
©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Spam Filtering based on Preference Ranking

Mingjun Lan, Wanlei Zhou
School of Information Technology, Deakin University
221 Burwood Hwy, Burwood, Vic 3125, Australia
mingjun.lan@gmail.com
wanlei@deakin.edu.au

Abstract

When the average number of spam messages received is continually increasing exponentially, both the Internet Service Provider and the end user suffer[1-3]. The lack of an efficient solution may threaten the usability of the email as a communication means. In this paper we present a filtering mechanism applying the idea of preference ranking. This filtering mechanism will distinguish spam emails from other email on the Internet. The preference ranking gives the similarity values for nominated emails and spam emails specified by users, so that the ISP/end users can deal with spam emails at filtering points. We designed three filtering points to classify nominated emails into spam email, unsure email and legitimate email. This filtering mechanism can be applied on both middleware and at the client-side. The experiments show that high precision, recall and TCR (total cost ratio) of spam emails can be predicted for the preference based filtering mechanisms.

1 Introduction

Email filtering is the process of monitoring incoming (or outgoing) email, and then taking certain actions when an email is considered to be SPAM [4]. Spam constitutes a major problem for both e-mail users and Internet Service Providers (ISP) [5]. In general the word "spam" is used to refer to unwanted, "junk" email messages. Spam can often be referred to as unsolicited commercial e-mail or unsolicited bulk email; however, not all unsolicited e-mails are necessarily spam.

A lot of users see spam as annoying e-mails they can simply delete. They do not realize their real monetary impact. Actually spam is costly for both users and the ISP [5]. The spam cost to the ISP is more dramatic and can be seen at two levels: an increase on the load of e-mail servers and the waste of bandwidth. In addition, the average number of spam messages received is increasing exponentially. Figure 1 shows recent statistics on the number of spam messages received by one e-mail user, and taken from [6]. Fighting spam is necessary. The lack of an efficient solution may threaten the usability of email as a communication means.

![Figure 1 Annual Spam Evolutions](image)

*The number (218704) of 2004 is the result from linear prediction

Spam filtering can be applied at the client level or the server level. Several options are available at the client level for spam filtering [1, 4]. However, such lists are used by service providers and network administrators to block an email before it is sent; the unintended consequence of maintaining these blacklists is that sometimes, innocent senders are inadvertently blocked from sending legitimate emails. Spam filters are also effective against mass mailings of spam mail.

In this paper we present the filtering mechanism based on the preference ranking. Preference ranking is to calculate the similarity among various documents from a user’s preference sources. Spam filtering in both middleware and client-side is taken into consideration by the preference filtering mechanisms. The rest sections of the paper are organized as follows. Firstly we briefly introduce the current anti-spam technologies and related research work in section 2. Then we present our preference based filtering mechanism in an Internet framework in section 3.
Section 4 provides our experiment results and analysis. Finally we summarize this chapter.

2 Anti-Spam Technologies and Related Researches

2.1 Anti-Spam Technologies

Over the past few years, a lot of anti-spam tools and solutions based on different technological approaches have been developed [7]. However, as you will see below, there are significant differences in terms of the effectiveness of each approach.

Centralized filterig server

In this architecture, a single anti-spam filter runs on a centralized organization-wide mail server [3]. This approach eliminates the need to deploy software to email clients or to train users. Centralized filters have the disadvantage that they do not typically use the specific preferences and opinions of the user.

Gateway Filtering

In this approach, all inbound email is routed through a filtering gateway before being delivered to the mail server. Gateway services work well with web-based and mobile access to email, and may increase robustness since they queue emails if the client network or server is off-line. On the other hand, the gateway itself is a single point of failure and may be difficult to manage in the presence of multiple mail servers within an organization [3].

List-based filtering

This was the first solution to be proposed to fight against spams. Unlike all the following, it is a coarse-grained technique operating at the server level [3, 8, 9]. Today, both blacklisting and white-listing are considered ineffective, although server-based solutions adopt them as an auxiliary technique often to be integrated with challenge/response. However, blacklisting sources has become less effective since spammers learned to change their source address to get around the recipient’s defenses.

Rule-based filtering

Rule-based filters assign a spam score to each email based on whether the email contains features typical of spam messages, such as keywords and HTML formatting like fancy fonts and background colors [1, 3, 8]. A major problem with rule-based scores is that since their semantics are not well-defined, it is difficult to aggregate them and to establish a threshold that can actually limit the number of false positives.

Heuristic Filtering

In essence, heuristic filtering is a method of spam detection that uses baseline artificial intelligence to deliver an automated spam deletion process [5]. These automated mechanisms categorize incoming email messages as spam or legitimate based on known spam patterns. In theory, the advantage of this process lies in its automated nature and the fact that it should require no human intervention in the process of message classification. In reality, however, the greatest advantage of heuristics emerges as its greatest weakness.

Collaborative spam filtering

In collaborative approaches, server-side automatic monitoring systems consider whether incoming messages are to be known spam after these messages are classified by an automatic mechanism or by final recipients [3, 8]. These solutions have achieved considerable success as they overcome the single point of failure typical of centralized architecture. All the solutions presented above have strengths and weaknesses.

It is clear that no single technology is powerful enough to block all the spam that might flood an average mail server [7]. In fact, most anti-spam solutions combine two or more technologies in an attempt to improve their overall effectiveness, while decreasing their false positives ratio.

2.2 Related research

In [9] the authors present a Markov Random Field model based approach to filter spam. Their approach examines the importance of the neighborhood relationship among words in an email message for the purpose of spam classification.

A solution exploiting the P2P potential is proposed to reduce the level of spam [3]. An important strength of this proposal is that it is based on an open distributed architecture and does not rely on any authority or centralized control. The solution offers the opportunity to demonstrate how research on P2P networks, that has until now been perceived by a great part of the research community as mainly a mechanism to share copyrighted material, can be immediately adapted to contribute to the solution of an important and visible problem.
An additional layer in the spam filtering process is presented as a new spam filter [5]. This filter is based on a representative vocabulary. Spam e-mails are divided into categories in which each category is represented by a set of tokens which form a Representative Text (RT). Tokens are strings of characters (words, sentences, or sometimes meaningless strings of characters). This RT is used to compute a resemblance ratio with incoming e-mails. With this ratio one decides whether the incoming e-mail is a spam.

3 Preference based Filtering Mechanism

In this section we present the filtering mechanism after applying the idea of the preference ranking. Preference ranking is to calculate the similarity among various documents from a user’s preference sources. We use the Vector model [10] to realize this function. The framework of the filtering mechanism is shown in Figure 2. In this framework, spam filtering in both middleware and client-side is taken into consideration. As one knows, legitimate and spam emails are mixed and delivered through the Internet after different users send them out. In the middleware, the ISP’s Gateway/Proxy will filter off some ‘spam’ emails using its preference filtering system when these emails pass through it. There is a filtering point T that is set to realize this function. T is a real number. An email is blocked when its similarity value with a preference-based spam email is more than T. The set of preference-based spam emails are collected from the ISP’s users. A user can submit an email to the Middleware filtering system whenever he/she regards it as a spam. To avoid false spam submissions from users, we propose that the preference filtering system should have the white-list function. The white-list function can reduce the risk of cutting off legitimate emails. Emails will be sent to clients after they pass through the middleware filtering system.

In the client-side, a preference filtering system works similarly to the middleware one. The differences are that there are two filtering points T1, T2 in the client-side system. Here T1 and T2 are real numbers as well. The idea of two filtering points is to reduce the risk of misblocks of legitimate email. In our system, we will consider the emails that have a higher similarity value (the maximum value) with a certain preference email than T1 to be spam. The emails that have a similarity value (the maximum value) between T1 and T2 are considered unsure. These emails can be put in an unsure folder to let clients do a further check. After a user checks these unsure emails, he/she can decide whether to submit these emails to client-side and middleware filtering systems. The emails that have a similarity value (the maximum value) lower than T2 are regarded as legitimate ones. If a user finds a spam email from the legitimate set, he/she can submit it to the client-side filtering system.

From the above description, it can be seen that it is essential that all clients are encouraged to submit their spam emails to a client-side filtering system. If a client thinks a type of email is harmful to other users, he/she can submit it to a middleware filtering system. The white-list function in the middleware filtering system can avoid false submissions. Since both middleware and client-side filtering systems are built on the preference data source, they have a high reliability performance. At the same time, the filtering systems can index the preference spam source regularly.

Another essential thing is the filtering points T, T1, and T2. They must be set properly to make both systems work well. In [5], a similar cut-off point as T1 is given to be 0.2 in the client-side filtering system through their experiment demonstration. After we evaluated our preference filtering system, we would suggest the filtering points T, T1 and T2 as 0.3, 0.2 and 0.1 respectively. This suggestion can be proved by the following experiments of performance measurement.

4 Performance Measure

In this section we introduce the performance measurement method used in [2]. We present our experiment results to evaluate our preference filtering mechanism by this measurement method.
4.1 Measurement Methods

Let S and L stand for spam and legitimate message, respectively. \( N_{L\rightarrow L} \) \( N_{S\rightarrow S} \) denote the numbers of legitimate and spam messages correctly classified by the system. \( N_{L\rightarrow S} \) represents the number of legitimate messages misclassified as spam (false positive), and \( N_{S\rightarrow L} \) is the number of spam messages wrongly treated as legitimate (false negative). Then spam precision (p) and spam recall(r) are defined as follows:

\[
\text{Precision}(p) = \frac{N_{L\rightarrow L}}{N_{L\rightarrow L} + N_{L\rightarrow S}} \quad (1)
\]

\[
\text{Recall}(r) = \frac{N_{S\rightarrow S}}{N_{S\rightarrow S} + N_{S\rightarrow L}} \quad (2)
\]

When filtering spam, misclassifying a legitimate mail as spam is much more severe than letting a spam message pass the filter. Letting a spam go through the filter generally does no harm while misblocking an important personal mail as spam can be a real disaster. The usual precision/recall measures tell little about a filter’s performance when false positive and false negative are weighted differently. To introduce some cost-sensitive evaluation measures that assign a false positive a higher cost than false negative, a weighted accuracy (WAcc) measure specially tailored for this scenario can be used. WAcc was introduced and used in several spam filtering benchmarks [11] [8]. WAcc is defined as

\[
WAcc = \frac{N_{L\rightarrow L} \cdot N_{S\rightarrow S}}{\lambda \cdot N_{L\rightarrow L} + N_{L\rightarrow S} + N_{S\rightarrow S}} \quad (3)
\]

where \( N_{L\rightarrow L} \) is the total number of legitimate messages, and NS denotes the total number of spams. WAcc treats each legitimate message as if it were \( \lambda \) messages: when false positive occurs, it is counted as \( \lambda \) errors; and when it is classified correctly, this counts as \( \lambda \) successes. The higher \( \lambda \) is, the more cost is penalized on false positives. Androutsopoulos et al. [11] also introduced three different values of \( \lambda \): \( \lambda = 1, 9, \) and 999. When \( \lambda \) is set to 1, spam and legitimate mails are weighted equally; when \( \lambda \) is set to 9, a false positive is penalized nine times more than a false negative; for the setting of \( \lambda = 999 \), more penalties are put on false positive: misblocking a legitimate mail is as bad as letting 999 spam messages pass the filter. Such a high value of \( \lambda \) is suitable for scenarios where messages marked as spam are deleted directly.

In practice, when \( \lambda \) is assigned a high value (such as \( \lambda = 999 \)), WAcc can be so high that it tend to be easily misinterpreted. To avoid this problem, it is better to compare the weighted accuracy and error rate to a simplistic baseline. One can use the case where no filter is present as a baseline: legitimate messages are never blocked and spams can always pass the filter. Then the baseline versions of weighted accuracy and weighted error rate are

\[
WAcc^b = \frac{\lambda \cdot N_{L\rightarrow L}}{\lambda \cdot N_{L\rightarrow L} + N_{L\rightarrow S} + N_{S\rightarrow S}} \quad (4)
\]

\[
WErr^b = \frac{N_{S\rightarrow S}}{\lambda \cdot N_{L\rightarrow L} + N_{L\rightarrow S} + N_{S\rightarrow S}} \quad (5)
\]

To allow easy comparison with the baseline, Androutsopoulos et al. [11] introduced the total cost ratio (TCR) as a single measurement of the spam filtering effects:

\[
TCR = \frac{WErr^b}{WErr} = \frac{N_{S\rightarrow S}}{\lambda \cdot N_{L\rightarrow L} + N_{L\rightarrow S} + N_{S\rightarrow S}} \quad (6)
\]

Here greater TCR values indicate a better performance. If a TCR is less than 1.0, then the baseline (not using the filter) is better. An effective spam filter should be able to achieve a TCR value higher than 1.0 in order to be useful in real-world applications.

4.2 Experiments

Although there are available online spam corpuses such as [12], they do not contain a large amount of spam and have an excessive number of multiple copies of the same message. Furthermore, they need to be preprocessed in order to be a reasonable text analysis for our filtering computation. For all these reasons we create our own corpus from a few e-mail users. A corpus of approximately 1000 emails was collected. These emails belonged to five different categories of topics and also had a different number of words. Then we sent these emails to several clients who set up preference filtering systems. After we had applied the measurement methods in section 4.1, we obtained two types of experiment results, see Table 1.

From Table 1, one can see that the filtering point T in the middleware system would be 0.3. For three types of \( \lambda \), i.e. 1, 9, 999, all the value of TCR for filtering point T=0.3 is greater than 1.0. At the same time, the precision is 100%. This means the middleware filtering system can cut off around 20% to 60% of spam emails without any false positive risk. One can set it to be much stricter in the client-side filtering system, such as T1=0.2 and T2=0.1. The end users would accept the precision as above 98% with a high recall rate (around 70%). One can also see that the unsure filtering point (T2=0.1) would cover all kinds of spam (recall=100%) with precision above 85%. One observes that the number of words in the email has a
higher weight in Recall when the filtering point is set at more than 0.3.

![Table 1 Precision, Recall and TCR Results for Preference Filtering Mechanism](image)

<table>
<thead>
<tr>
<th>Filtering Point</th>
<th>Precision (p)</th>
<th>Recall (r)</th>
<th>TCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp 1*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>100%</td>
<td>18.7%</td>
<td>5.2</td>
</tr>
<tr>
<td>0.2</td>
<td>99.5%</td>
<td>66.7%</td>
<td>8</td>
</tr>
<tr>
<td>0.1</td>
<td>85.6%</td>
<td>100%</td>
<td>8</td>
</tr>
<tr>
<td>Exp 2#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>100%</td>
<td>62.5%</td>
<td>2.67</td>
</tr>
<tr>
<td>0.2</td>
<td>98.5%</td>
<td>78.3%</td>
<td>3.3</td>
</tr>
<tr>
<td>0.1</td>
<td>91.4%</td>
<td>100%</td>
<td>6</td>
</tr>
</tbody>
</table>

*In Exp 1, the number of words in an email is more than 500.  
#In Exp 2, the number of words in an email is less than 300.

![Figure 3 Precision and Recall trends in different long spam emails](image)

Figure 3 and Table 1 shows that the precision decreases and the recall increase when the set filtering point is set at a low value. At the same time, the false positive risk increases as well. However, middleware filtering systems can still improve their filtering performance after they collect a number of preference spam emails. For example, a spam sender might change the keywords, email address and subjects in his/her second spam group to overcome the most popular spam filters. With our preference filtering system, the similarity value would still be higher than 0.3. After a client submits one of a specific type of spam email, all successive emails can be blocked in the middleware filtering system. In this sense, high precision, recall and TCR would be predicted for our preference based filtering system.

5 Conclusions

In this paper we applied our preference based algorithms to spam filtering. we presented our preference based filtering mechanism for both middleware and client-side after introducing current anti-spam technologies. Instead of using many evaluations about precision and recall factors, we provided a false positive factor TCR to estimate the risk that misclassifies a legitimate mail as spam. Through our experiment results, we can provide reasonable filtering points for middleware and client-side filtering systems. Furthermore, high precision, recall and TCR would be predicted for successive spam emails after our preference based filtering systems was applied.

References