning over "noisy" training data and will inevitably make mistakes at run time. This raises several interesting questions, such as "what percentage of mail belongs to gray mail?" and "how does gray mail affect filter performance?" In this subsection we measure the pervasiveness of gray mail and evaluate the use of noise reduction techniques to build a traditional global filter. While we do observe improved performance over a scheme in which the gray mail problem is simply ignored, we discuss why this is a less preferable approach to the problem. We then quantify the upper bounds that gray mail places on the prediction accuracy of any global filtering scheme and highlight the need for personalized filtering.

Pervasiveness of Gray Mail: In order to measure the ratio of gray mail versus all email messages, we analyze messages received in April and May of 2007. The number of total messages of this collection is 2,672,222. Among them, 1,553,519 (58.1%) were labeled as spam and the remaining 1,118,703 (41.9%) were labeled as good. Applying the near-duplicate detection method on this collection, we discovered 41,068 campaigns containing at least 5 messages, which accounts for 848,153 (31.7%) messages in total. Although a large number of these campaigns are either true spam or good campaigns, many of them are gray mail campaigns. Figure 1 shows the label consistency of these campaigns. The x-axis is the campaign spam ratio (i.e., the number of messages labeled as spam versus the total number of messages in a campaign) and the y-axis is the total number of messages in all campaigns with that spam ratio. As we can see from the figure, 28.6% of the messages belong to campaigns with spam ratio 1.0, the unambiguous spam campaigns. Similarly, at the other end of this figure, 4.6% of the messages belong to the "good" campaigns with spam ratio 0. Whether the messages from other campaigns are label errors or gray mail is less certain. If we assume there is no label error, then all campaigns other than spam and good are treated as gray mail, which has 66.8% of the campaign messages. However, if we treat only email campaigns with spam ratio between 0.2 and 0.8 as gray mail, then gray mail accounts for 25.4% of all campaign messages. In other words, at least 8.1% to 21.2% of all messages can be categorized as gray mail. The actual ratio could be higher since the near-duplicate detection method does not capture all campaigns due to the sampling issue discussed above.

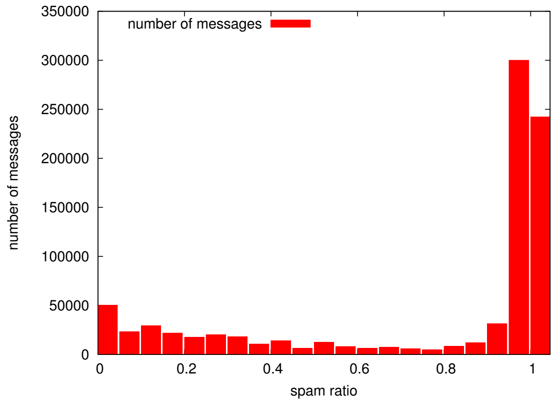

Figure 1: Volume of campaign mail by spam ratio.

A Label Noise Problem? Since gray mail presents challenges to global filters both during training and evaluation, we next quantify the effects of treating the gray mail problem as another form of label noise. How much better can global filtering become if we remove gray mail label "noise" from the training set? Also, given that a global filter cannot satisfy different user opinions on the same mail, how would the performance of a global filter change if we removed this "noise" from the testing set?

We investigate these questions as follows. First we choose January through March 2007 as our training period and April through May 2007 as our testing period. We then compare 4 configurations: cleaning the training data only, cleaning the testing data only, cleaning both the training and testing data, and no cleaning at all. To clean a given dataset, we first apply the campaign detection method on just that dataset and then force all campaign messages to have the same label as the majority vote within each campaign.

Using randomly selected 184,337 campaign messages from the training period, a spam filter is trained using content features such as words in the subject and body by logistic regression (Goodman, 2002). This filter is then tested using another 50,841 randomly selected campaign messages from the testing period.

Figure 2 shows the ROC curves of these four configurations in the low false-positive region. As indicated in the figure, although cleaning the training data con-

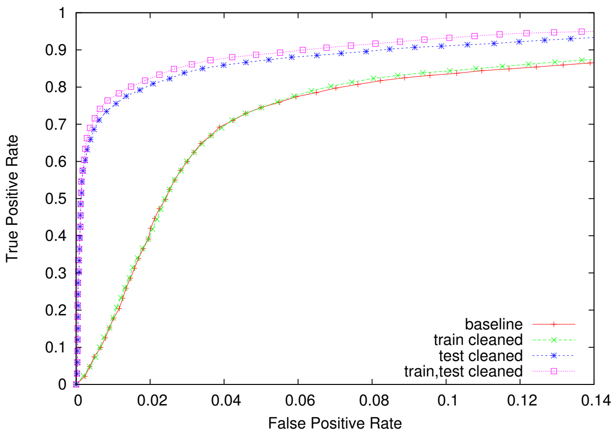

Figure 2: Global filter performance on campaign mail when treating gray mail as a label noise problem.

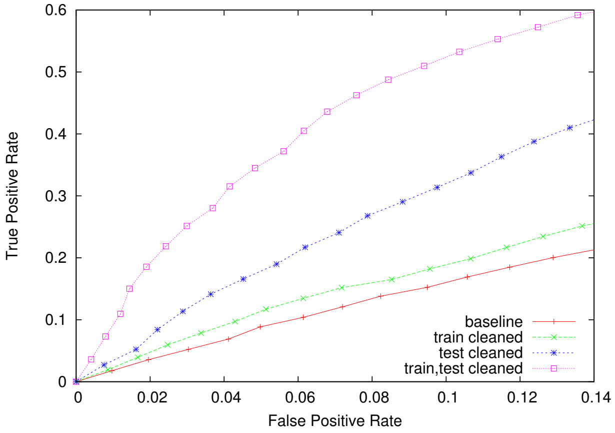

Figure 3: Global filter performance on gray mail in campaign messages when treating gray mail as a label noise problem.

sistently improves the filter (regardless of whether the testing data is cleaned or not), the gain is minimal. In contrast, most gain comes from cleaning the *testing* data. That is, if the gray mail is treated as label noise, then the performance of our spam filter on these campaign messages is in fact much better.

If we focus on messages that belong to clearly gray mail campaigns, then the performance difference is even more substantial. We preserved messages in campaigns with spam ratio between 0.2 and 0.8 in both training and testing data and repeated the experiments. Figure 3 shows the ROC curves of the corresponding four configurations. As also indicated by this figure, the most gain still comes from cleaning the testing data instead of cleaning the training data.

The above results seem to suggest that if gray mail is considered a label noise issue, then a global filter can perform well. This is especially true during evaluation if we use "cleaned" data to judge the effectiveness of our filter. Unfortunately, this view is not fair or practical. Users have different preferences and any majority-rules approach will not satisfy the needs of all users.

Optimal Global Filtering: A natural question to ask, then, is how good could an optimal global filter perform? To answer this question we assume that any standard global filter will output the same label for all messages in a campaign. An "optimal" filter will then take a majority-rules approach for each campaign to minimize errors. For example, if a campaign has 20 messages where 3 are considered spam by the recipients, then the filter should label all 20 messages as good, which generates 3 false-negatives. Applying this principle to these email campaigns, we found 23,749 false-positive cases and 17,319 false-negative cases. In other words, even an optimal classifier will have 1.54% classification error when the false-positive rate and the false-negative rate are about the same. In practice, though, filters are never perfect, and are usually tuned to operate in the low false-positive region. Applying a filter trained on the data collected in the training period to the messages sampled from the testing period, an internal study found that gray mail accounted for at least 9% of uncaught spam when operating at a low false-positive rate.

3 Incorporating User Preferences

Since treating gray mail simply as a label noise issue is unfair to some users, the spam filtering problem becomes more challenging as gray mail places considerable limitations on global filtering schemes. In particular, the correct email label not only depends on the message, but also on the recipient. In this paper we propose a personalized approach for handling gray mail. Unlike traditional personalized approaches, which often build personalized filters using training sets with similar distributions to the messages received by each user (Bickel & Scheffer, 2007; Segal, 2007), we seek a solution that respects the fact that different users have different opinions even on the same mail. Furthermore, we search for a solution that is appropriate for large-scale, Web-based email systems, such as Hotmail, Yahoo! Mail, GMail and AOL. In this setting, with a large number of users and nothing more than a Web browser on the client side, it is impractical to learn and apply heavyweight filters for each user. Complete feedback on message labels from each user cannot be assumed always available either. In short, the personalization scheme has to (1) respect each user's mail distribution and individual preferences, (2) incur negligible storage, training, and processing costs beyond a standard global filtering system, and (3) do not require complete feedback from each user.

To satisfy these requirements we propose using the partitioned logistic regression (PLR) model (Chang et al., 2008) that learns content and user models separately. While users share the same content model trained on all mail, the user model can be built efficiently using only a few statistics of the messages received by each user. The final prediction can be treated as a simple multiplication of these two models. In this section, we first briefly introduce partitioned logistic regression and then present how we learn the user model given either complete or partial user feedback.

3.1 Partitioned Logistic Regression

Conceptually, the *partitioned logistic regression* (PLR) model can be treated as a set of local classifiers that are trained by logistic regression using the same examples, but on different *partitions* of the feature space. When applied to the task of spam filtering, a message is represented by a feature vector $\mathbf{X} = \mathbf{X}_c\mathbf{X}_u$, where $\mathbf{X}_c$ and $\mathbf{X}_u$ are the content and user features, respectively. Given an example $\mathbf{X}$, the task is therefore to predict its label $Y \in \{0, 1\}$, which represents whether the message is good or spam. In the PLR model, such conditional probability is proportional to the multiplication of posteriors estimated by the local models.

$$\hat{P}(Y|\mathbf{X}) \propto \hat{P}(Y|\mathbf{X}_c)\hat{P}(Y|\mathbf{X}_u) \tag{1}$$

In particular, both the content and user models (i.e., $\hat{P}(Y|\mathbf{X}_c)$ and $\hat{P}(Y|\mathbf{X}_u)$) are logistic functions of the weighted sum of the features, where the weights are learned by maximizing the conditional likelihood of the training data.

The PLR model enjoys several advantages in practice. For example, its functional form is identical to the traditional logistic regression model learned on all the features. For a system that uses the logistic function for estimating probabilities, to change the model is straightforward – simply replacing the weights with the ones learned by the PLR model. It can also be shown that the multiplication of the local predictions in Equation 1 is equivalent to stating that different groups of features are conditionally independent given the class label, which makes partitioned logistic regression a hybrid model of the generative model, naive Bayes, and its discriminative counterpart, logistic regression. Finally, by training local models on different groups of features, the smoothing parameters can be easily tuned separately, which often yields better final predictions. For more discussions on the PLR model, see (Chang et al., 2008).

When the logistic regression model is used for binary classification, it is quite often that the conditional log-odds instead of the posterior is used as the decision function. While these two options produce equivalent ranking results, the log-odds is more convenient to use in practice since it is the weighted sum of the features. The final binary prediction of the message label is an indicator function – if the decision function is larger than a pre-selected threshold θ, then the message is classified spam.

Incorporating user preference in the PLR model as stated in Equation 1 can be viewed as if each individual user has his own decision threshold. Let o be the odds of the label given the example and let o_c and o_u be the odds of the content and user models, respectively. Then from Equation 1,

$$\begin{aligned} \log(o) > \theta \quad & \Leftrightarrow \quad \log(o_c) + \log(o_r) + k > \theta \\ & \Leftrightarrow \quad \log(o_c) > \theta - k - \log(o_u) \\ & \Leftrightarrow \quad \log(o_c) > \theta_u, \end{aligned}$$

where $k = -\log(P(Y = 1)/P(Y = 0))$, which is independent of u, and $\theta_u \equiv \theta - k - \log(o_u)$ is the new threshold for the mail recipient u.

3.2 User Model

As discussed previously, the goal of the user model is to capture the basic labeling preference of each mail recipient. In other words, we would like to know how likely a message will be labeled as spam by a user, without knowing the content of the email. Although some demographic information of a user, such as age or gender, may be loosely related to his mail preference, such information may not always exist and could be inaccurate. Therefore, in this work we choose a more direct and simple user feature – the recipient user id, which is treated as a binary feature. For example, if there are n users, then for a message sent to the j-th user, the corresponding user feature, x_j will be 1, but all other $n - 1$ features will be 0.

Note that by only using the user id in the user model, the model in fact estimates the "personal spam prior", $P(Y|u)$, for each user u, which is equivalent to estimate the percentage of spam in all messages this user receives. Despite the fact that such a model can be trained using traditional logistic regression learning methods, we can use a direct way to estimate the "inbox spam ratio" of the target user by counting the number of spam messages and all messages received by him in the training period. In the following, we first examine how we derive this model when complete user feedback on message labels is available. Perhaps more importantly, we also discuss how robust the model is when such feedback is limited.

3.2.1 Complete User Feedback

When the labels of messages sent to a target user are available, we use the spam ratio of these messages with a smoothing technique that is similar to using a Dirichlet prior. Let $cnt_{spam}(u)$ be the number of spam messages sent to user u, $cnt_{all}(u)$ the number of total messages this user receives, and $P_{spam} \equiv \hat{P}(Y=1)$ the estimated probability of a random message being spam. The user model is derived using the following formula.

$$\hat{P}(Y=1|\mathbf{X}_u) = \frac{cnt_{spam}(u) + \beta P_{spam}}{cnt_{all}(u) + \beta}, \qquad (2)$$

where β is the smoothing parameter. Notice that this maximum likelihood estimation is the same as logistic regression learning with feature vectors X_u; the only difference is the smoothing technique used in this method.

Similar to the common smoothing techniques used in the naive Bayes model, as the number of labeled messages increases, $cnt_{spam}(u)$ and $cnt_{all}(u)$ will be the dominant terms and the prior becomes less important. On the other hand, if there is no feedback from this user in the training period, the user model will reduce to the class prior, P_{spam}, which is simply the spam ratio of all the email in our collection.

3.2.2 Partial User Feedback

In more practical settings we will not know the *true* labels of all messages that a user has received. Even in this case, we can see from Equation 2 that it is still not difficult to set the denominator, which is essentially the total number of messages a user receives. The challenge, though, is to estimate the number of spam messages received by this user. It is common, however, to be able to collect some statistics to help make this estimation. For example, although only a very small portion of Hotmail users participate the Hotmail Feedback Loop, ordinary users still provide a form of feedback through "report as junk" buttons. This is a common UI in most Web Mail systems. When a spam message passes the filter and is delivered to someone's inbox, the user can press the button to move this message from the inbox to the junk folder, and report this message to the system.

There are a couple important issues when using the number of junk mail reports as the substitute of the real counts of spam messages. First of all, the user does not see all the messages sent to him. Messages that are highly likely to be spam may either be deleted or put in the junk folder directly by the filter. Second, not all users report junk mail. Therefore, the junk mail reports are in fact a specific subset of the spam messages sent to the user. Considering these two issues, we propose two formulas based on Equation 2.

The first formula assumes that all the spam messages delivered to the inbox have been reported as junk mail by the user. The total number of spam messages is therefore the count of junk mail reports plus the spam that is captured by the filter. Let *prec* be the *overall* precision of the filter[1]; namely, the number of true positives divided by the number of positive predictions. The number of caught spam messages of a recipient u, $ct(u)$, is thus $prec \cdot cnt_{filtered}(u)$, where $cnt_{filtered}(u)$ is the number of messages sent to this user but considered as spam by the filter. Let $jmr(u)$ be the number of junk messages reported by recipient u during the training period, then the final formula is:

$$\hat{P}(Y=1|\mathbf{X}_u) = \frac{ct(u) + jmr(u) + \beta P_{spam}}{cnt_{all}(u) + \beta} \qquad (3)$$

Equation 3 assumes that all the spam messages sent to the inbox have been reported by the user, which is often not true. One way to adjust this assumption is to add a term to estimate the number of spam messages that are *not* reported. Let $miss(u)$ be the number of spam messages that are not captured by the filter nor reported by the user. We use the following equation to estimate this term.

$$miss(u) = P_{spam} \cdot (cnt_{all}(u) - ct(u) - jmr(u))$$

The user model is therefore estimated as:

$$\hat{P}(Y=1|\mathbf{X}_u) = \frac{ct(u) + jmr(u) + miss(u) + \beta P_{spam}}{cnt_{all}(u) + \beta} \qquad (4)$$

Note that Equations 3 and 4 are not the only ways to estimate $P(Y=1|\mathbf{X}_u)$, and more sophisticated methods may exist. However, as we will show next in the experiments, these two models can already improve the performance significantly given partial feedback.

4 Experiments

We evaluate the proposed user models experimentally in this section. In all the experiments, email messages received between January and March 2007 are used for training, while messages received between April and May 2007 are used for testing. We first discuss the method of collecting most gray mail messages in an online spam filtering setting and then compare our user models in different scenarios.

[1] The *prec* parameter is estimated by applying the filter on the development set. Ideally, the precision of the filter should be estimated on messages of different users individually. In practice, the limited sample messages per user may not be able to provide a robust estimation. Therefore, we use the overall precision instead.

4.1 Data: Mixed-sender Mail

Although with labeled email messages, the campaign detection method described in Section 2 can capture gray mail with high precision, there exist several difficulties in applying it to detecting gray mail in an online, real-time spam filter. For example, despite the fact that near-duplicate messages sent in roughly the same short period can be clustered, knowing which of them belongs to gray mail campaigns still needs the labels of at least some of the messages. Unfortunately, because email is not always read right after received by the system, it takes some time to collect labels from volunteer users through means like the Hotmail Feedback Loop. A decision on whether an incoming message is gray mail cannot thus be reliably made *immediately* via the campaign detection method. Besides this critical issue, the coverage of detecting gray mail is also limited due to the sampling issue as discussed earlier.

One alternative of finding gray mail is to train a gray mail classifier using a corpus obtained via campaign detection techniques. While this approach has been proposed in (Yih et al., 2007), it seems to have limited success, partially due to the diversity of gray mail messages. On the other hand, identifying accurately gray mail messages may not be necessary since it is only an intermediate goal. Separating a subset of email that contains most gray mail and applying the proposed personalization schemes to improve spam filtering would be sufficient.

Because of the above practical considerations, we apply our methods to only the mail from *mixed senders*. Mixed senders are the IP addresses that used to send both good and spam messages in the past. Although some of them are clearly spam or good mail, these messages also cover most gray mail. Using the mixed-sender messages as the substitute of gray mail is also an efficient solution in practice since it only needs to maintain a list of mixed-sender IPs. Formally, we define the mixed senders as follows. Given an IP address i, let m_i be the set of messages sent from this IP address during a selected period. The spam ratio, r_i, is then the number of spam messages in m_i, divided by $|m_i|$ (the total number of messages in m_i). We then treat senders who sent greater or equal to 5 messages in this period with spam ratio between 0.2 and 0.8 as mixed senders. The set of mixed senders, S_{mixed}, is thus:

$$S_{mixed} = \{i \mid 0.2 \leq r_i \leq 0.8, |m_i| \geq 5\}.$$

Messages sent from S_{mixed} in the training and testing periods are sampled to construct our training and testing data and used in the experiments[2].

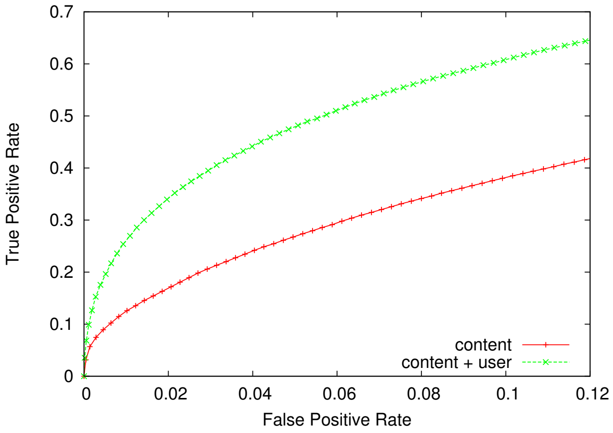

Figure 4: The ROC curves of the content-based filter and the model that incorporates user preferences.

4.2 Results

To fully evaluate the methods described in Section 3, we design two corresponding experimental scenarios. The first one is the *Complete User Feedback* scenario, which assumes that users provide the labels to all messages they receive. The other is the *Partial User Feedback* scenario, where we assume that for a group of users, only some spam labels are given through the junk mail report mechanism. We present the details of these two scenarios, along with the experimental results next.

4.2.1 Complete User Feedback Scenario

In this set of experiments, we use the traditional experimental setting: the content-based filter and the user model are trained using mail received in the training period, and the filters are tested on messages received in the testing period. In particular, we would like to examine how our personalization model can improve the accuracy of spam prediction over the regular filter on the mixed-sender mail, when the complete user feedback is available.

To build the conventional content-based filter, we train a logistic regression model using 700,000 randomly sampled mixed-sender messages received in the training period. The features used in this model are words in the subject and body fields, plus a very small set of some proprietary features. When combining the user

[2] Compared to detecting gray mail campaigns, treating mixed-sender messages as gray mail obviously has more coverage but less precision. By examining a subset of messages in an internal study, we found mixed-sender mail does have a high proportion of gray mail.

preference with the content-based filter via partitioned logistic regression (cf. Section 3), the user model is estimated by Equation 2, where the spam ratio of messages each user receives is also derived from the messages received in the training period. The smoothing parameter, β, is set to 1 for all users. Figure 4 shows the ROC curves of these two filters when applied to the 1,875,321 testing mixed-sender messages received by 197,183 different users in the testing period.

From the figure, the first thing we notice is that the conventional filter that relies only on the email content performs poorly on mixed-sender messages, where most of them are gray mail. For example, the true-positive rate at the false positive rate 0.1 (TPR@FPR=0.1) is merely 38.2%, indicating that a lot of spam messages in the gray mail category can easily pass the content-based filter. This result is essentially consistent with previous analysis on the gray mail corpus obtained using the campaign detection techniques (cf. Figures 2 and 3). However, incorporating the user model does improve the result quite substantially. As discussed earlier, our model can be treated as if each individual user has his own decision threshold of the filter. In spite of its simplicity, the true positive rate at 0.1 false-positive rate jumps from 38.2% to 60.8%, which indicates the importance of personalization in handling the gray mail issue.

4.2.2 Partial User Feedback Scenario

As discussed in Section 3.2.2, when applying the filter to mail sent to ordinary users who do not provide their label judgments, the main challenge is to construct the user model based on partial user feedback – the junk mail reports. In order to simulate this scenario, we further separate our data as follows. The recipients of the mail in our collection are first randomly split into two user groups of roughly equal size. The original messages used for training and testing are separated accordingly, as illustrated in Figure 5. We treat user group 1 as "known users" and user group 2 as "new users". In order words, the labels of all messages in collection A are assumed available, but only the labels of a subset of spam messages in collection C are revealed through junk mail reports. For this set of experiments, we will use the messages in collection A to train a content-based filter and use the partial labels of messages in collection C to build a user model. The combined model is then evaluated using mail in collection D.

	Jan-Mar, '07	Apr-May, '07
User Group 1	A	B
User Group 2	C	D

Figure 5: The data split for experiments of the partial user feedback scenario.

Recall that a junk mail report is essentially an uncaught spam message reported by the user. Therefore, to simulate such user behaviors, a base spam filter has to be built first. We build a content-based classifier using the messages in collection A, where the learning algorithm and features are the same as used in Section 4.2.1. We assume this filter operates at 0.1 false-positive rate due to the inherent difficulty of handling gray mail or mixed-sender messages, and select the decision threshold through cross validation on mail collection A. The precision of this classifier (used in Equations 3 and 4) is also estimated similarly. When applying this content-based filter to mail collection C, messages with probabilities of being spam lower than the threshold are predicted as good mail and delivered to the inboxes. The false-negative cases (i.e., uncaught spam) may be reported by the users. We introduce a parameter α as the report rate or the likelihood that an uncaught spam message will be reported as junk mail, and vary this parameter in the experiments to observe how the number of junk reports affects the results. In other words, for the *spam* messages which should not appear in the inbox, we assume the user have probability α to report it as a junk.

Notice that this approach of simulating junk mail reports is a simplified setting. In practice, a spam filter is often updated frequently using the latest training data and the uncaught spam messages are in fact the prediction results of various filters trained using messages received in different time periods. Using the mail received in the same period to train the filter for the purpose of simulating junk mail reports, we believe, captures the behavior of an continuously updated filter. Note that the true testing data, the mail in collection D, is still messages received in a non-overlapping time period.

We compared the two user models proposed in Section 3.2.2 when combined with the content-based filter and tested on mail collection D. Model 1 (Equation 3) assumes all uncaught spam messages are reported and model 2 (Equation 4) includes a correction term to estimate the counts of unreported junk mail messages. We assume the filter operates at 0.1 false-positive rate and show the corresponding true-positive rates of these two user models at different report rates. Figure 6 presents the results, where the x-axis is α, the probability of reporting the mistakenly classified spam messages, and the y-axis is the true-positive rate.

From the figure, we notice that the performance of both models is consistently improved as α increases. Moreover, model 2 performs better when the report rate is low, but not as good when this parameter be-

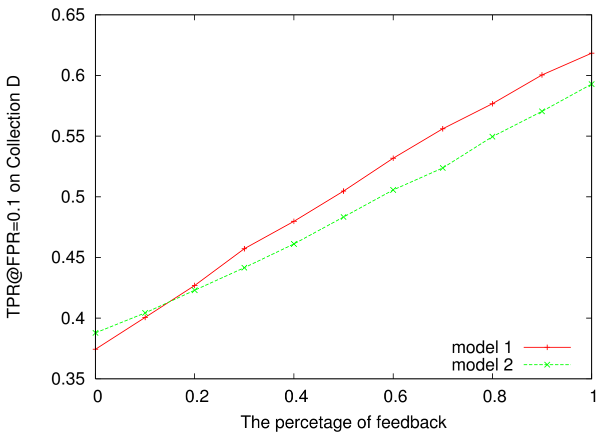

Figure 6: The true-positive rates at 0.1 false-positive rate of two user models when the feedback is limited. Model 1 assumes all uncaught spam messages are reported; model 2 includes a correction term that estimates the count of unreported junk mail messages.

comes larger. This phenomenon seems to imply that the correction term used in model 2 is useful only when most uncaught spam messages are not reported. Notice that even when such user feedback is limited, this additional information can still bring some improvement to the spam filter when processing gray mail or mixed-sender mail. For example, from Figure 6, when the report rate α is 0.2, model 1 increases the true-positive rate from 0.37 to 0.43, and model 2 has a similar improvement.

5 Related Work

Although the gray mail problem is a commonly observed issue in production spam filtering systems, it is often treated as normal label errors and has attracted little attention in the research community. A pioneering study on this problem was first done by Yih et al. (2007), where they proposed using campaign detection techniques to find gray mail and then build a classifier to distinguish gray mail from regular mail. Although they managed to show some improvement on spam filtering using a gray mail classifier, the scale of the experiments there was relatively small. In addition, gray mail was still processed by a regular content-based filter without taking the mail recipients into account. In contrast, we show the importance of email personalization to the gray mail problem and conduct our experiments using larger datasets.

Email personalization is treated as incorporating user preferences with a content-based classifier in the filter that is learned in the framework of partitioned logistic regression (Chang et al., 2008). This model can be viewed as a novel hybrid model between the generative model, naive Bayes, and its discriminative counterpart, logistic regression. It is especially suitable to our task since the content features and user features fall into different categories naturally. In this paper, we further enhance the user model and suggest alternative training methods that can also handle partial user feedback.

Note that our email personalization strategy is quite different from previous approaches. Personalized email spam filtering has typically been viewed as training a model that fits better individual mail distribution, instead of adjusting the filter to learn user preference. In particular, the class label of an email message is assumed to be independent of the recipient of the mail. For example, a Dirichlet process model to re-sample training data for each user is used in (Bickel & Scheffer, 2007), where the goal is to make the distribution of this new training dataset closer to the messages received by the user. Nevertheless, the strategy of training individual filters for different users is computationally expensive for a Web mail system that has hundreds of millions of user accounts.

A model combination approach has also been proposed recently for personalized spam filtering by Segal (2007), where a globally learned model is combined with a model trained using only messages sent to the target user. In comparison, our user preference model does not require the messages of individual users, but only the email labels. Partial feedback from the junk mail reports can also be used to enhance the filter when handling gray mail.

6 Conclusions & Future Work

In this work we addressed a difficult challenge for spam filters in practice – gray mail. Using a large mail corpus labeled by Hotmail users, we found that gray mail is a common problem and has placed significant limitations on global filtering schemes, even with the help of traditional noise-reduction techniques.

To address this challenge we proposed a personalized filtering approach based on the partitioned logistic regression model. We showed that, by incorporating individual user preferences directly into the model, we were able to significantly improve filter performance on gray mail. Perhaps more importantly, we also showed how our scheme was better suited to our target application – large-scale Web Mail systems – than previous work. Although there exist other personalized frameworks, most of them incur large storage and processing costs that may not be practical in such settings. Furthermore, some require extensive knowledge of each user and thus may not work well when only partial user feedback is available. In contrast, our scheme incurs very little additional cost above traditional global filtering schemes, and is designed to work even with only partial user feedback.

In the future we would like to explore additional personalization schemes to help solve the gray mail problem. While our approach effectively learns different filtering thresholds for each user, another complementary direction is to build explicit lists of black/white senders for each user. Despite the fact that most Web-based email systems today allow users to build such lists, gray mail is still a current problem. Therefore we would like to investigate this direction to help make it more effective in practice (e.g., by increasing user participation). Another possibility is to automatically infer these lists after observing user behavior. There are still several unanswered questions, though, such as determining what types and levels of user behavior are necessary to construct quality sender black/white lists? We feel that our approach is complimentary to this overall direction, though, since it provides a more personalized filter even in the cases when user black/white lists are ineffective (e.g., in first-contact scenarios). An effective gray mail solution may require a combination of several personalized schemes, and we feel that the solution we've proposed in this paper is a solid step in this direction.

Acknowledgements

We thank Alek Kołcz for helping us find the email campaigns in our Hotmail Feedback Loop message collection. We are also grateful to Chris Meek for his valuable comments.

References

Bickel, S., & Scheffer, T. (2007). Dirichlet-enhanced spam filtering based on biased samples. *Advances in Neural Information Processing Systems 19 (NIPS-2006)* (pp. 161–168).

Chang, M., Yih, W., & Meek, C. (2008). Partitioned logistic regression for spam filtering. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-08).*

Chowdhury, A., Frieder, O., Grossman, D., & McCabe, M. C. (2002). Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems (TOIS)*, *20*, 171–191.

Cormack, G. (2006). TREC 2006 spam track overview. *Proceedings of TREC-2006.*

Cormack, G., & Lynam, T. (2005). TREC 2005 spam track overview. *Proceedings of TREC-2005.*

Email Sender and Provider Coalition (2007). Consumers savvy about managing email according to ESPC survey results; embrace numerous tools and methods to manage spam reporting and unsubscribing. Email Sender and Provider Coalition (ESPC) press release, http://www.espcoalition.org/032707consumer.php.

Fallows, D. (2003). Spam: How it is hurting email and degrading life on the Internet. *Pew Internet and American Life Project.*

Goodman, J. (2002). Sequential conditional generalized iterative scaling. *ACL '02.*

Kolcz, A., & Chowdhury, A. (2007). Hardening fingerprinting by context. *Proceedings of the 4th Conference on Email and Anti-Spam.*

Lowd, D., & Meek, C. (2005). Good word attacks on statistical spam filters. *CEAS-2005.*

Segal, R. (2007). Combining global and personal anti-spam filtering. *CEAS-2007.*

Yih, W., McCann, R., & Kolcz, A. (2007). Improving spam filtering by detecting gray mail. *CEAS-2007.*