

The challenges of service-side personalized spam filtering: scalability and beyond

Aleksander Kolcz, Michael Bond, James Sargent
America Online Inc., 44900 Prentice Drive, Dulles, VA 20166, USA

Abstract—Spam filtering of the email stream at the enterprise level poses many challenges especially at the scale of large Email Service Providers (ESPs). The problem is compounded if filtering is to be done on a personal level, with different configurations being adapted on a per-user basis. Commonly, the cost and performance issues are avoided by pushing personalized filtering to the client machine owned by the user, but this changes the user experience depending on the client used to access the mailbox. When implementing personal spam filters as a services, the benefits stemming from increased spam-detection accuracy need to be carefully balanced with the associated costs, especially in view of a large users population and co-existence with user-independent detection engines. The paper describes the challenges associated with implementing large-scale personalized spam-filtering service ranging from the need to scale with the user population to the challenge of being constrained by a fixed budget.

I. INTRODUCTION

Spam has become a major nuisance for email users across the Internet. Although the volume of spam in relation to other mail oscillates over time and despite both legal and technological efforts to curb spam, the levels of spam continue to be significant and spam accounts for the majority of email delivery attempts. Spam is a problem not only for end users, but also for network infrastructure providers and email service providers (ESPs), who need to expend bandwidth, storage and computational resources necessary to handle both the wanted and the unwanted messages. It is therefore often popular to perform some form of filtering at the enterprise or ESP level, which allows one to eliminate a portion of the undesirable email stream before it reaches users mailboxes. However, such global approaches run the risk of eliminating some good mail as well (after all what is spam to one person may be desirable advertising to another), while at the same time being more vulnerable to dedicated attacks, e.g., of the “good word” variety [1].

One effective approach to address these shortcomings is to perform email filtering on a personal level, which allows each user to define the spam/non-spam boundary based on a set of examples specific to that user. And indeed personal filters have shown to perform very well both in research and field studies, and numerous implementations, both commercial and non-commercial are available. However, such solutions are deployed typically on the email client (i.e., the personal computer belonging to the end user), which is very flexible but has certain undesirable effects, both for the user and for the ESP. The ESP still needs to accept and store the messages till they are fetched by the client, while the users may get

inconsistent spam filtering performance based on the client used to access their mailbox (e.g., personal computer vs. a mobile device). On the other hand, hosting the personal filters by the ESP itself is often viewed as challenging, since it poses storage and computational requirements of its own.

AOL has been one of the ESPs who decided to face the personal spam filtering challenge. While this paper does not discuss the AOL’s solution, it discusses the nature of the challenges involved when serving a user population ranging in tens of million. In particular, we draw attention to the multi-faceted nature of the challenges, including the need to balance the increase in spam detection with the costs associated with building the additional spam filtering infrastructure.

The paper is organized as follows. In Section 2 we briefly review various approaches to filtering spam, emphasizing the ones which have been successful in performing filtering on a personal level. In Section 3 we outline and categorize the major challenges involved in designing and implementing a large scale personal filtering complex. The focus of Section 4 is on the scaling of implementation costs with the number of users, while in Section 5 we combine the costs and benefits within a single framework and analyze situations where implementing filtering systems is actually beneficial from the ESP’s perspective. The paper is concluded in Section 6.

II. SPAM FILTERING: A BRIEF OVERVIEW

The problem of spam (also known as junk email or: *Unsolicited Commercial Email (UCE)*) filtering has been receiving increasing attention from the commercial and research communities [2]. The approaches taken can be roughly divided into those based on sender/origin reputation, those based on detecting forgery via header/transmission analysis and those based on the analysis of message content. The first two approaches have been popular since the spam problem became apparent, but because of the inadequacies of the email standards they alone are often not sufficient to provide satisfactory levels of spam detection. Content-based approaches to spam detection are more recent and have been particularly popular in the research community since they allow to extend the available body of results in the areas of semi-structured and unstructured document classification to the spam filtering domain [3]. A variety of machine learning techniques have been found effective, including Naive Bayes [4], Maximum Entropy, SVMs [5] and boosting [6]. Variants of Naive Bayes [7] have been particularly popular in practical implementations

due to their simplicity and straightforwardness of incremental model update.

Traditionally, filtering has been performed by ESPs on a global (i.e., user-independent) level, by taking a view that most people agree with what is spam and what is not. While this may be true for many types of spam, it is not necessarily universal. Messages that some users consider unwelcome may be looked upon by others as desirable special offers. Also, for many types of commercial messages (sometimes referred to as *bulk email*) only the recipient is able to determine (and even then it may not be straightforward to do so) whether or not the message was actually solicited. Thus, inherently, personal spam filtering can offer superior filtering accuracy, assuming the direct or indirect user feedback (e.g., sufficient numbers of examples for spam and non-spam) can be obtained to guide the filtering process. The personal definition of spam is typically expressed in terms of content-specific features (e.g., words or other lexical attributes) and the complexity of the function separating spam from non-spam (also referred to as *legitimate email* or *ham*) is highly user dependent.

In practice, seldom one filtering technique is applied on its own, and large filtering systems can often be visualized as a pipeline, whose various stages can have the power of declaring a message as spam [8]. Within such a framework, personalized filtering can be viewed as one of the final stages of the filtering pipeline.

III. CHALLENGES OF LARGE-SCALE PERSONAL SPAM FILTERING

A. Parsimonious model representation

Many successful personal spam filter implementations build classification models based upon hundreds of thousand of features (e.g., [9][10]). Yet such solutions may be costly in terms of the filter memory footprint if deployed for a large population of users. Depending on the type of features and the classification algorithm, evaluation of such models may also be costly in terms of the evaluation time. It is therefore advantageous to consider solutions which are parsimonious in the number of model parameters without adversely affecting filtering accuracy.

B. Efficient and robust model update

Depending on the type of the underlying learner, the classification model for each user can be updated whenever the filter makes an error or nearly so or, alternatively, regardless of the correctness of the current model but such that a representative sample of the user's spam and ham emails is taken into account. Obtaining correct class labels directly from users may be difficult and error prone. On the other hand, deriving such labels implicitly may also result in an error, however. Regardless of how this is done, though, updating the current user model and ensuring that the changes are propagated in a timely fashion can pose quite a few engineering challenges. Additionally, depending on the type of learner used, updating of a model may or may not require the existence of some historical data, and if this is the case then maintenance of

such data needs to be carefully considered as well (i.e., how much to store).

C. Integration with user-independent models

Integration of user-specific models with ones that are serving the overall population depends on the types of the models involved. Typically, a large scale spam detection complex will employ a variety of techniques, each having different error characteristics. Successful combination of such models needs to take the strength and weaknesses of the models involved into account.

D. Real time evaluation for a large population of users

Efficient evaluation and coexistence of models corresponding to a large fraction of the user population can be challenging both from the machine learning perspective of and from the system architecture point of view. In the case of the latter, issues related to keeping up with a growing population of users are of particular importance. Ideally, a model corresponding to every user should be available in memory for immediate evaluation. In practice, that might not be necessary since many of the users may receive email rather infrequently, which suggests that some form of caching might be an attractive option.

IV. IMPLEMENTATION COST SCALING

Out of the various challenges involved, the one having the most impact on the overall system costs is the need to keep a portion of user models in memory for fast evaluation. In principle, the trained user models could reside on disk or in the database and be fetched as needed, but to keep up with a high volume of incoming mail the cost of I/O (in terms of time) might be too high. We will therefore consider the resource impact of maintaining at least a fraction of personal spam filters resident in memory. We will assume that the overall email filtering and delivery system is already capable with the load of incoming email and the main consideration is its extension to incorporate the personal filters. Let us define the following:-

- V - the daily volume of emails handled by the system.
- $size_{msg}$ - average size of an email message in bytes.
- $recip_{cnt}$ - average number of recipients of a single message (where personal spam filters for recipients exist).
- $size_{flt}$ - average size of a memory-resident user filter in bytes.
- N - number of users for which memory-resident filters are being held in RAM.

A. Bandwidth

The system is already assumed to be able to handle the volume of incoming email. However, because the expected recipient count per message is greater than one and because each of the recipients is assumed to possess a distinct filter, the overall bandwidth needs to be capable of having the capacity to support at least

$$V \cdot recip_{cnt} \cdot size_{msg} \text{ (bytes/day)}$$

as opposed to the original bandwidth of

$$V \cdot size_{msg} \text{ (bytes/day)}$$

Fortunately, typically the average recipient count per message is only marginally greater than 1, which makes the importance of the bandwidth considerations rather small.

B. Memory footprint and box/unit count

Ignoring the (important) details of how personal filter evaluation is actually implemented, at the high level it can be viewed as a distributed computing complex requiring a certain amount of RAM to hold the personal spam filters corresponding to N users. The cost of provisioning such a distributed system can be understood in terms of the number of computing units (e.g., Linux “boxes”) needed to house the amount of memory required. Other resource requirements, such as CPU power and disk storage will also have an impact but can be considered as secondary.

Given the constraints placed by the operating systems on the maximum amount of physical memory that can be handled by a single box, as well as the constraints of the maximum amount of memory addressable by a single process, the memory footprint translates itself into a certain number of physical boxes (or rack-mount units) that need to be maintained in order to keep the system operational. This number will depend on the type of the operating system, the number CPUs and network connectivity of a single unit. Assuming that a single unit contains RAM_{unit} GB of physical memory (e.g., 4, 8, 16 or 32GB) let us consider the dependence of the number of units required with on number of users for which memory resident filters are kept (N) and the average size of average filter ($size_{flt}$). The unit count can be estimated as:

$$cnt_{unit} = \frac{N \cdot size_{flt}}{RAM_{unit}}$$

For example, when $RAM_{unit}=8$ GB with 10 million users and with the profile size of 100 KB per user (a hypothetical value), the number of physical units needed would be 125, which goes down to 63 assuming 16GB per box. Assuming \$10,000 per box, the estimated cost of the system would be over \$500,000. These numbers can of course vary and are provided only for illustrative purposes, but clearly, reducing the size of a filter profile can have a significant impact on the overall cost of the system.

In practice, not all users receive emails with the same frequency and while some may get hundreds or thousands messages per day, others will receive just a few or none at all. This suggests that, in any time interval, the stream of incoming email needs to be evaluated only against a fraction of the overall filter population. A caching system for user filters is therefore likely to be beneficial in reducing the resource requirements. Depending on whether caching is used, N will correspond to the overall user population or to the fraction that is being cached.

V. COSTS VS. BENEFITS

At which point does it make sense to justify the expenditure necessary to implement a personal spam filtering complex? From the point of view of providing users with a consistent (and better) performance, it is a business decision that is hard to quantify. Within the cost-sensitive filtering context, the reduction of the false positive (i.e., spam) and/or false negative (i.e., not-spam) error rates needs to outweigh the cost of putting the system together (amortized over some period of time). When supporting a very large population of users the challenge presents itself not only in terms of performing computations on a large scale but also in terms of the implementations costs attached to it. Clearly, there needs to be a perceived economic advantage for implementing any such large-scale system.

The cost consideration can be quite complex. Not only may the different types of errors (i.e., false positives vs. false negatives) be weighted differently but also, within each class of errors, there may be a non-uniform distribution of misclassification costs depending on message content (e.g., personal emails vs. advertising emails) [11] and the recipient of the message. Exact cost values are hard to obtain, especially for the false-spam category. We will initially take the simplistic assumption that personal filtering will not introduce any additional false-spam errors, and focus on the increased spam detection rate provided. The benefit of implementing such a personalized spam filtering complex would thus manifest itself in both increased user satisfaction and reduced storage costs, assuming that messages classified as spam have only a limited lifetime. We will ignore the monetary value of increased user satisfaction and focus on the savings resulting from fewer messages needed to be stored by the ESP.

Let V_s be the average number of spam messages (undetected by other means) penetrating the existing system on a daily basis. Let TS be defined as the fraction of spam correctly identified by the personal spam filtering complex. The daily savings resulting from having the complex deployed can be expressed as:

$$savings_{day} = V_s \cdot TS \cdot size_{msg} \cdot cost_{store}$$

where $cost_{store}$ is the cost of storing a byte of data. The above translates for example to the annual savings of

$$savings_{year} = 365 \cdot savings_{day}$$

or can be used in calculating the savings over other time intervals (assuming that the values of V_s , TS , $size_{msg}$ and $cost_{store}$ remain invariant over that time). For example, in a system handling 10 spam million messages on a daily basis and being able to detect 70% of them, the annual savings amount to approx. \$38,000 assuming $size_{msg} = 15$ KB and that the cost of storing 1GB of message data is one dollar. Of course, these values are just hypothetical and should be substituted with ones that are meaningful to a particular system.

Note that while the cost of implementing the system is driven by the need to support the total number of users N

and the full volume of incoming mail, the benefit is dependent on the amount of spam able to penetrate the original spam-detection system. Thus when large numbers of users are involved, and in cases where the original system was already quite accurate, the cost of implementing the personalized spam filtering may outweigh its benefits (at least in terms of savings in storage). This is because the cost of implementing the system scales primarily in the number of users, while its benefit scales in terms of the number of spam messages detected. In principle, one might assume that the number of spam messages penetrating system defences is proportional to the size of the user population. However, it is often the case that most of the spam is targeting a sub-population of the larger user population, in which case the dependence of V_s on N becomes sub-linear. In such situations the ratio of the deployment cost to the benefits effected will grow with N , and at some point the cost might outweigh the benefit.

A. The cost sensitive classification framework

The analysis above assumed that the personal filters do not introduce errors of the false-spam type. This may be not the case however and one needs to look at the costs vs. benefits argument within the context of the cost sensitive classification framework [12]. Generally, costs are attached to making the mistakes of the false-spam and false-legitimate type and one needs to consider them in conjunction with the costs of building and operating the spam filtering complex. Such an analysis can be performed at any stage of the email filtering pipeline. Here we assume specifically that the input state to the analysis consists of the email stream that has so far been classified as legitimate. The spam contamination rate π denotes the fraction messages in the stream that are actually spam (we will assume this fraction to be fixed). Depending on what other filtering mechanisms were deployed upstream, this fraction can range from as high as 0.7 [13] to 0. Prior to implementing a filtering system all of the spam messages remain undetected, so their cost to the system is

$$C_{before} = V \cdot \pi$$

assuming that the cost of not detecting a spam message is unity (this combines the cost of storage with the cost incurred by the users for having to deal with spam). Once a system characterized by the false-legitimate rate of FL and the false-spam rate of FS is put in place, the system cost is transformed to

$$C_{after} = V \cdot (\pi \cdot FL + \text{cost} \cdot (1 - \pi) \cdot FS) + \text{imp}$$

where imp is the one-time cost of implementing the system. Thus the gain resulting from spam filtering can be expressed as:

$$\text{gain} = \frac{C_{before}}{C_{after}} = \frac{1}{FL + \text{cost} (1/\pi - 1) FS + \frac{\text{imp}}{V \cdot \pi}} \quad (1)$$

Assuming a constant user population (i.e., $\text{imp} = \text{const}$) and a growing volume of mail to be filtered V , the initial cost of implementing the system will eventually be marginalized

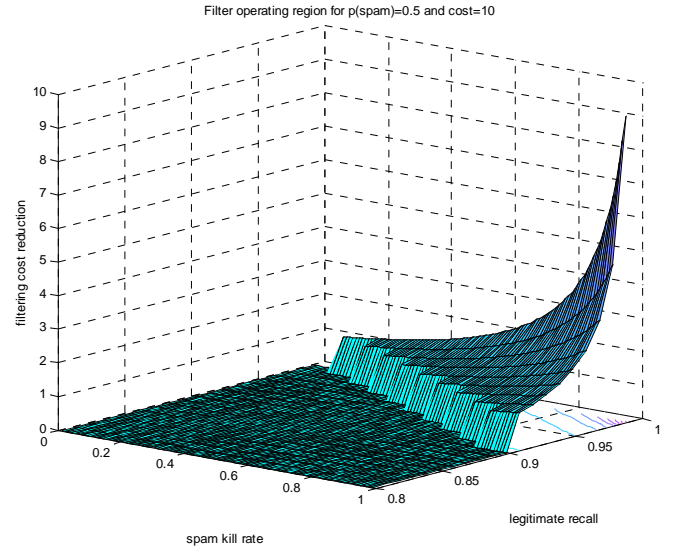


Fig. 1. Acceptable performance region for a system where the spam contamination level is 50% and the relative cost of a false-spam mistake exceeds 10 times the cost of admitting spam. For better clarity, the regions for which gain is less than 1 have been set to 0.

and whether or not spam filtering is beneficial will depend solely on the prevalence of spam (π), the accuracy of the filtering system (FL and FS) and the relative cost of false-spam mistakes (cost). Note that for the system to be beneficial it is required that $\text{gain} > 1$. As illustrated in Figure 1, for a given prevalence of spam and the value of cost , only filters of certain accuracy offer performance that can actually be considered as beneficial, where the region of acceptability will shrink with growing values cost of and decreasing values of π .

However, the implementation costs cannot always be ignored and, realistically, we are likely to be interested in obtaining a net benefit within a certain time (e.g., 1 year) since deployment. To use formula (1), V will then denote the volume of mail filtered within the time period of interest. Note that for gain to be greater than 1, none of the terms in the denominator of (1) can be greater than one and therefore the implementation cost cannot exceed the cost of not filtering at all. As illustrated in Figure 2, greater values of $\frac{\text{imp}}{V \cdot \pi}$ place more restrictions on the quality of the filters. In particular, given that generally the misclassification of legitimate email as spam is much more expensive than the opposite, a useful system needs to have a very low rate of the such error (i.e., low FS). Quite intuitively, the more expensive the system (when compared to doing no filtering), the more stringent requirements need to be placed on the highest allowable rate of false-spam misclassifications. Note that Figure 2 uses particular values of cost and π that would need to be substituted with values that make sense in a particular system. Depending on these settings, the boundary of system usefulness will lie closer to or farther away from the origin, but the qualitative picture remains the same.

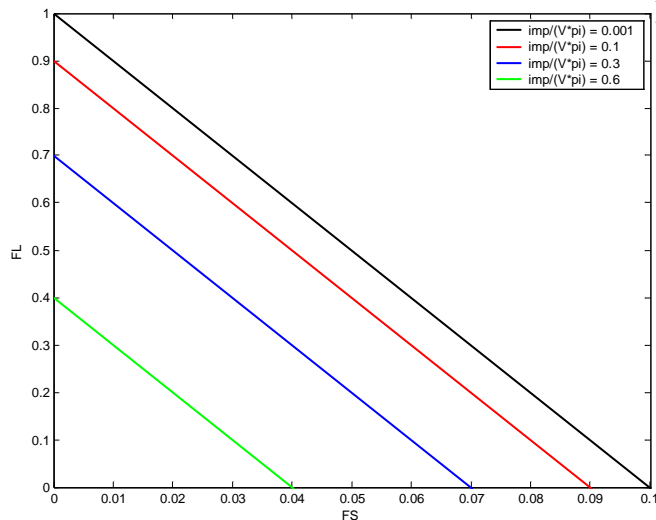


Fig. 2. Boundaries of usefulness of spam filtering in a setting where $cost = 10$ and $\pi = 0.5$. Several ratios of implementations costs to no-filtering costs are considered where the region of beneficial performance is located in the region between the boundary line and the origin. Note that more expensive systems need also to be more accurate, especially in terms of avoiding false-spam misclassifications.

VI. CONCLUSIONS

Spam filtering remains a challenging problem for ESPs and end users. When handled on a large scale one has to trade off the benefits stemming from increased spam filtering accuracy with the costs involved in implementing a more accurate system. In particular, personal spam filtering allows users to provide their own definition of spam, yet when implemented as a service the cost of operating such a filtering complex for a large population of users can outweigh the benefits if the implementation is not careful enough. Factors such as model parsimony can play a critical role in making the implementation economically viable (although here the challenge lies in maintaining high detection accuracy despite model parsimony), especially when faced with a growing population of users and coexistence of many different spam detection mechanisms. In general, the benefits of spam filtering depend on the prevalence of spam, the balance of false-spam and false-ham mistakes of the spam filters and the balance of costs associated with these mistakes. In situations where the level of spam attack is low, the benefit provided to the users can actually be negative (due to the non-zero probability of misclassifying legitimate mail as spam). Also, when prevalence of spam is low and/or a large enough fraction thereof is detected by user-independent techniques, the implementation costs associated with the personal spam filtering service may outweigh the benefits resulting from increased detection rate and reduced storage costs. On the other hand, in situations where spam prevalence is high and a significant fraction of spam remains undetected by user independent techniques,

implementation of a personalized spam filtering complex may well be worth the effort.

REFERENCES

- [1] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in *Proceedings of the First Conference on E-mail and Anti-Spam*, 2005.
- [2] T. Fawcett, "'In vivo' spam filtering: A challenge problem for data mining," *KDD Explorations*, vol. 5, no. 2, pp. 203–231, 2003.
- [3] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," in *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998.
- [4] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering," in *Proceedings of the Workshop on Machine Learning in the New Information Age: 11th European Conference on Machine Learning (ECML 2000)*, 2000, pp. 9–17.
- [5] H. Drucker, D. Wu, and V. Vapnik, "Support Vector Machines for Spam Categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.
- [6] X. Carreras and L. Márquez, "Boosting trees for anti-spam email filtering," in *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigris Chark, BG, 2001.
- [7] D. D. Lewis, "Naive (Bayes) at forty: the independence assumption in information retrieval," in *Proceedings of the 10th European Conference on Machine Learning*, 1998, pp. 4–15.
- [8] R. Segal, J. Crawford, J. Kephart, and B. Leiba, "Spamguru: An enterprise anti-spam filtering system," in *Proceedings of the First Conference on E-mail and Anti-Spam*, 2004.
- [9] W. Yezounis, "Sparse binary polynomial hashing and the CRM114 discriminator," in *MIT Spam Conference*, 2003.
- [10] S. Chhabra, W. Yezounis, and C. Siefkes, "Spam filtering using a markov random field model with variable weighting schemas," in *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, 2004.
- [11] A. Kolcz and J. Alspector, "SVM-based filtering of e-mail spam with content-specific misclassification costs," in *Proceedings of the Workshop on Text Mining (TextDM'2001)*, 2001.
- [12] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001, pp. 973–978.
- [13] "MessageLabs intelligence annual email security report," MessageLabs, Tech. Rep., 2004.