"May I borrow Your Filter?" Exchanging Filters to Combat Spam in a Community*

Anurag Garg   Roberto Battiti   Roberto G. Cascella
Dipartimento di Informatica e Telecomunicazioni,
Università di Trento,
Via Sommarive 14, 38050 Povo (TN), Italy.
{garo,battiti,cascella}@dit.unitn.it

Abstract

Leveraging social networks in computer systems can be effective in dealing with a number of trust and security issues. Spam is one such issue where the "wisdom of crowds" can be harnessed by mining the collective knowledge of ordinary individuals. In this paper, we present a mechanism through which members of a virtual community can exchange information to combat spam.

Previous attempts at collaborative spam filtering have concentrated on digest-based indexing techniques to share digests or fingerprints of emails that are known to be spam. We take a different approach and allow users to share their spam filters instead, thus dramatically reducing the amount of traffic generated in the network. The resultant diversity in the filters and cooperation in a community allows it to respond to spam in an autonomic fashion. As a test case for exchanging filters we use the popular SpamAssassin spam filtering software and show that exchanging spam filters provides an alternative method to improve spam filtering performance.

Keywords: Email filters, spam messages, collaborative recommendation systems, collaborative networks, trust, autonomic communication.

1 Introduction

An estimated 60% to 90% of email traffic on the Internet today can be called "spam". The problem of spam shows no signs of abating with every new popular spam filtering solution quickly being breached by new spam methods. Clearly, any static spam solution is doomed to failure. While early efforts at spam control concentrated on one-size-fits-all solutions that were situated at the Mail Transfer Agent (MTA), newer solutions recognize that what is spam for one person may in fact be ham for another. This has led to an emergence of personalized filters that can be customized by the user.

However, personalized filters often suffer from a lack of sufficient training data which reduces their effectiveness at least during the initial training period. Most ordinary users do not have the time or the skill required to train their filters properly. This often leads them to forgo personalized solutions and abandon themselves to the mercy of a third-party that decides for them what is spam and what is not. While this may not be such a bad solution for naive users, it raises concerns of censorship, as blacklists may be used to keep out dissent, and lack of diversity.

Autonomic computing offers a solution through the construction of a community that responds autonomically to spam by using collaborative spam filters that use machine learning. Decentralizing the filter exchange process gives the community a mechanism for self-management. Spamming is a volume business and a single spam email must be sent to hundreds or thousands of users to be economically profitable. If the first user who sees a spam email can share this information with others, they can automatically delete the email from their inboxes. This approach exploits the fact that the more the pairs of eyes that check for spam, the better the results should be.

Damiani et al. presented a solution in [9] where users' opinions are collected on what messages are spam and this collective judgment is used to block propagation of spam to other users. This requires identifying similarity among spam emails as spammers often introduce random content in spam emails to increase their chances of getting through

*This work was partly supported by the Provincia Autonoma di Trento under the WILMA Project and by the European Commission under the E-Next project FP6-506869 and the Integrated Project CASCADAS "Component-ware for Autonomic, Situation-aware Communications, And Dynamically Adaptable Services" (FET Proactive Initiative, IST-2004-2.3.4 Situated and Autonomic Communications) within the 6th IST Framework Program.

filters.

In this paper, we present an alternative to collecting user opinions that use digest-based indexing techniques. We present a framework that allows individuals in a community to exchange personalized filters instead of information about every spam email. As example of possible usage, two filters may be combined for increased effectiveness. A new user may "borrow" a filter from an existing community member to get him started. Thus a user may exchange a naive Bayesian filter for a rule-based filter by using our mechanism.

Individual filters may be trained using machine learning techniques [13]. A user may decide to give different weights to the acquired filters. She may decide that the filter received from a more technically savvy friend or from a friend who shares her interest more closely or that a filter that has been trained on more data deserves to be given a higher weight. Filter weights may also be adaptive and change according to the performance of individual filters.

Our second contribution is a general framework that can be used to combine the filters obtained by different users based on the trustworthiness and reputation of the filter provider.

The remainder of this paper is organized as follows. In Section 2 we discuss previous work and provide a brief overview of existing solutions. In Section 3 we discuss our core idea of exchanging filters instead of spam emails or their digests. This is followed by a brief discussion on the use of reputation to weight filters from users within the community. In Section 4 we present our experimental results and we conclude in Section 5.

2 Related Work

A number of different spam filtering techniques have emerged in the last few years. They include:

1. **List-based filtering** relies on white-lists (set of email address of users whose messages are allowed) and/or blacklists (email or IP addresses known to be used spammers). Lists are vulnerable to address spoofing and may also exclude receiving legitimate messages from users who are not in a white-list or who are present in a blacklist by mistake.

2. **Bayesian classifiers** classify email messages based on features extracted from the message. The features can be the words in a message or subsequences of words or characters [14].

3. **Rule-based filtering** defines a set of rules and corresponding weights (usually assigned through machine learning). Each email is checked against each rule to see if the rule is activated. In this case the rule weight is added to the message score. A message whose score exceeds a threshold is labeled as spam. Rules may include processing of email headers to check for abnormalities such as malformed headers or invalid return addresses and word or feature tests.

4. **Spam traps** publicize fake email addresses that do not belong to any user or group. Any message whose recipient list includes a spam trap address is discarded as spam.

5. **Sender authentication** verifies the identity of the sender before accepting an email. It can be performed by a challenge-response mechanism where the recipient sends a challenge to the sender which must be answered before the email is accepted.The challenge-response technique requires synchrony (it requires the sender and recipient to act within the specified time limit), thus negating a major advantage of email, and is vulnerable to address spoofing which can be used to launch a denial-of-service attack.

6. **Distributed spam identification** such as Vipul's Razor [2], SpamWatch [4], etc. Users detect spam messages and send periodically their reports to a central database so that subsequent arrivals of the same spam can be detected.

7. **Social email networks** exploit the already existing social structure in email networks to prevent spam. This approach is based on the assumption that users who exchange e-mail messages are connected in a social trusted network [7].

Combining and correlating classifiers has been used effectively in many fields such as document classification, speech recognition, optical character recognition (OCR) etc. A combination of classifiers can yield better results than those obtainable by the individual classifiers. Battiti and Colla [6] studied combining classifiers for OCR and found that teams work better than individual classifiers. They also studied the rejection/accuracy compromise: cases where the classification is dubious (e.g., the different classifiers disagree) are "rejected" and directly presented to the user who takes the final decision. In spam filtering, the rejection of classification a few emails (which can be placed in a specific folder) can be acceptable to the user if this strategy manages to reduce costly false positives. In general, a combination works only when classification errors are not completely correlated. This makes diversity in spam filters desirable because diverse filters tend to have uncorrelated classification errors.

In the field of spam recognition, Sakkis et al. [15] and Hershkop and Stolfo [11] have proposed combining spam classifiers. In particular, Hershkop and Stolfo combine classifier confidence factors instead of just a binary output and show that the latter strategy performs better.

3 Advantages of Exchanging Spam Filters

There are several reasons why it is preferable for users in a community to exchange spam filters rather than exchanging opinions about which messages are spam. Filter exchange and combination is an example of trusted community knowledge exchange that allows a community to respond to spam collectively and in an autonomic fashion. Using an already existing overlay network to exchange filters allows building email communities that span multiple administrative domains on the basis of shared email preferences. Further, situating filters at the individual user harnesses the computational power of user machines that far exceeds the power of a centralized spam filter.

Existing collaborative spam filtering mechanisms collect user opinions on whether a message is spam or not. Whenever a user detects a spam message she creates a hash digest of that email. The digest can be made resilient to common word-based attacks [8]. Digest information can either be collected at a central point [1, 2] or be shared among the members of a community using an overlay network [16]. In either case, the amount of traffic generated depends on the number of spam messages received by community members. A solution based on exchanging filters obviates the need for exchanging messages every time a spam message is received. Filters will be exchanged relatively rarely and the frequency of filter exchanges will be independent of the number of spam messages received.

Combining filters can be very useful in the case of multilingual users. Let us take the case of an individual from Italy who moves to the United Kingdom. This individual will now receive all work-related emails in English while still receiving personal emails in Italian. Such a user could easily retain the old "Italian spam filter" that he was using and simply combine it with an "English spam filter" borrowed from a co-worker or friend who receives her email in English.

Another important reason to prefer filter exchange is that of diversity. We know that in nature a population with a gene pool that is diverse has a greater chance of surviving a disease. Similarly, a community of users with a diverse set of filters is likely to be more robust against any one kind of spam. The diversity ensures that it will be highly unlikely that a particular spam is able to get through all the different filter combinations. A spammer would not be able to use a single set of tricks to defeat all filters.

This diversity and flexibility can also be related to the level of sophistication of the user. A naive user may simple "borrow" a filter from a user he trusts or an expert user who is known and trusted in the community. A more sophisticated new user may decide to combine filters from a number of different users. She may also decide to monitor the performance of the filter and tune the filter combination adaptively.

In order to allow users to exchange filters without compromising privacy it is important to separate the public and private parts of each filter (e.g., the white or black lists are in some cases private). We expect that users will not wish to exchange some personalized aspects of their filters. If these can be separated and only the public part of a filter is exchanged, any privacy concerns are adequately addressed. This separation must also be easily understandable by the user.

3.1 Using Trust for Effective Filter Exchange

Kong et al. [12] have proposed using a collaborative filtering approach within a social e-mail network for weighing reports of spam. Users have a trust value based on the strength of their ties to the network and the correctness of the spam report is weighted according to this trust value. They call their mechanism MailTrust and use an algorithm similar to the one used in PageRank.

We use the reputation of a user to weight a filter received. While the social email network can be used for the initial weights for the exchanged filters, our algorithm goes further and uses filter performance to modify the trust values of users in the community. It is possible to use a reputation management algorithm such as ROCQ [10] that allows users to rate the filters they send to other members of the community and the filter recipients to rate filter performance in turn and thus the trust value of users that send them the filters. In this manner, the community is made robust against malicious users (such as spammers) who may join the community and exchange filters that actually allow spam instead of blocking it.

4 Experimental Results

In this section we demonstrate a prototype filter exchange mechanism by combining rule-based filters. Due to space constraints we present results from our experiments with exchanging filters based on SpamAssassin only. SpamAssassin [3] is one of the most widely-deployed spam filtering solutions. It is also relatively flexible and combines many different kinds of spam filters such as header-processing, white-lists and blacklists and Bayesian techniques. SpamAssassin's rule-set is manually generated and it uses a simple neural network (perceptron). Each email is tested against the rules to determine the relative score.

Messages that score above a fixed threshold are marked as spam.

To validate the claim that the combination of individual filters improves the classification of spam messages, two distinct filters are defined to simulate filters produced by different users and then combined to produce a unique filter that is characterized by the union of both feature sets. The validation is given by testing the three filters on the same set of messages.

We randomly sort the set of rules available in SpamAssassin $3.1.0$ into two buckets so that each bucket characterizes a separate filter. The division is not completely random as it takes the in-built dependencies of the rules into account. The combined filter is constructed by merging the set of rules of each individual filter.

Our email workload consists of 4870 spam messages, downloaded from *http://www.spamarchive.org/* and of 4837 personal ham messages received by the authors. The combined workload is processed using the two filters individually and the filter derived by their combination. The output of the pre-processing stage for each of the three filters consists of a vector of 0s and 1s that indicates whether a specific rule has been activated in the email message (1 if the rule is activated 0 otherwise). This vector is the input for a *neural network* that uses a supervised learning strategy.

The neural network is trained over pre-classified messages so that the relationship between the input (vector of rules) and the output (classification of the message: spam or ham) is determined. Incoming messages are then classified based on the constructed neural network.

We consider the simplest form of neural network, the single-layer perceptron, with weights (w_i) assigned to each individual rule. The output is calculated as the scalar product of the vector of weights and the vector of inputs. A non-linear sigmoidal transfer function ($f(x) = 1/(1 + e^{-x})$) is applied to the result to constrain the output between 0 and 1: 0 for ham messages and 1 for spam messages. A message is classified spam if the output is greater than a fixed threshold (thr). The output of the neural network is continuous and may be interpreted as the degree of belief that a message is spam or ham.

The neural network is trained using information on second derivatives to determine the weight of each input. Details of the method used, *one-step-secant* (OSS), can be found in [5]. The error is represented by the energy function that is calculated as the sum-of-squared-differences between expected and evaluated values. The total energy, $E(w)$ for an iteration is:

$$E(w) = \frac{1}{2}\sum_{p=1}^{|P|} E_p = \frac{1}{2}\sum_{p=1}^{|P|}(t_p - o_p(w))^2 \qquad (1)$$

where the expected (target) value for pattern p is t_p and the computed value is o_p. The latter is a function of the weights,which are adjusted through an iterative minimization of the energy function. The number of iterations executed is determined in order to optimize validation performance.

The individual filters and the combined filter are tested on the same set of messages. We divided the collected messages in two sets, one for the training stage and one for validation stage. The training set consists of 5000 messages, with both spam and ham in equal proportion. The validation set consists of 4707 messages of which 2370 are spam messages and 2337 are ham messages.

The experiments are performed using different classification threshold thr values that vary from 0 to 1 with a step of 0.1. The percentage of false positives, i.e., ham messages that have been wrongly classified as spam messages and the resultant classification energy are evaluated for each of the three classifiers.

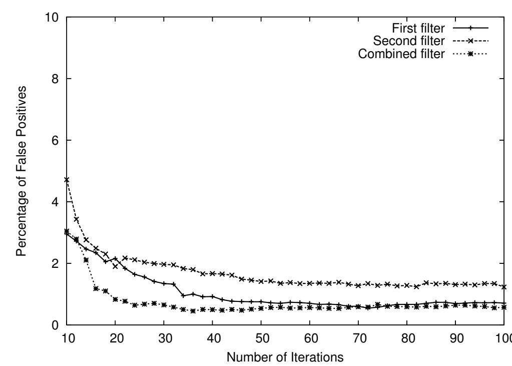

Figure 1. False positive rate - percentage of ham messages classified as spam messages with a classification threshold of 0.7.

Figure 1 shows how the false positive rate for the three filters varies with the number of iterations run on the training set. We see that the combined filter performs better than both individual filters and the percentage of false positive decreases as the number of the iteration increases. The sharp drop in the percentage of false positives during the initial phase (when the number of iterations is low) indicates incomplete classifier training.

Table 1 shows a snapshot of the performance of the classifiers at iteration 50. The True Positive (TP) rate is the number of real spam messages correctly predicted divided by the total number of spam messages and the False Positive (FP) rate is the number of real ham messages classified as spam divided by the total number of ham messages. The True Negative and False Negative rates are correspondingly defined. The combined filter performs better than the two individual filters since the false positive rate is the lowest and the true positive (detection) rate is the highest.

Classification	Classif. 1	Classif. 2	Combined
Spam as Spam (TP)	94.31%	96.30%	97.81%
Spam as Ham (FN)	5.69%	3.70%	2.19%
Ham as Spam (FP)	1.76%	2.74%	1.43%
Ham as Ham (TN)	98.24%	97.26%	98.57%

Table 1. Performance of the three classifiers

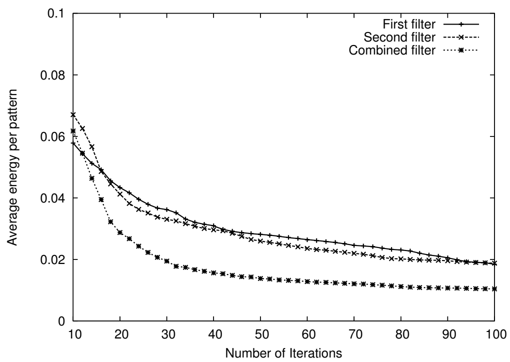

Figure 2. Energy function value per pattern for the classification.

As defined in equation1, the energy function depends on the classification error. Figure 2 plots the average energy per pattern for each classification or $\frac{E(w)}{|P|}$. Again, the average energy for the combined filter is lower than that for both filters. Thus, we can conclude that combining the two filters gives us better performance.

5 Conclusions

We have presented a new approach to collaborative filtering of email messages based on the exchange of filters developed by different members of the community.

Sharing and exchanging filters instead of emails or digests has many potential advantages ranging from a much lower required communication cost (e.g., in neural networks, a filter built from millions of messages can be described with some hundreds of byte to describe the network weights), a potential large diversity, a much greater flexibility in using and combining different filters. Preliminary experimental results have been provided to motivate the validity of the approach.

References

[1] http://www.cloudmark.com/.

[2] Razor's vipul. http://razor.sourceforge.net/.

[3] Spamassassin. http://spamassassin.apache.org/.

[4] Spamwatch. http://www.cs.berkeley.edu/ zf/spamwatch/.

[5] R. Battiti. First-and second-order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4:141–166, 1992.

[6] R. Battiti and A. M. Colla. Democracy in neural nets: Voting schemes for accuracy. *Neural Networks*, 7(4):691–707, 1994.

[7] P. O. Boykin and V. P. Roychowdhury. Leveraging social networks to fight spam. *IEEE Computer*, 38(4):61–68, April 2005.

[8] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. An open digest-based technique for spam detection. In *2004 Intl. Workshop on Security in Parallel and Distributed Systems*, San Francisco , CA, United States, Sept. 15-17, 2004.

[9] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. P2p-based collaborative spam detection and filtering. In *Proceedings of the 4th Intl. Conf. on Peer-to-Peer Computing (P2P'04)*, pages 176–183, August 2004.

[10] A. Garg, R. Battiti, and R. Cascella. "Reputation Management: Experiments on the Robustness of ROCQ". In *Workshop on Autonomic Communication for Evolvable Next Generation Networks - 7th Intl. Symp. on Autonomous Decentralized Systems*, Chengdu, China, April 4-8 2005.

[11] S. Hershkop and S. J. Stolfo. Combining email models for false positive reduction. In *KDD '05: Proceeding of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 98–107, Chicago, Illinois, USA, 2005. ACM Press.

[12] J. S. Kong, P. O. Boykin, B. A. Rezaei, N. Sarshar, and V. P. Roychowdhury. Let your cyber alter-ego share information and manage spam, 2005.

[13] B. Massey, M. Thomure, R. Budrevich, and S. Long. Learning spam: Simple techniques for freely available software. In *Proceedings of USENIX 2003 Annual Technical Conference, FREENIX track*, pages 63–76, June 2003.

[14] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*.

[15] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. In *Proceedings of EMNLP-01, 6th Conference on Empirical Methods in Natural Language Processing*, 2001.

[16] F. Zhou, L. Zhuang, B. Y. Zhao, L. Huang, A. D. Joseph, and J. Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *Middleware 2003: ACM/IFIP/USENIX International Middleware Conference*, volume 2672 of *Lecture Notes in Computer Science*, pages 1–20, Rio de Janeiro, Brazil, June 16-20, 2003. Springer.