SVM-based Filtering of E-mail Spam with Content-specific Misclassification Costs

Aleksander Kołcz* Joshua Alspector†

Abstract

We address the problem of separating legitimate emails from uncolicited ones in the context of a large-scale operation, where the diversity of user accounts is very high, while misclassification costs are content-dependent and highly asymmetric. A category specific cost model is proposed and several effective methods of training a cost sensitive filter are studied, using a Support Vector Machine (SVM) as the base classifier. Clear benefits of explictly accounting for varied misclassification costs, either during training or as a form of post-processing, are shown.

1 Introduction

Large-scale e-mail filtering, as performed by hosting services and large IT departments[1], is a challenging problem that embodies many of the most difficult tasks of machine learning: non-stationarity of the data source, severe sampling bias in the training data, and non-uniformity of misclassification costs—with the true value of costs largely unknown. In contrast, most research in the area of spam detection to date has focused on relatively small problems, primarily suited to the task of performing "soft" spam filtering (i.e., without actually rejecting the messages) for individual users' mailboxes. In this work we explicitly address the complexities pertinent to a large scale spam filtering task and discuss training and evaluating spam filters in a cost sensitive environment. We focus on the use of Support Vector Machines (SVMs) [24] as our base classifier, since their demonstrated robustness and ability to handle large feature spaces makes them particularly attractive for this task. A number of different approaches to creating cost-sensitive spam filters based on SVMs are studied and their performance is evaluated on a corpus of diversified email data. In particular, we examine the merits of training SVMs with instance-specific misclassification costs against the typical approach of using only class-specific ones. Advantages of directly accounting for content-specific misclassification costs are demonstrated.

2 Machine learning for filtering spam

Spam filtering based on the textual content of email messages can be seen as a special case of text categorization, with the categories being spam and non-spam. Although the task of text categorization has been researched extensively, its particular application to email data and, especially, detection of spam is relatively recent.

*Personalogy, Inc., 24 South Weber, Ste. 325, Colorado Springs, CO 80903. email: ark@personalogy.net

†Personalogy, Inc., 24 South Weber, Ste. 325, Colorado Springs, CO 80903. email: josh@personalogy.net

[1] Throughout the paper we will use the term E-mail Service Provider (ESP) to describe the organization responsible for delivering and filtering email to a large and diverse group of users

The first research studies primarily focused on the problem of filtering spam were those of Sahami *et al.* [22] and Drucker *et al.* [10]. In [22], the authors applied Naive Bayes (NB) to the problem of building a personal spam filter. NB was advocated due to its previously demonstrated robustness in the text-classification domain, and due to its ability to be easily implemented in a cost-sensitive decision framework. Although high performance levels were achieved using word features only, it was observed that by additionally incorporating non-textual features and some domain knowledge, the filtering performance can be improved significantly. Application of SVMs to the spam-filtering task was also suggested.

The validity of SVMs' effectiveness in spam detection was verified by Drucker *et al.* [10], who compared SVMs with RIPPER [8], a TF-IDF based classifier and a boosted ensemble of C4.5 trees. Both SVMs (especially when using binary features only) and boosted trees performed very well on the dataset used, but SVMs proved to be much faster and had a preferable distribution of errors. Additionally, the authors noted that SVMs can easily handle large feature spaces (such as text) thus being able to bypass the expensive feature selection process.

In a series of papers, Androutsopoulos *et al.* [1][2][3] extended the NB filter proposed in [22], by investigating the effect of different number of features and training-set sizes on the filter's performance. The accuracy of the NB filter was shown to greatly outperform a typical keyword-based filter [2], while being comparable with a k-nn based filter [3]. NB and k-nn were also successfully combined in a stacking framework [23]. The need for cost-sensitive spam filtering was advocated in [3][12].

Other spam-filtering studies include `SpamCop` [18], where NB was found to outperform RIPPER , as well as the work of Katirai [14], where NB was marginally better than a genetic programming approach.

2.1 The choice of a base classifier

All in all, different machine learning algorithms can be reasonably effective at the spam-filtering task. Many researchers obtained good results with NB: a simple Bayesian classifier assuming mutual independence of all features. Potential disadvantages of using NB in an ESP setting is the need for careful feature selection. When the number of features grows beyond a few hundred its performance tends to deteriorate, and the class-membership probabilities produced approach either 0 or 1 [4]. At the same time, in order to capture the variability of messages (potentially multi-lingual) handled by an ESP the use of large feature spaces is likely to be necessary. We focused on the use of SVMs, which are currently placed among of the best-performing classifiers and have a unique ability to handle extremely large feature spaces (such as text), precisely the area where most of the traditional techniques fail due to the "curse of the dimensionality". Apart from their encouraging performance in the spam-filtering domain [10], SVMs have been shown to be very effective in the field of text categorization [13][11], which we will utilize in this study.

3 Construction of an optimal classifier

To most users spam is a nuisance, while the loss of legitimate email is much more serious, so the optimal filter is defined as one that rejects a maximum amount of spam while passing all legitimate emails. In other words, the ideal filter maximizes spam recall while maintaining a 100% spam filtering precision. In practice, we seek to minimize the expected cost (or loss) due to the filter's erroneous decisions.

Let $\mathcal{X}$ be the space of all possible emails, where each message is assigned a label of *spam*

or *legit* (we will assume that the labels *spam* and *legit* are numerically equivalent to -1 and +1, respectively) according to the joint probability distribution $P(X,Y)$, where $X \in \mathcal{X}$ and $Y \in \{spam, legit\}$. Let $C_{S\to L}(X)$ denote the cost incurred by the system when a spam message X is classified as legitimate and, conversely, let $C_{L\to S}(X)$ denote the cost incurred by the system when a legitimate message X is classified as spam. We assume that there is no cost associated with a correct classification of a message, i.e., $C_{L\to L}(X) = C_{S\to S}(X) = 0$. Then, the expected costs of misclassifying X as *spam* or *legit* are given by

$$\begin{aligned} C_{legit}(X) &= C_{S\to L}(X)P(spam|X) \\ C_{spam}(X) &= C_{L\to S}(X)P(legit|X) \end{aligned} \tag{1}$$

and an optimal classifier F should minimize the expected cost[2] for each X, i.e.,

$$F : \frac{C_{spam}(X)}{C_{legit}(X)} \geq 1 \;\rightarrow\; \begin{cases} true \;\Rightarrow legit \\ false \;\Rightarrow spam \end{cases} \tag{3}$$

According to (2) an optimal classifier should have access to accurate posterior probability and misclassification-cost estimates. One can then directly assign class labels minimizing the expected loss. Zadrozny and Elkan [26] use the term *direct cost-sensitive decision making* to describe such a process. Not all learners directly produce posterior probability estimates or have provisions for cost-sensitive learning. Recently, Domingos proposed MetaCost [9], a framework in which any generic classification scheme can be made cost sensitive. He noted that the labels assigned to the elements of the training set, even if correct, may not be optimal when misclassification costs are taken into account. It was observed that bagging [5] can be used do estimate the $P(j|X)$ probabilities for any learner, upon which the training points are re-labeled so as to minimize the expected cost over the training data. Such a modified training set is then used for training the particular learning algorithm chosen. Since most models minimize the overall error rate, re-labeling the elements of the training set increases the relative frequency of examples whose misclassification is particularly costly.

4 The nature of email misclassification costs

4.1 Cost of misclassifying spam as legitimate ($C_{S\to L}$)

From an ESP perspective, failure to block spam results in extra storage costs, which may become quite substantial considering that for some ESPs, 30% of incoming messages are spam, and the number of messages processed daily ranges in millions. Given some basic statistics about the average size of a message, one can easily specify the nominal cost of admitting a spam message into the system. Another cost of failing to detect spam relates to the quality of service (QoS) issue and is more difficult to quantify. Generally, the heavily spammed users are more likely to complain or abandon the service altogether. All in all, though, the cost of admitting a spam message may be considered fairly nominal (at least within a certain time window) and, in this work, we will use the unit cost, i.e., $C_{S\to L} = 1$.

[2] In a general N-class problem, an optimal classifier F assigns class i to point X, such that

$$F(X) = \min_i \sum_j C_{j\to i}(X)P(j|X) \tag{2}$$

where $C_{j\to i}(X)$ is the cost of misclassifying X as a member of class i when in fact it belongs to class j; $P(j|X)$ is the conditional probability of class j given message X. Thus, for each point, a cost matrix is defined, with elements $C_{j\to i}(X)$, $i, j \in \{1, .., N\}$.

4.2 Cost of misclassifying legitimate email as spam ($C_{L\to S}$)

The consequences of losing an important message may potentially be devastating, so it is important that, from the QoS perspective, rejection of legitimate email remains a truly rare event.

Most research in spam filtering assigns a uniform cost value to $C_{L\to S}$. It is clear, however, that the true costs are likely to be dependent the type of the message rejected (and other factors, such as the addressee), and in this work we concentrate on the content-specific misclassification costs. Generally, for any legitimate message X, its misclassification cost is given by $C_{L\to S}(X)$. However, since exact measurement of such a cost is very difficult, considering truly message-specific costs is largely impractical. It is possible, though, to estimate costs associated with misclassifying legitimate messages belonging to several broad content categories:

- Sensitive personal messages: misclassifying this type of email could be potentially most harmful, and any filtering system should strive to reduce these misclassification incidents essentially to zero. Our *a priori* setting is $C_{L\to S}(X|X \in personal) = 1000$.
- Business related messages: these are emails exchanged between co-workers, current and/or potential business partners, etc. The cost of losing business related emails is certainly high and, depending on the service provider's business model, it may be higher or lower with relation to the loss of personal messages. We consider the case of a typical ESP, where most users are private individuals, in which case the loss of personal messages should be higher. Our *a priori* setting is $C_{L\to S}(X|X \in business) = 500$.
- e-commerce related messages: These messages include registration/order/shipment confirmations and/or reminders and are an important factor in ensuring user confidence in e-commerce and web services. Our *a priori* setting is $C_{L\to S}(X|X \in e-commerce) = 100$.
- special interest mailing lists/discussion forums: Many users subscribe to a variety of information sources, ranging from news, through professional interests to hobbies and adult interests. Our *a priori* setting is $C_{L\to S}(X|X \in list) = 50$.
- promotional offers: As a price for providing relevant content, many web sites reserve the right to contact their visitors regarding various promotions and/or commercial offers. Such messages are also regularly received from e-commerce companies with which a user had already done business before. Our *a priori* setting is $C_{L\to S}(X|X \in promotion) = 25$.

The categories $\mathcal{C} = \{personal, ..., promotional\}$ outlined above are by no means exhaustive, and their relative frequency and importance is likely to change over time. We believe, however, that they capture the basic usage of email today and the importance of weighing the costs of rejecting legitimate email according to content. The use of broad content categories should facilitate the measurement of costs in a real-world environment.

Assuming that the proposed sub-categories are mutually exclusive and that they completely cover the legitimate email class, the cost of misclassifying a message as spam (see eq. (1)) can be re-stated as:

$$C_{spam}(X) = P(legit|X) \sum_{cat\in\mathcal{C}} C^{cat}_{L\to S} P(cat|legit, X) \tag{4}$$

5 Cost-sensitive training of an SVM-based spam filter with imprecise cost and probability estimates

5.1 Direct cost-sensitive SVM training

Given a T-element training set $\left\{(x^i, y^i) : x^i \in \Re^D,\, y^i \in \{-1, 1\};\; i = 1, ..., T\right\}$ a linear SVM classifier (the non-linear case will not be considered here – see [24] for details)

$$F(x) = \sum_i \alpha_i y^i x \bullet x^i + b = x \bullet \sum_i \alpha_i y^i x^i + b = x \bullet w + b \tag{5}$$

where $\alpha_i \geq 0$, b is a bias term and D is the dimension of the input space; $\bullet$ is a dot-product operator, and w is the normal vector of the classification hyperplane. Typically, the multipliers α_i have non-zero values only for a small subset of the training set, which is called the support set and its elements the *support vectors* [24]. The optimal hyperplane is found such as to maximize the classification margin, given by $2/\left\|w\right\|^2$, where w denotes the normal vector of the hyperplane. The *soft-margin* optimization task is formulated as:

$$\begin{aligned} \text{minimize} \quad &: \quad \sum_i C_i \xi_i^p + \frac{1}{2}\left\|w\right\|^2 \\ \text{subject to:} \quad &\quad y(w \bullet x + b) \geq 1 - \xi_i \end{aligned} \tag{6}$$

where $C_i \geq 0$, $p \geq 1$ and $\xi_i = (1 - y(w \cdot x + b))_+$. ; $(z)_+ = z$ for $z \geq 0$ and is equal to 0 otherwise; The slack variables ξ_i take non-zero values only for bound support vectors, i.e., points that are misclassified or lie inside the classifier's margin. Note that in eq. (6) the accuracy over the training set is balanced by the "smoothness" of the solution. We will consider the case of $p = 1$, for which a number of highly efficient computational methods have been developed (e.g., [19]).

When the classification problem is linearly separable, it is possible to find a solution (by setting all soft margin constraints sufficiently high, i.e., $C_i \geq C$ for some value of C) for which all slack variables are zero, i.e., $\sum_i C_i \xi_i = 0$ [15], but this may not be desirable from the generalization point of view, especially when misclassification costs are varied. This observation is particularly relevant in the textual domain in which, due to the high dimensionality of text patterns, training points corresponding to different classes very often are linearly separable. Thus, although the soft-margin method has been developed for cases where training points are not linearly separable in the kernel-induced space, the approach is generally useful in cases where the training data are noisy and the misclassification costs differ.

In most published results, (6) has been considered either with $C_i = C$, for all i, or with C_i being class specific ([17]), i.e.,

$$C_i = \begin{cases} C^+ & \text{if } y^i = +1 \\ C^- & \text{if } y^i = -1 \end{cases} \tag{7}$$

The latter case still involves finding just one parameter, since the ratio C^+/C^- is constant for all i.

In this work we extend the approach represented by (7). Unlike other examples of applying asymmetric misclassification costs to SVM training (e.g., [17][7]), the costs are allowed here to be instance specific, rather than just class-specific. Recall that in the proposed cost model, $C_{S \to L} = 1$, while $C_{L \to S}$ depends on the type of the legitimate

message. Consequently, with $C_{S\to L}$ as a reference, each type of $C_{L\to S}$ can be expressed as a fixed multiple thereof. The misclassification penalty term in (6) is then

$$\sum_i C_i\xi_i = C^- \sum_{cat\in\mathcal{C}} C^{cat}_{L\to S} \sum_{i:x_i\in legit\,\wedge\, cat} \xi_i \tag{8}$$

The value of C^- in (8) is not a cost but, rather, a regularization parameter that has to be chosen so that the expected performance of the classifier is maximized. Note that such a formulation can be easily extended to any problem where the misclassification cost for one of the classes is constant, while being variable for the other.

5.2 Cost-sensitive SVM training in a probabilistic framework

5.2.1 Probabilistic calibration of SVM outputs Although a trained SVM does not automatically produce posterior probabilities, its outputs can be normalized for that purpose. For any given point, the SVM returns the point's signed distance from the classification hyperplane, where points lying farther away from it have a higher chance of being classified correctly. The relationship between the distance and the confidence in the classification outcome is not strict but may be, empirically, determined to a good approximation. Platt [20] has proposed an effective method of normalizing the output of a trained SVM, via a sigmoid transformation, which we will adopt in this work.

5.2.2 Cost-sensitive training via MetaCost [9] MetaCost (as outlined in Section 3) represents an attractive framework for cost-sensitive training of classifiers, which requires obtaining accurate estimates of $P(j|x)$ for elements of the *training set*. To avoid overfitting, Domingos [9] suggested bagging [5], where B (e.g., $B = 10$) bootstrap samples of the original training set are drawn, and each one is used to train a separate SVM classifier. Subsequently, the estimate $P(j = 1|x)$ for a training point x is computed using "out-of-bag" (OOB) [6] averaging. After MetaCost adjustment of training-set labels, a regular SVM is trained on the modified training set with a uniform soft-margin constraint, i.e., $C_i = C$ for all i in (6).

5.2.3 Achieving cost-sensitivity via categorization As discussed in Section 4.1, $C_{S\to L}$ may be reasonably assumed to be message independent, while $C_{L\to S}$ depends on the category of the legitimate message and is assumed to be fixed for all messages in that category. Therefore, the cost-sensitive decision process depends on estimating $P(legit|x)$, and $P(cat|x, legit)$ for all cat, so that the expected penalty for misclassification can be computed using (1) and (4). To that end, $N+1$ two-class SVM models are created: one to estimate $P(legit|x)$, and the rest to distinguish between the N categories of the legitimate class.

6 Cost-sensitive evaluation of classification performance

Traditionally, evaluation of classification experiments has been carried out by simply calculating the error rate for elements in the test set $\left\{(x^i, y^i) : i = 1, ..., S\right\}$, i.e., $Err_F = \frac{1}{S}\sum_i \left[F\left(x^i\right) \neq y^i\right]$. In most cost sensitive classification research, it has been assumed that the evaluation sample to which a trained classifier is applied is distributed according to the target distribution and that the misclassification costs for elements of that sample can be accurately measured, in which case the average normalized cost over the test sample can be used as a good performance measure, i.e., $NCost_F = Cost_F / \sum_i \max_j C_{y^i\to j}$, where $Cost_F = \sum_i C_{y^i\to F(x^i)}$ and $C_{y^i\to F(x^i)}$ is the cost of misclassifying the ith test point, which belongs to class y^i as a member of class $F\left(x^i\right)$. In [16] it was recently pointed out that such

TABLE 1
Distribution of the legitimate portion of the dataset.

category	message count	$P(category\|legit)$
private	316	0.052
business	655	0.108
e-commerce	162	0.027
special	3891	0.644
promo	1019	0.169

a performance measure can be misleading, since in situations where misclassification costs are varied and there is high imbalance in the distribution of different classes, even a fair test sample may contain very few elements of the rare classes for which the misclassification cost is very high. *BCost*, an effective bootstrap-based method of estimating the confidence interval for $Cost_F$ was proposed in [16].

For the spam filtering application, Androutsopoulos *et al.* [2] proposed to use a ratio of the cost of a given classifier to the cost incurred when no filtering is used (i.e., all messages are classified as legitimate). This measure, termed *True Cost Ratio* (TCR) is defined as

$$TCR_F = \frac{\sum_i [y^i = spam] \cdot C_{S \to L}}{\sum_i C_{y^i \to F(x^i)}} \tag{9}$$

and will be used in here.

7 Experimental Setup

7.1 Data acquisition

Despite a number of published studies in the area of spam detection, few standard benchmark data sets have been defined. The `SpamBase` corpus available from the UCI repository uses a very restricted feature set (48 words only) and thus does not capture the variability we expect to find in an email stream. The `LingSpam` corpus [1] contains a reasonable collection of spam, but its legitimate section consists of emails from only one specialized mailing list. The `PU1` collection [2] does improve on the variety of legitimate email sources, but the contents of messages are encoded, which makes it impossible to differentiate the costs of misclassifying different types of legitimate email. In order to properly illustrate the problems discussed in this study, a sizable diversified collection of messages was therefore acquired, with the legitimate emails assigned to the categories considered [3].

The spam portion of the data was gathered from a variety of sources, which included: a) publicly available collections; b) spam donated by a number of participants; c) spam acquired via "spam traps"; d) from a spam blackhole-redirect service. The legitimate email collection consisted of: a) emails donated by a group of volunteers; b) emails resulting from subscription to a wide range of mailing lists, including professional/technical interests, promotional opt-in marketing and leisure and adult-interest related forums.

Overall, the data set contained 5,365 (47%) spam messages and 6,043 (53%) legitimate messages. Table 1 details the distribution of message categories in the legitimate portion of the data. 75% of the data were used for training and the rest were used for testing. The split

[3] We intend to make the dataset available at http://www.personalogy.net/reserach/.

was performed at random and was stratified, i.e., for each email category, the proportion of messages of that category in the training/test set was the same as in the original collection. In the following experiments, we assumed that the proportion of each email category in the data set reflected the true target distribution and no additional adjustment of priors was performed.

Given the cost scheme used in this study, achieving low misclassification cost on the test data was quite challenging. In particular, there were 1,508 legitimate test messages and 1,341 spam test messages. So a filter which would classify all spam emails correctly while making just two mistakes in the legitimate class, by misclassifying one *private* and one *business* message, would lead to a cost higher than using no filtering at all!

7.2 Data pre-processing

For each message, two sets of features were extracted. The non-word features were identified first. To this end, a number of condition checks were performed on the header and body components of each message, with the conditions representing our domain knowledge of spammers' tactics. Additional processing extracted real-valued features related to word and character statistics of a message (as inspired by [22])[4]. These were subsequently discretized using the Minimum Description Length principle.

For each email, its textual portion was represented by a concatenation of the subject line and the body of the message. The word-extraction process was preceded by substituting all non-alphanumeric characters with whitespace, where HTML tags were removed first, if present. `Base64`-encoded attachments were decoded only for the `text/plain` and `text/html` content types. Words were defined as contiguous strings of characters delimited by whitespace. All characters were converted to lowercase, but if a word consisted of all capital letters, it was effectively treated as two words: all lowercase all uppercase (as inspired by [10]). Zipf's law was applied to eliminate word features whose frequency in either class was less than 3. Out of 28,132 such word features, 10,000 most relevant ones were chosen by applying the Mutual Information criterion [11]. In principle, all features could have been used in SVM training, but that would have resulted in long computation times.

All messages were represented as binary vectors, with elements corresponding to the presence or absence of words and the true or false status of conditions. For the discretized real-valued attributes, 1-of-N encoding was used to convert them into a binary-string format. We also investigated the influence of normalizing[13] the SVM inputs and considered two general cases: one where inputs were given by standard binary vectors, and one where the initial binary vectors where normalized so that each had the Euclidean norm of one.

7.3 Variants of SVM filters investigated

Based on the discussion of cost-sensitive learning of SVMs presented in Section 5, we considered the following classifier configurations (in all cases the regularization parameter, C or C^-, was chosen according to the Generalized Approximate Cross-Validation (GACV) criterion [25]):

[4] This set included: the ratio of the number of letters to the total number of characters (separately for message subject and body); the ratio of the number all-capital words to the total number of words; the ratio of the number capital letters to the total number of words; the numbers of ! and ? in the message.

Standard SVM (`standard`): All misclassification costs were treated equally. For probabilistic outputs (`prob`), the classification threshold was adjusted via OOB estimation to minimize the misclassification cost for the training set.
Average-cost SVM (`cost-avg`): The average misclassification cost for the legitimate class was used during training to differentiate between the weight of positive and negative errors (see eq. 7).
All-in-one (true cost) SVM (`cost`): Variable misclassification costs were accounted for by appropriately adjusting the cost ratio for each training sample, according to (**??**) as discussed in Section 5.1.
MetaCost followed by distribution-adjusted SVM (`meta`): The initial probabilistic SVM (in 10 bagged incarnations) was used (via OOB) to estimate $P(legit|x)$ and $P(spam|x)$, and the original training sample was relabeled by taking these probability estimates and the misclassification costs into account. A regular SVM was then trained on the relabeled data.
Probabilistic SVM with category-specific models (`cat`): A standard probabilistic SVM was built to estimate $P(legit|x)$. Additionally, the legitimate portion of the training set was used to train of number of probabilistic SVMs: each to distinguish one category of legitimate emails from all others.

8 Results

8.1 Problems with MetaCost and optimal setting of classification thresholds

MetaCost uses bagging to estimate the posterior class-membership probabilities and, given a misclassification cost matrix, it relabels the original training examples so that a minimum-cost decision is taken in each case. The procedure assumes that accurate probability estimates are readily available, or else it uses bagging to estimate them. Unfortunately, when there is a large variation between different misclassification costs, direct application of MetaCost may result in assigning all training-set points to the class for which the misclassification costs are highest. For example, in our case $P(spam|x)$ has to be greater than 0.993 (on average) for a spam message to retain its label during the MetaCost relabeling process. On the other hand, the estimated class-membership probabilities tend *not* to take extreme values, which results in *all* spam examples labeled as legitimate thus rendering the procedure useless. Indeed, in [9] Domingos considered two-class problems with cost ratios no greater than 8.

To overcome this problem, we *estimated* (via OOB) the optimal cost factor (by minimizing the training set cost) for use with MetaCost and when applying the trained classifier to the test data. To this end, we used the OOB estimates of $P(spam|x)$ for the training data to find the "optimum" value $C_{L\to S}$. All training points were ranked according to $P(spam|x)$ and, for each point, we calculated the overall misclassification cost resulting from making the particular value of $P(spam|x)$ the threshold point of the classier. The value resulting in lowest cost was then used to derive the corresponding value of $C_{L\to S}$.

As in the case of MetaCost, we found that naive use of true misclassification costs in the decision-making process leads to highly pessimistic results. The optimal cost factors were found instead by adjusting the classification threshold via 10–fold OOB and cross-validation procedures. This is in contrast to other published studies (e.g., [22][2]), where probabilistic methods for spam filtering use the same cost factors in decision making and in performance measurement, which leads to disappointing results when misclassification costs are highly asymmetric.

TABLE 2

TCR costs and their 0.95 BCost confidence intervals for each of the SVM configurations (see text for details) considered. Results using standard (i.e., not normalized) and normalized binary inputs are shown. All methods of incorporating asymmetric costs into the SVM learning process are seen to outperform the baseline.

configuration	TCR_{std}	TCR_{norm}
standard	0.40 [0.20, 1.17]	0.36 [0.18, 1.04]
cost-avg	1.55 [0.55, 4.72]	3.54 [0.94, 3.87]
cost	1.56 [0.56, 5.10]	1.57 [0.55, 4.88]
meta	1.23 [0.52, 3.44]	2.12 [0.76, 2.57]
prob	1.58 [0.55, 4.95]	4.18 [0.92, 6.24]
cat	1.78 [0.72, 2.27]	2.41 [0.81, 2.83]

8.2 Operating point results

We report the misclassification cost in terms of the True Cost Ratio (TCR) (see eq. (9)). This provides for a clear statement of the benefits of applying a spam filter. In each case, a 95% confidence interval is calculated using *BCost*, with the algorithm settings chosen according to the recommendations provided in [16].

The results for the operating point determined by the cost matrix chosen and the empirical distribution of the data are presented in Table 2. For each SVM configuration, the TCR results corresponding to using binary and normalized binary feature vectors are given. When the standard error rate is used to optimize SVM during training, the resulting classifier actually performs worse (i.e., costs more) than when no filtering is used at all. This is because both types of misclassification errors are treated in the same way, while the $C_{L\to S} \gg C_{S\to L}$. For all methods where the asymmetric nature of misclassification costs is accounted for, the performance is better than the baseline, and the hypothetical ESP using the filter would save up to four times in operating costs. It can be seen that normalization of SVM inputs appears to be beneficial.

8.3 Receiver Operating Characteristic (ROC) analysis

We have so far analyzed the performance of different SVM configurations at the chosen operating conditions. It is useful however to examine the general relationship between these classifiers for a wide range of operating conditions, which is the subject of ROC analysis [21].

Using the notion of a false positive and negative error rates, the misclassification cost can be expressed as

$$Cost_F = \pi^- FP_{spam} + \pi^+ \sum_{cat\in\mathcal{C}} \pi^{cat} C^{cat}_{L\to S} FN_{cat} \tag{10}$$

π^- and π^+ are the probabilities of spam and legitimate emails, respectively, FP_{spam} is defined as the ratio of the number of spam emails classified as legitimate to the total number of spam emails in the test set; for each category of legitimate email, FN_{cat} is defined analogously as the ratio of the number of legitimate emails in that category misclassified as spam to the total number of legitimate emails in that category. The tuple $(FP_{spam}, FN_{personal}, ..., FN_{promo})$ defines an operating point on an ROC. The full characteristic is produced by varying the classifier's decision threshold. One might consider

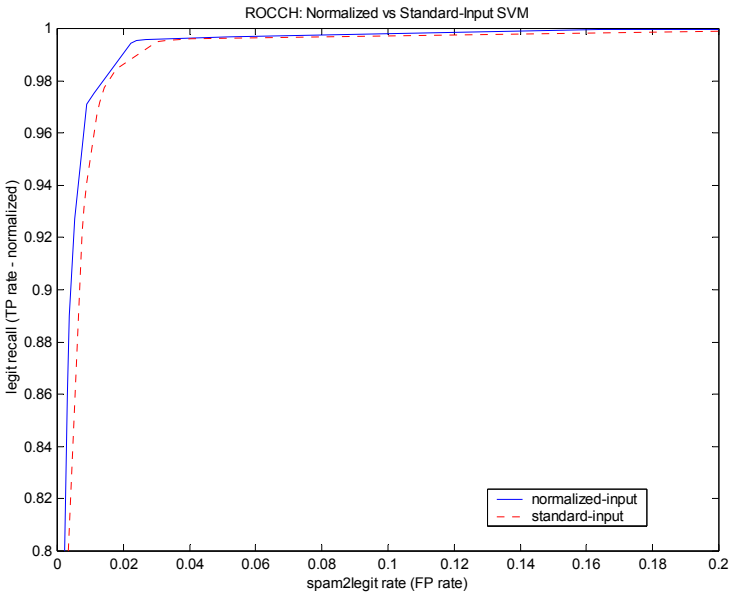

FIG. 1. *ROCCH combining all of the SVM configurations considered with and without SVM input normalization. Normalization is clearly benefitial.*

studying the ROC behavior in the multidimensional space, but such an approach would be difficult to visualize. Instead, as suggested in [21], the false negative rate is normalized by weighing different misclassification error proportionally to their costs. Thus, eq. (10) is rewritten as

$$Cost_F = \pi^- FP_{spam} + \pi^+ C^{avg}_{L\to S} FN_{norm} \qquad \text{where } C^{avg}_{L\to S} = \sum_{cat\in\mathcal{C}} \pi^{cat} C^{cat}_{L\to S}$$

and $FN_{norm} = \sum_{cat\in\mathcal{C}} \pi^{cat} C^{cat}_{L\to S} FN_{cat} / \sum_{cat\in\mathcal{C}} \pi^{cat} C^{cat}_{L\to S}$. This representation allows us to compare the different classifiers using the familiar two-dimensional ROC curves. In the ROC space, regions of constant misclassification cost correspond to lines of constant slope, $\frac{\pi^-}{\pi^+ C^{avg}_{L\to S}}$. Therefore, different sections of an ROC curve correspond to classifier settings that are optimal for different combinations of the average expected misclassification cost and the spam–legitimate distribution, assuming that the cost and category distribution within the legitimate class remains the same. Provost and Fawcett [21] pointed out that this is particularly useful when the true distributions and costs are unknown during the time of training. Given the ROCs for a set of different classifiers, the ROC Convex Hull (ROCCH) method [21] can be used to eliminate clearly inferior models and identify the ones that are optimal under some operation conditions.

To this end we performed to ROCCH analysis, treating the cases with and without input normalization separately. The results are depicted in Figure 1, which confirms the superiority of normalizing SVM inputs in this application (the normalized-input ROCCH fully dominates the other one). In either case, only the `cat`, `std` and `cost` configurations are potentially optimal. Note that `cost-avg` outperforms `cost` in Table 2, which indicates sub-optimal threshold selection for `cost`.

9 Conclusions

We proposed a category-specific cost model for filtering spam, where the cost of misclassifying legitimate is content-specific and the cost of misclassifying spam assumed to be uniform. The need for cost-sensitive training and evaluation of spam filters was discussed, and several methods of incorporating variable costs into SVM learning were investigated. The key observations resulting from this study can be summarized as follows:

When misclassification costs and the target probability distribution are known, all of the methods for making SVMs cost-sensitive investigated here performed well and offered a clear improvement over the standard SVM classifier, which may actually underperform the baseline case where no filtering is deployed.

The most effective and promising approaches to cost-sensitive SVM training were those which explicitly incorporated instance-specific misclassification costs into the optimization process, and those operating in a categorization framework.

The standard SVM, trained with uniform misclassification costs may be turned into a (surpassingly effective) cost-sensitive classifier by appropriately adjusting the classification threshold. Such a setting may be preferable if the cost/distribution settings are likely to change, since it does not require retraining — the standard SVM performs well under a wide range of operating conditions, as shown its ROC.

We demonstrated the practical difficulty of applying MetaCost when misclassification costs are highly asymmetric. A simple method of estimating the optimum cost factor for use with MetaCost was proposed as an effective way of circumventing the problem.

When SVMs are used in the probabilistic decision framework, it is crucial to *estimate* the cost factors used in decision making, rather to use the true cost factors directly. This should be seen as a form of optimizing the decision threshold of the classifier.

There seem to be an advantage of normalizing the binary SVM inputs for use in the spam-filtering application.

References

[1] 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), G. Potamias, V. Moustakis, and M. van Someren, eds., 2000, pp. 9–17.

[2] ——, *An experimental comparison of naive bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages*, in Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), N. Belkin, P. Ingwersen, and M. Leong, eds., 2000, pp. 160–167.

[3] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos, *Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach*, in Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), H. Zaragoza, P. Gallinari, and M. Rajman, eds., 2000, pp. 1–13.

[4] P. Bennett, *Assessing the calibration of naive bayes' posterior estimates*, tech. report, Dept. of Computer Science, School of Science, Carnegie Mellon University, 2000.

[5] L. Breiman, *Bagging preductors*, Machine Learning, 24 (1996), pp. 123–140.

[6] ——, *Out-of-bag estimation*, tech. report, Department of Statistics, University of California Berkeley, 1996.

[7] M. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. Sugnet, M. Ares, and D. Haussler, *Support vector machine classification of microarray gene expression data*, Tech. Report UCSC-CRL-99-09, University of California, Santa Cruz, 1999.

[8] W. Cohen, *Fast effective rule induction*, in Proceedings of the Twelfth International Conference on Machine Learning, 1995.

[9] P. Domingos, *MetaCost: A general method for making classifiers cost-sensitive*, in Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, ACM Press, 1999, pp. 155–164.

[10] H. Drucker, D. Wu, and V. Vapnik, *Support vector machines for spam categorization*, IEEE Transactions on Neural Networks, 10 (1999), pp. 1048–1054.

[11] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, *Inductive learning algorithms and representations for text categorization*, in Proceedings of 7th International Conference on Information and Knowledge Management, 1998, pp. 229–237.

[12] J. G. Hidalgo, M. M. López, and E. P. Sanz, *Combining text and heuristics for cost-sensitive spam filtering*, in Proceedings of the Fourth Computational Natural Language Learning Workshop, CoNLL-2000, Lisbon, Portugal, 2000, Association for Computational Linguistics.

[13] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*, in Proceedings of the Tenth European Conference on Machine Learning (ECML-98), 1998, pp. 137–142.

[14] H. Kaitarai, *Filtering junk e-mail: A performance comparison between genetic programming and naive bayes*, Tech. Report (presented at the Fall 1999 Meeting of the CMU Text Learning Group), Department of Electrical and Computer Engineering, University of Waterloo, November 1999.

[15] C. Lin, *Formulations of support vector machines: A note from an optimization point of view*, Neural Computation, 13 (2001), pp. 307–317.

[16] D. Margineantu and T. Dietterich, *Bootstrap methods for the cost-sensitive evaluation of classifiers*, in Proceedings of the 2000 International Conference on Machine Learning, ICML-2000, 2000.

[17] K. Morik, M. Imhoff, P. Brockhausen, T. Joachims, and U. Gather, *Knowledge discovery and knowledge validation in intensive care*, Artificial Intelligence in Medicine, 19 (2000), pp. 225–249.

[18] P. Pantel and D. Lin, *Spamcop: A spam classification & organization program*, in Learning for Text Categorization: Papers from the 1998 Workshop, Madison, Wisconsin, 1998, AAAI Technical Report WS-98-05.

[19] J. Platt, *Fast training of support vector machines using sequential minimal optimization*, in Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press, 1999.

[20] ——, *Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods*, in Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, eds., MIT Press, 1999.

[21] F. Provost and T. Fawcett, *Robust classification for imprecise environments*, Machine Learning, 42 (2001), pp. 203–231.

[22] 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.

[23] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos, *Stacking classifiers for anti-spam filtering of e-mail*, in Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001), L. Lee and D. Harman, eds., Carnegie Mellon University, 2001, pp. 44–50.

[24] V. N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, 1998.

[25] G. Wahba, *Support vector machines, reproducing kernel hilbert spaces and the randomized GACV*, in Advances in Kernel Methods: Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press, 1999, pp. 69–88.

[26] B. Zadrozny and C. Elkan, *Learning and making decisions when costs and probabilities are both unknown*, Tech. Report CS2001-0664, UCSD, 2001.