Analyzing Behaviorial Features for Email Classification

Steve Martin **Anil Sewani** **Blaine Nelson** **Karl Chen** **Anthony D. Joseph**
{steve0, anil, nelsonb, quarl, adj}@cs.berkeley.edu
University of California, Berkeley
Berkeley CA 94720-1776

Abstract

Many researchers have applied statistical analysis techniques to email for classification purposes, such as identifying spam messages. Such approaches can be highly effective, however many examine incoming email exclusively — which does not provide detailed information about an individual user's behavior. Only by analyzing outgoing messages can a user's behavior be ascertained. Our contributions are: the use of empirical analysis to select an optimum, novel collection of behavioral features of a user's email traffic that enables the rapid detection of abnormal email activity; and a demonstration of the effectiveness of outgoing email analysis using an application that detects worm propagation.

1 Introduction

Electronic mail has become one of the most ubiquitous methods of communication. By 2006, global email traffic is expected to surge to 60 billion messages daily (International Data Group, 2002). However, this explosive growth comes with a variety of problems. Unsolicited marketing messages, or spam, account for more than half of the total daily message traffic (Message Labs, 2004). In addition, at least eight out of the ten computer worms most frequently reported during 2004 to a prominent anti-virus company spread via email (Sophos Corporation, 2004). Finally, phishing attacks are a growing concern.

Current methods for detecting email system abuse mostly work by examining characteristics of incoming messages. For example, spam detectors calculate statistical features on received email for classification. Current commercial virus scanners compare hash values calculated on each arriving message to human-generated signatures.

While such approaches are quite effective, we believe that several improvements can be made. First, to the best of our knowledge, the features used in current techniques only examine incoming email, which is usually composed of messages from several distinct users and could be contaminated with spam and virus email. Thus, mail in a specific user's inbox cannot be used to profile that user's behavior. Outgoing email, however, can be observed to characterize a user's normal email behavior, after which abnormal behavior caused by a compromised machine can be detected and contained at the source. This individual-user based analysis, when combined with techniques that examine incoming mail, could form an extremely strong defense against the spread of novel worms and spam.

Second, we find that many proposed email classification techniques take advantage of statistical methods. However, we believe that their performance can be improved by more judicious feature selection. From our survey of current literature, we feel that the consequences of feature selection have often been underemphasized in the anti-spam and anti-virus community.

To address the first problem, we provide a collection of novel features designed to capture a user's outgoing email behavior, around which statistical classification models can be built. However, any large, diverse user population will have users that send email infrequently, making initial per-user model creation difficult for such users. Nonetheless, through empirical analysis of the Enron dataset (Klimt & Yang, 2004), we observe that users can be grouped into common clusters enabling sets of users to be largely represented by a single behavioral model.

This work was supported in part by grants from Hewlett Packard, Sun Microsystems, the State of California under the MICRO program, and jointly by the US Department of Homeland Security Advanced Research Projects Agency (HSARPA) and the National Science Foundation (NSF) under grant ANI-0335241.

For the second problem, we present techniques from statistical learning theory that can be applied to feature analysis. We demonstrate the utility of these techniques by applying them to our feature set within the context of detecting novel worm propagation.

This paper is organized as follows: Section 2 discusses relevant previous research, Section 3 describes our email analysis methods, Section 4 presents our feature analysis approach, and we demonstrate our techniques with an application in Section 5. We close with a few thoughts on our work and future directions.

2 Related Work

Statistical classification of email is an active research area. Some of the features we use to characterize user behavior have been used previously for classifying spam (Graham, 2002) and detecting novel email viruses (Stolfo et al., 2003; Stolfo et al., 2004). However, previous techniques have not examined the contributions of these features to their classification or the sensitivity of their model to those features.

The work on spam classification using feature selection has mostly been based on heuristics (Meyer & Whateley, 2004), and in a fewer cases, has applied well-known statistical methods from text classification (Sahami et al., 1998; Kolcz et al., 2004).

One interesting method of examining messages is the construction of social networks (Boykin & Roychowdhury, 2004; Newman et al., 2002). In these models, users within a network are considered as nodes of a graph, and communication between any two nodes is indicated via an edge between the nodes. Cliques of nodes form a social network, indicating common communication patterns. Communication that violates these behavioral patterns is considered abnormal.

Stolfo et al. created an email data mining system that uses social network analysis along with other user behavior features to identify viral propagations (Stolfo et al., 2004; Stolfo et al., 2003). The system maintains user cliques for every user in the system. Other features considered include variance in number of distinct recipients, send rate, and number of emails with attachments over a window of emails. Histograms based on these features are then constructed to profile a user's current and long term email behavior. Our results include features used by the authors.

3 Feature Descriptions

The term *feature* describes a statistic that represents a measurement of some aspect of a given user's email activity or behavior. We focus in this paper on features we believe from observation could be useful in detecting abnormal sending behavior that results from a worm or virus infection. Similar techniques could also be applied to design features for spam detection.

We selected and implemented two dozen separate features with the underlying goal of obtaining a set of statistics that accurately distinguishes between normal and abnormal email activity. Each feature returns either a continuous or multinomial value — as an example, a frequency calculation returns a number, whereas a feature involving types of email attachments is represented as an array of bits, where each bit represents the presence of a specific type of attachment.

Our features consist of those calculated on a single email (*i.e.,* single points in ongoing email activity) and those that examine several emails over a fixed amount of time (*i.e.,* trends in message characteristics, such as a running average of the number of characters in a single user's email subjects).

The following sections briefly describe our feature choices and why they are included. We revisit the question of the importance of each feature in distinguishing between separate users in Section 4.

3.1 Per-Email Features

The following sections describe numerical values calculated on a per-email basis.

3.1.1 Single Email Multinomial-Valued Features

Features in this category represent their output as one or more bits. Multi-bit or multinomial return values are in the form of a bit string.

Presence of HTML: There are exploits resulting from buggy HTML parsing by the mail user agent, *e.g.,* the Kak worm (Symantec Corporation, 2005).

Presence of script tags/attributes: These statistics are particularly useful in detecting emails that are potential security risks.

Presence of embedded images: Embedded images are often used by spammers to verify address lists, and could be used to exploit buggy image processing, *e.g.,* the Microsoft JPEG vulnerability (Microsoft Corporation, 2004).

Presence of hyperlinks: Several worms propagate by emailing links to infected web pages, *e.g.,* the Bubbleboy virus (Symantec Corporation, 2005).

MIME types of file attachments: The MIME type of a file is assigned by the sending mail user

agent either using magic numbers (see below), or through table-lookup on the filename extension. Each binary value represents the presence of a specific type of file.

Presence of binary, text attachments: This multinomial statistic helps in the case where an email has a binary or text file attached whose type is corrupt or unknown.

UNIX "magic number" of file attachments: Worms often assign misleading MIME types to fool virus scanners, *e.g.,* Nimbda (Symantec Corporation, 2005). The magic number is an accurate method of determining the true file type. If an attachment's magic number does not correspond to its MIME type, it could be malicious.

3.1.2 Per-Email Continuous Features

Number of attachments: Most people do not attach many files to their email, however several worms send messages that require opening an attachment to propagate.

Number of words/characters in the subject and body: These features help build a basic profile of the user's writing characteristics. Most virus text is randomly chosen, and spam messages have been found to share certain characteristics (Graham, 2002).

3.2 Features Calculated Over a Sending Window

We now describe numerical values calculated over a window typically consisting of the user's last twenty messages. All statistics are continuous.

Number of emails sent: Worms and spam-bots tend to send emails faster than the average user.

Number of unique email recipients: This feature counts addresses in the To:, CC:, and BCC: (if available) headers. The frequency with which one sends mail to distinct users captures an important aspect of email behavior.

Number of unique sender addresses: Many users have multiple active accounts on the same machine. However, a single machine sending from a large number of addresses at a high rate could indicate that the machine is compromised.

Average number of words/characters per subject, body; average word length: These features capture trends in email wording that could separate normal email from malicious activity, and among users.

Variance in number of words/characters per subject, body; variance in word length: These types of features have been used in previous work with some success to detect the behavior of email viruses.

Ratio of emails with attachments: Most users do not send large amounts of consecutive emails with attachments, whereas most worms do.

4 Feature Analysis

Our feature set is designed to capture specific elements of user email behavior that separate normal from abnormal (worm propagation) activity. To better understand the individual contributions of each feature to the overall effectiveness of our technique, we next present an analysis of the ability of each to capture information specific to individual behavior.

The methods we apply have been discussed in statistics literature and used in previous work to identify spam and classify text. However, to the best of our knowledge, they have not been applied to behavioral analysis for detecting novel worms. While our features are specific to behavioral analysis, these techniques are easily adapted to improve other classification problems through increasing understanding of the role of each feature.

We present our analysis in several parts: the feature selection problem background, feature histograms that capture separate elements of unique per-user behavior, methods for using covariance between the user labels and the data to choose the most relevant features for distinguishing viruses from normal user behavior, and a method for performing greedy feature selection.

4.1 Background

The problem of optimally selecting statistical features can be categorized as *feature extraction* and *feature selection*. Feature extraction creates a smaller set of features from linear combinations of the original features, while feature selection simply chooses a subset of the original features (in essence a boolean version of feature extraction).

In feature selection, choosing the subset of features that optimally predicts the desired function is computationally infeasible (NP-complete) - the number of subsets of a set grows exponentially with the set size. There are approximation algorithms, including a method using Principle Component Analysis (PCA), a feature extraction approach, to find directions in feature space that maximize variance. Classical PCA determines such directions, but fails to find individual

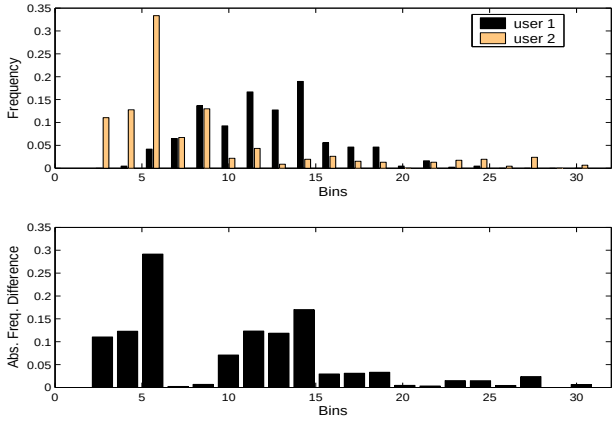

Figure 1: Feature histograms for two users in the Enron dataset and their differences

features that maximize variance. Instead, it determines linear combinations of the feature set. Naïve greedy selection of features by choosing the dominant feature in each principle component is effective, but fails to account for redundancy in the feature set. The selectivity of PCA can be enhanced by modifying the optimality criterion to favor sparse directions of maximum variance by imposing either an $L1$ constraint on the principle components (Zou et al., 2004) or a sparsity constraint through Semi-Definite Programming (d'Aspremont et al., 2004). These PCA-driven approaches can easily be incorporated in a framework for Directions of Maximum Covariance (Shawe-Taylor & Christiani, 2004) to discover directions in the data that maximize variation in labels.

4.2 Feature Histograms

One of the methods used in previous work on virus detection to classify behavior is histogram analysis. Looking at the distance between histograms of a specific feature over the data of two separate users is one method of estimating how similar the two are. We apply similar techniques to demonstrate the difference in per-feature distributions among individual users. For the analysis to follow, we used data for all users in the Enron data set that had sent-mail folders. There were a total of 126,078 emails between 148 users, with each user having between 3 and 8926 emails. Most users had under 1000 emails.

The top graph in Figure 1 shows normalized histograms for two users in the Enron data set of the values for the feature calculating the number of distinct addresses email is sent to over a window of messages. By taking the absolute value of the difference of each bin over these two histograms, we generate the bottom graph in Figure 1, which gives a visual representation of how different the two users' behavior is with regards to this feature. By summing up the histogram difference, we can generate a rough metric of per-feature user similarity.

To illustrate how our features separate individual behavior, we consider different pairs of users to plot trends in this metric. Figure 2 shows the per-feature histograms of normalized histogram differences between all combinations of users in the Enron dataset. Note that the maximum value of this difference is 2 (when the histograms being compared do not have any overlaps) and the minimum value is 0 (when the histograms completely overlap each other). For all histograms shown in Figure 2, we used a bin size of 0.02 (by dividing the range between 0 and 2 into 100 equal-size bins).

Figure 2 shows two important characteristics. First, it demonstrates that our behavioral features are different per user. Second, each feature behaves slightly differently over all users; certain statistics vary more widely than others. Both of these points motivate the analysis presented in the next section.

4.3 Covariance Analysis

Many statistical analyses exist for identifying the relevance of a feature to a given dataset. One of the more well known techniques is PCA which determines the directions in feature space that maximize variance of the multivariate random variable X.

However, while PCA is useful in many settings, its choice of directions in feature space do not necessarily lead to good classification. Instead, we apply a similar method, called *maximum covariance* (Shawe-Taylor & Christiani, 2004), that determines the directions in feature space that maximize the covariance between observations and their labels (correct classifications). This is accomplished through a singular value decomposition of the covariance matrix $C_{xy} = \mathrm{cov}\,[X, Y]$, or the correlation matrix $\mathrm{cor}\,[X, Y]$ where Y is the corresponding set of labels for each observation.

To apply the directions of maximum covariance technique, consider X to be an m-dimensional random variable representing the features of a given observation and Y to be an k-dimensional random variable corresponding to the label of X. Given a set of n observations of pairs $\{(X_i, Y_i)\}_{i=1}^{n}$, the empirical covariance matrix is given by

$$\begin{aligned}\hat{C}_{xy} &= \mathbf{E}\left[(X - \mathbf{E}\,[X])(Y - \mathbf{E}\,[Y])^T\right] \\ &= \tfrac{1}{n}\textstyle\sum_i X_i Y_i - \tfrac{1}{n^2}\sum_i X_i \sum_i Y_i\end{aligned}$$

The singular value decomposition decomposes C_{xy} by $C_{xy} = U\Sigma V^T$ where U is a $m \times m$ unitary matrix of x-principle components, Σ is a diagonal matrix of covariances, and V is a $k \times k$ unitary matrix of y-principle

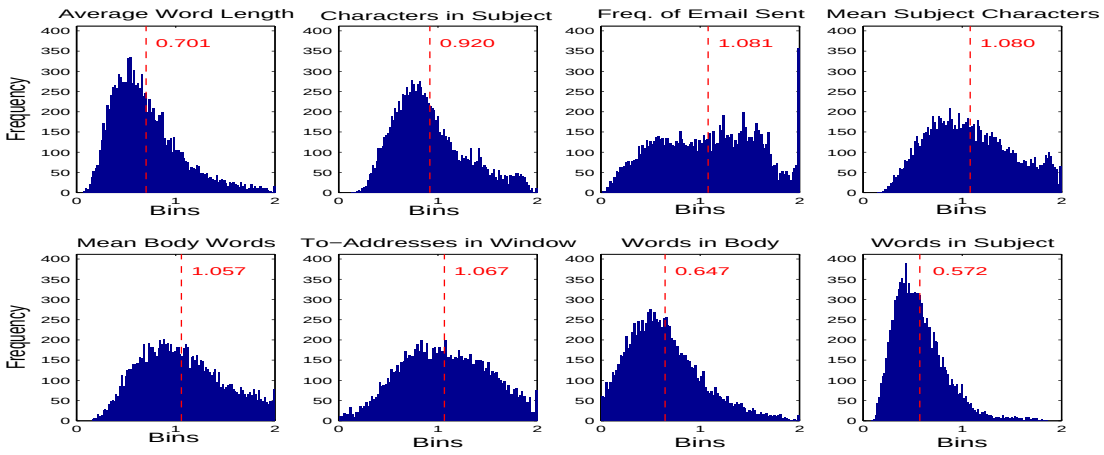

Figure 2: Per-feature histograms of histogram differences between all combinations of users in the Enron dataset. The dotted lines show the means for each histogram.

components. By computing directions of maximum covariance, the resulting principle components maximize the covariance between the random variable X and Y.

Figure 3 demonstrates this approach. We performed the maximal covariance analysis to distinguish users in the Enron dataset generating a "signature" for each user based on their distinguishing features. This signature is depicted as a single column in Figure 3. By clustering the users based on these signatures we were able to identify groups of users with similar behavioral characteristics as shown in the figure. The clusters indicate that several canonical behaviors can account for the majority of individual user behaviors making the deployment of systems based on per-user models feasible.

4.4 Feature Ranking

The task of feature selection, a concept that has been well studied in the statistics and machine learning literature (Guyon & Elisseeff, 2003; Blum & Langley, 1997), is the process of choosing a subset of a feature space that best represents the problem at hand while introducing the minimal amount of noise. We now present a simple method for using the results of covariance analysis to reduce the feature set by concentrating on the features that contribute the most covariance with the desired target labels Y. Moreover, we show in Section 5 that judicious feature selection significantly enhances classifier accuracy.

The simplest method to perform feature selection via covariance analysis is a greedy approach in which features are ranked according to their contribution to the first principle component of the covariance matrix. Suppose that the first principle component is given by $u_1 = \langle u_{1,1}, u_{1,2}, \ldots, u_{1,m} \rangle$. Then we simply rank the i-th feature according to its corresponding squared contribution, $u_{1,i}^2$. However, this naïve selection mechanism entirely ignores the possibility of redundancy between features.

To remove some degree of feature redundancy, we can deflate the covariance matrix as features are chosen. In this technique, features are selected in a greedy fashion, but after each selection, the covariance matrix is deflated by the basis vector corresponding to the selected feature. The result of this operation is

$$C'_{xy} \leftarrow \left(I_m - e_i e_i^T\right)^T C_{xy} \left(I_k - \alpha_i \alpha_i^T\right)$$

where e_i is the i-th basis vector, I_m is the $m \times m$ identity matrix, , I_k is the $k \times k$ identity matrix, and $\alpha_i = C_{xy} e_i / \|C_{xy} e_i\|$. By recalculating the principle components from the deflated covariance matrix, covariance captured by the selected feature is removed so that subsequent selections concentrate on portions of the covariance not captured by initial choices. By deflating, redundancy can be reduced, but the selection process is still greedy so the optimal subset of features is not necessarily chosen.

The greedy approaches presented here suffice for our demonstration of feature selection. In particular, in analyzing the features most relevant for distinguishing between a user and our test viruses, we found the following partial ranking:

1. Ratio of emails with attachments
2. Binary attachment
3. MIME type `application/octet-stream`
4. Magic type `application/x-ms-dos-executable`
5. Unclassified binary magic type
6. Frequency of emails in window
7. Number of attachments

Not surprisingly, the dominant features personify the

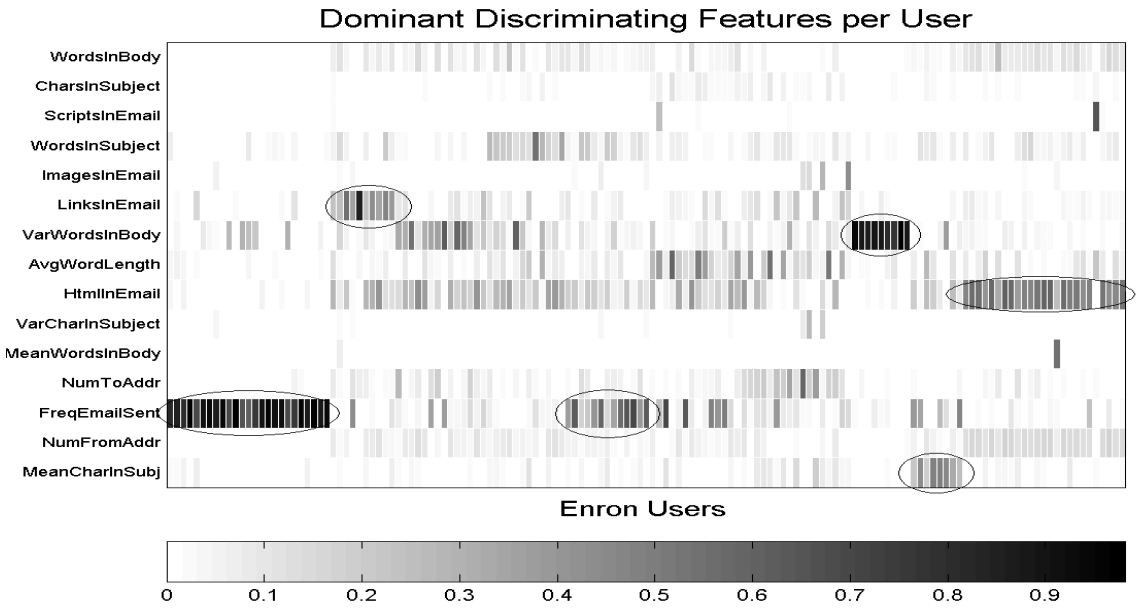

Figure 3: A plot of the direction of maximal covariance for each user in the Enron data. The users are clustered based on these directions to emphasize the similarity between users. The obvious clusters are circled for clarity.

fact that viruses' behavior deviates primarily in the number and type of attachments used as well as the rate at which they send email. This list demonstrates the redundancy problem in that three of the top five features are related to whether an executable attachment is present. In the future, more judicious pruning of redundant features is warranted to capture the essential information. Preliminary feature selection based on Fisher discriminant analysis (Shawe-Taylor & Christiani, 2004) produced results similar to our method.

5 Application: Novel Worm Detection

One of the most prevalent security problems in computing today is the rampant proliferation of malicious, self-propagating computer viruses known as worms. While vulnerabilities exploited by virus authors vary widely, some of the most damaging worms have used email to propagate. As an example, the 2004 MyDoom email worm was one of the fastest spreading email worms to date, and some of its variants continue to slip by virus scanners.

While protection against computer viruses in general continues to be an area of intense research, traditional anti-virus defenses deployed in the field have not changed significantly for many years. Unfortunately, recent worms have demonstrated that in the time it takes to generate new signatures and apply them to anti-virus scanners, widespread infection among vulnerable hosts can already occur.

Our method of profiling email user behavior is very applicable to the problem of detecting novel worm propagation. To choke off avenues for infection as quickly as possible, we use our features to classify outgoing mail traffic so that machines suspected of being compromised can be quickly isolated.

We demonstrate the effectiveness of our feature set at determining worm infections with several model types, including Support Vector Machines (SVMs), and simple Naïve Bayes classifiers. In addition, we show how the analysis presented in Section 4, when applied to feature selection, dramatically impacts their performance.

We discuss the application in three parts: construction of the training and testing data, results using SVMs, and results using Naïve Bayes classifiers.

5.1 Evaluation Methods

We used several virus-free data sources: a corpus created by a custom real-time email interception framework that collected data from 20 volunteers in our department, the Enron data set, and several user's sent mail folders. These data sets helped us determine the distribution shapes of continuous features, which in our experiments were Gaussian or Exponential.

We captured real email worm messages from the Bagle.f, Netsky.d, MyDoom.u, MyDoom.m, and Sobig.f email worms by infecting VMWare virtual machines and using a transparent SMTP proxy setup to

intercept all SMTP traffic on port 25. We chose these worms both due to their virulence and because each behaves in a slightly different manner with regards to our features.

We constructed training and test sets by creating artificial traces combining our clean and infected email data. The clean email trace was obtained from one of the authors' 'Sent mail' folder. To simulate worm activity, we interleaved viral emails into the clean email corpus. The dates of the infected messages were corrected to maintain consistency, but interarrival times were kept the same so that frequency information was retained. The training set consisted of 800 normal emails and 800 viral emails from 2 different viruses (400 emails from each virus). The test set consisted of 3000 normal emails and 1200 viral emails from 3 different viruses.

We next examine the performance of classification models in detecting viral traffic using our artificial email traces. In winnowing down our feature set to the most relevant features, we use several well-known statistical methods to show general degradation in performance as the feature set grows in size.

5.2 Support Vector Machines

The effectiveness of feature selection can be seen in the performance of anomaly detection via the one-class Support Vector Machine (SVM). In this paradigm, the objective is to learn the region of support of a distribution of "normal" data; that is, the area that contains most of the probability mass. A one-class SVM applies a linear algorithm that attempts to maximally separate the "normal" data from the origin via a hyperplane boundary. This technique's properties enable it to be transformed into a non-linear algorithm by application of a similarity measure known as a kernel. The details of this technique are beyond the scope of this paper. See (Shawe-Taylor & Christiani, 2004) for a thorough explanation of kernel techniques and SVMs.

For our purposes, the relevant detail of the one-class SVM is our choice of kernel. For this exposition, the Gaussian (RBF) kernel was applied after standardizing the data. The one-class SVM was trained to allow only a small fraction, 0.1%, of outliers during training.

5.3 Naïve Bayes Classification

To further demonstrate the importance of judicious feature selection, the process was applied to a two-class Naïve Bayes classifier. Naïve Bayes models classify by applying Bayes rule to observed data via class-conditional distributions. The probability of the data is given by the distribution's fit to known infected data. The simplifying assumption made by Naïve Bayes models is that the features of an observation are independent given its classification. While this assumption is often violated, the model is widely used in spam detection (Meyer & Whateley, 2004; Segal et al., 2004) and suffices for our purposes of demonstrating model degradation due to irrelevant features.

5.4 Discussion

The results of testing the SVM and Naïve Bayes models with a variable number of features are shown in Figure 4 as plots of the overall classification accuracy. While not shown in the figure, the false positive rate generally started high due to the lack of generality caused by fewer features and decreased as more features were added. The false negative rates generally increased due to overfitting as extraneous features were added (although a few experiments initially had high false negative rates due to the lack of generality of a limited number of features).

It is also evident in Figure 4 that the degradation in performance of the Naïve Bayes and SVM models differ. The step-like performance of Naïve Bayes is attributable to the thresholding of the posterior probability in determining the classification of an email. Meanwhile, the more erratic behavior of the SVM is likely due to its instance-based nature. The SVM's classification boundary is supported by representative emails which may change substantially as the feature set is altered.

These experiments reflect well known guidelines; regardless of the model, too few features are insufficient for generality while too many features cause overfitting. In addition, the curse of dimensionality, which says that the size of the training set necessary to learn the classification function grows exponentially with respect to the dimensionality of the data, further deters a bloated feature set.

6 Conclusion and Future Work

This paper presents an approach to virus detection using feature generation on outgoing email traffic to build models of user behavior. The approach is augmented by pruning irrelevant features. This feature set is shown to be effective in capturing the differences between user and virus behavior. Moreover, initial analysis indicates that user behavior can be clustered into sets of common models that describe the general behavior patterns of most users hence making a large scale detection system feasible.

As was demonstrated in Section 5, a judicious selection of features can significantly improve the performance

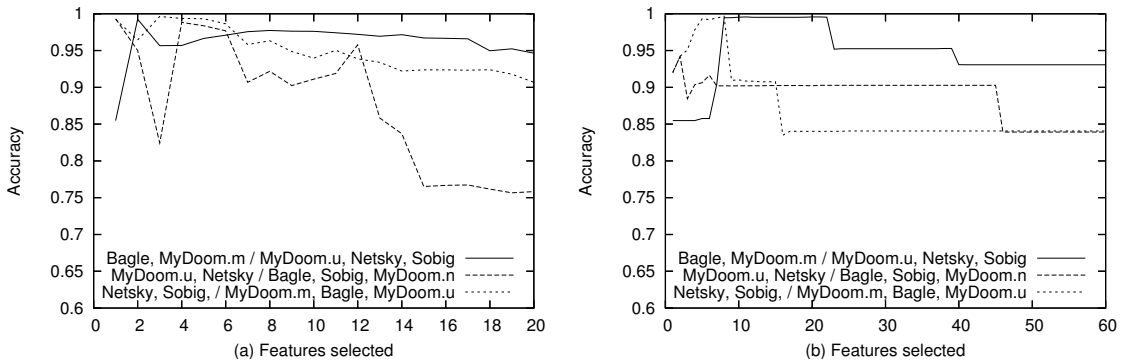

Figure 4: (a) Change in model accuracy as more features are greedily added to an SVM model. (b) Change in model accuracy as more features are greedily added to a Naïve Bayes Classifier. In the key, the first two specify the viruses trained against, and the last three the viruses tested against.

of statistical learning techniques. This is because features that are irrelevant to the classification problem can cause a classifier to learn sub-optimal rules resulting in overfitting. While this problem is not ubiquitous among all classifiers (some incorporate feature selection directly into learning, *e.g.,* decision trees), pruning out irrelevant features often improves performance by decreasing the dimensionality.

The prototype explained in Section 5 is a first step in designing a complete system for monitoring outgoing traffic to detect local infections. There are several features based on word distributions and social network analysis that can be included in our feature set for better prediction of user behavior. In addition, any deployable system will have to account for the temporal changes in user behavior via periodic retraining. These considerations are being incorporated in our ongoing design of a virus detection engine.

References

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*, 245–271.

Boykin, P. O., & Roychowdhury, V. (2004). Personal email networks: An effective anti-spam tool. `http://arxiv.org/abs/cond-mat/0402143`.

d'Aspremont, A., Ghaoui, L. E., Jordan, M. I., & Lanckriet, G. R. G. (2004). A direct formulation for sparse PCA using semidefinite programming. *Advances in Neural Information Processing Systems*.

Graham, P. (2002). A plan for spam. [online]. `http://www.paulgraham.com/spam.html`.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 245–271.

International Data Group (2002). Worldwide email usage 2002 - 2006: Know what's coming your way.

Klimt, B., & Yang, Y. (2004). Introducing the Enron corpus. *CEAS*.

Kolcz, A., Chowdhury, A., & Alspector, J. (2004). The impact of feature selection on signature-driven spam detection. *CEAS*.

Message Labs (2004). Message Labs Intelligence: Annual email security report 2004. `http://www.messagelabs.com/intelligence/2004report/`.

Meyer, T. A., & Whateley, B. (2004). SpamBayes: Effective open-source, Bayesian based, email classification system. *CEAS*.

Microsoft Corporation (2004). Microsoft security bulletin MS04-028. `http://www.microsoft.com/technet/security/bulletin/MS04-028.mspx`.

Newman, M. E. J., Forrest, S., & Balthrop, J. (2002). Email networks and the spread of computer viruses. *Physical Review*, *E 66*.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk E-mail. *Learning for Text Categorization: Papers from the 1998 Workshop*. Madison, Wisconsin: AAAI TR WS-98-05.

Segal, R., Crawford, J., Kephart, J., & Leiba, B. (2004). Spamguru: An enterprise anti-spam filtering system. *CEAS*.

Shawe-Taylor, J., & Christiani, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Sophos Corporation (2004). War of the worms: Netsky-p tops list of year's worst virus outbreaks. `http://www.sophos.com/pressoffice/pressrel/uk/20041208yeartopten.html`.

Stolfo, S. J., Hershkop, S., Wang, K., Nimeskern, O., & Hu, C. W. (2003). A behavior-based approach to secure email systems. *Mathematical Methods, Models and Architectures for Computer Networks Security*.

Stolfo, S. J., Li, W. J., Hershkop, S., Wang, K., Hu, C. W., & Nimeskern, O. (2004). Detecting viral propagations using email behavior profiles. *ACM TOIT*.

Symantec Corporation (2005). Symantec Security Response. `http://www.symantec.com/avcenter/`.

Zou, H., Hastie, T., & Tibshirani, R. (2004). Sparse principal component analysis. *Technical Report, Statistics Dept., Stanford University*.