Anomaly Detection in Dynamic Systems Using Weak Estimators

Justin Zhan[*], B. John Oommen[†] and Johanna Crisostomo [‡]

Abstract

Anomaly detection involves identifying observations that deviate from the normal behavior of a system. One of the ways to achieve this is by identifying the phenomena that characterize "normal" observations. Subsequently, based on the characteristics of data learned from the "normal" observations, new observations are classified as being either "normal" or not. Most state-of-the-art approaches, especially those which belong to the family parameterized statistical schemes, work under the assumption that the underlying distributions of the observations are stationary. That is, they assume that the distributions that are learned during the training (or learning) phase, though unknown, are not time-varying. They further assume that the same distributions are relevant even as new observations are encountered. Although such a "stationarity" assumption is relevant for many applications, there are some anomaly detection problems where stationarity cannot be assumed. For example, in network monitoring, the patterns which are learned to represent normal behavior may change over time due to several factors such as network infrastructure expansion, new services, growth of user population, etc. Similarly, in meteorology, identifying anomalous temperature patterns involves taking into account seasonal changes of normal observations. Detecting anomalies or outliers under these circumstances introduces several challenges. Indeed, the ability to adapt to changes in non-stationary environments is necessary so that anomalous observations can be identified even with changes in what would otherwise be classified as "normal" behavior. In

[*]This author can be contacted at: Computer Science Department, North Carolina A&T State University. E-mail address: justinzzhan@gmail.com.

[†]*Chancellor's Professor*; *Fellow: IEEE* and *Fellow: IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail address: oommen@scs.carleton.ca.

[‡]This author can be contacted at: CyLab Japan Campus, Carnegie Mellon University, USA. E-mail address: jcrisostomo@andrew.cmu.edu.

this paper, we proposed to apply a family of weak estimators for anomaly detection in dynamic environments. In particular, we apply this theory to spam email detection. Our experimental results demonstrate that our proposal is both feasible and effective for the detection of such anomalous emails.

Terms: *Design, Algorithms, Performance*.
Keywords: *Anomaly Detection, Dynamic Systems, Weak Estimator*.

1 Introduction

The state-of-the-art in anomaly detection works with the assumption that both normal and anomalous data follow data distributions that are stationary. While this is, in one sense, acceptable, it is still a limitation on the types of detections possible, and sets the boundary on what is currently solvable. The aim of this paper is to relax this limitation, and to consider how novel estimation methods can be used to achieve spam filtering and the detection of anomalous data even when the underlying distributions change with time. In that sense, this paper is of a pioneering sort!

Spam filtering has been a very active field of research primarily because of the vast amount of unsolicited electronic mail that fills up users' mailboxes which, in turn, result in wasted resources and loss of productivity. Such mails can be used as a medium by which malicious entities compromise the user's system, and among other things, gain confidential information. Rather than consider anomaly detection in the larger context, in this paper, we present a statistical *spam* filtering approach based on weak estimation methods [17] [23].

In text classification, spam filtering, like most text filtering tasks, can be considered as a special case of single-label Text Classification (TC). In the spam filtering case, often referred to as binary TC, each document must be assigned to either category c_i or to its complement $\bar{c}_i$ [19]. In text classification, filtering is used to block the delivery of documents that the receiver deems irrelevant, so that only relevant documents are presented. Spam filtering may also be considered an anomaly detection task where non-spam emails, also referred to 'ham', are considered normal observations, while spam emails are considered anomalies, or deviations from the normal observation.

Statistical approaches classify a new observation by determining the probability of it being generated from the model derived during training. Moreover, as mentioned in [4], statistical parametric techniques estimate the parameters from data with certain assumptions about the underlying distribution. This implies that the quality of the results of the detection process depends greatly on the estimator used. The problem, however, occurs when the distribution of the underlying

data changes as the number of observations increase. Methods designed to solve this problem with non-stationary environments are briefly discussed in [17], where typical strategies such as sliding windows or change-point detection methods are applied.

The objective of this paper is to produce a personalized spam filter that is capable of adapting to changes brought about by variations in the distribution of ham and spam emails. The proposed solution is to employ a Stochastic Learning-based Weak Estimator (SLWE) [17], which is a novel estimation method that has demonstrated promising results in detecting source changes for the purpose of adaptive file compression. The rationale for choosing a weak estimator for non-stationary environments is that estimators that converge with a probability of 1 (e.g. the Maximum Likelihood Estimation (MLE) and Bayesian estimates) cannot easily unlearn and adapt to the changes in the new environment.

2 Related Work

2.1 Spam Filtering

Methods for filtering mail are continuously being developed and improved in order to become more robust against spam, which are themselves continuously being improved and crafted so as to go undetected by current mail filters. Spam filtering approaches range from rule-based methods such as SpamAssassin [10], to collaborative methods harnessing social networks [8]. However, most of the well-explored techniques belong to statistical approaches. Zhang *et al.* evaluated several of these techniques in [24], while Metsis *et al.* [11] compared the performance of several Naive Bayesian spam filters. Moreover, some researchers have worked on the application of online linear classifiers [20] (such as the perceptron algorithm and SVMs) to spam filtering.

Another focus for improvement is to minimize false positives, which are legitimate mail items that are classified as spam, since these can also be very detrimental to productivity. Perhaps, this is what separates spam filtering from other text classification problems: false positives are very costly. As such, recent works such as [7] have looked into evaluation methods for assessing the performance of such filters, and efforts have been made to design novel cost-sensitive measures [3] that better capture the impact of false positives.

Similar to text classification, most spam filtering techniques employ the use of a set of features, usually words, to aid the filters in recognizing spam. Feature selection methods that are used in text classification can also be applied in spam filtering. In text categorization, effects of the feature selection process on classifier

performance was explored in [22, 13], while its effects on online spam filters have been studied in [20].

2.2 Stochastic Learning-based Weak Estimator

The SLWE has been utilized in a variety of applications that involve estimating distributions in non-stationary environments. One of its major applications is in data compression where the SLWE was used to estimate the probabilities of the source symbols allowing for an adaptive single-pass encoding process [18]. Moreover, results from pattern classification experiments with synthetic data have also shown that the use of weak estimation is more robust than MLE methods in identifying data source changes [17]. A recent work on an efficient routing algorithm for mobile ad-hoc networks has also utilized the aforementioned estimation scheme [16]. In their proposed solution, the authors have used the SLWE in the route selection algorithm to efficiently estimate the packet delivery probability among different available paths. Finally, a newly proposed strategy for a source address reputation system involves the use of the SLWE in conjunction with a linear classifier for packet classification [5]. In this scheme, each packet is composed of symbols which have to be classified as belonging to one of two classes. Since the actual distribution of the symbols is unknown, the SLWE was utilized to update the estimates for each new observation. These and various other works have explored the applications of weak estimation. In this paper, we investigate its applicability in spam filtering tasks.

3 Weak Estimation Approach

There are a few problems which we have recently encountered, where strong estimators pose a real-life concern. One scenario occurs in pattern classification involving moving scenes. The same situation is also encountered when one attempts the adaptive encoding of files which are interspersed with text, formulae, images and tables. Similarly, if we are dealing with adaptive data structures, the structure changes based on the information about the underlying data distribution, which is given by the estimator. Thus, if the estimator used is "strong" (i.e., it converges w. p. 1), it is unlikely that the learned data structure will change from a structure that it has converged to. Indeed, we can conclusively demonstrate that it is sub-optimal to work with strong estimators in such application domains, i.e., when the data is truly *non-stationary*.

In this section, we will introduce[1] a Stochastic Learning Weak Estimator (SLWE),

[1]The rest of this section essentially cite the results from [17], and these are included here to render

and which is developed by using the principles of stochastic learning [17]. In essence, the estimate is updated at each time instant based on the value of the current sample. However, this updating is not achieved using an *additive* updating rule, but rather by a *multiplicative* rule, akin to the family of linear action-probability updating schemes [14, 15]. The formal results that we have obtained for the binomial distribution are quite encouraging. To render the explanation simple, let us assume that the learning updating rule has a user-defined coefficient, λ. We shall show that our new estimator converges weakly, and that this convergence is *independent of the value of* the learning coefficient, λ. Furthermore, the rate of convergence, is determined *completely* by the eigenvalue of the transition matrix of the underlying learning process, which is an explicit function of only λ. Besides, the variance of the estimate is also controlled by the *same* learning coefficient, λ. Analogous results are available for the multinomial case.

Let us assume that the estimated parameters follow a binomial/multinomial distribution. The binomial distribution is characterized by two parameters, namely, the *number* of Bernoulli trials, and the parameter characterizing *each* Bernoulli trial. In this regard, we assume that the number of observations is the number of trials. The aim is thus to estimate the *Bernoulli* parameter for each trial, which is achieved here by using stochastic learning methods.

Let X be a binomially distributed random variable, which takes on the value of either '1' or '2'[2]. We assume that X obeys the distribution S, where $S = [s_1, s_2]^T$. In other words,

$$\begin{aligned} X &= \text{'1' with probability } s_1 \\ &= \text{'2' with probability } s_2\,, \end{aligned}$$

where, $s_1 + s_2 = 1$.

Let $x(n)$ be a concrete realization of X at time 'n'. The intention of the exercise is to estimate S, i.e., s_i for $i = 1, 2$. We achieve this by maintaining a running estimate $P(n) = [p_1(n), p_2(n)]^T$ of S, where $p_i(n)$ is the estimate of s_i at time 'n', for $i = 1, 2$. Then, the value of $p_1(n)$ is updated as per the following simple rule[3] :

$$p_1(n+1) \leftarrow \lambda p_1(n) \qquad \text{if } x(n) = 2 \tag{1}$$

$$\leftarrow 1 - \lambda p_2(n) \quad \text{if } x(n) = 1\,. \tag{2}$$

where λ is a user-defined parameter, $0 < \lambda < 1$, and $p_2(n+1) \leftarrow 1 - p_1(n+1)$.

In the interest of simplicity, we omit the index n, whenever there is no confusion, and thus, in such cases, we use P and $P(n)$ interchangeably.

---

this paper to be a self-contained document.

[2]We depart from the traditional notation of the random variable taking values of '0' and '1'.

[3]This rule is analogous to the L_{RI} updating scheme widely acclaimed in the field of learning automata.

The result below shows that the mean of P, obtained as per Equations (1) and (2), converges exactly to S.

Theorem. *Let X be a binomially distributed random variable, and $P(n)$ be the estimate of S at time 'n'. Then, $E[P(\infty)] = S$.*

Proof. Based on the updating scheme specified by Equations (1) and (2), the conditional expected value of $p_1(n+1)$ given P can be seen to be:

$$\begin{aligned} \mathrm{E}[p_1(n+1)|P] &= \lambda s_2 p_1 + s_1 - \lambda s_1 + \lambda s_1 p_1 && (3)\\ &= (1-\lambda)s_1 + \lambda p_1(s_1+s_2) && (4)\\ &= (1-\lambda)s_1 + \lambda p_1 . && (5) \end{aligned}$$

Taking expectations a second time, we can write (5) as:

$$\mathrm{E}[p_1(n+1)] = (1-\lambda)s_1 + \lambda \mathrm{E}[p_1(n)] . \tag{6}$$

As $n \to \infty$, $\mathrm{E}[p_1(n+1)]$ and $\mathrm{E}[p_1(n)]$ both converge[4] to $\mathrm{E}[p_1(\infty)]$. Solving for $\mathrm{E}[p_1(\infty)]$ from (6) leads to:

$$\mathrm{E}[p_1(\infty)](1-\lambda) = (1-\lambda)s_1 , \tag{7}$$

implying that $\mathrm{E}[p_1(\infty)] = s_1$. Similarly, $\mathrm{E}[p_2(\infty)] = s_2$, and the result follows. □

The next results which we present, indicates that $\mathrm{E}[P(n+1)]$ is related to $\mathrm{E}[P(n)]$ by means of a stochastic matrix. We prove this result and its implications.

Theorem. *If the components of $P(n+1)$ are obtained from the components of $P(n)$ as per Equations (1) and (2), where* **M** *is a stochastic matrix. Thus, the limiting value of the expectation of $P(.)$ converges to S, and the rate of convergence of P to S is* fully determined *by λ.*

Proof. Consider the vector form of (6), obtained by replacing the term $(1-\lambda)s_1$ by $(1-\lambda)s_1 p_1 + (1-\lambda)s_1 p_2$, since $p_1 + p_2 = 1$. Substituting the above equality, simplifying and taking expectations again leads to the following vectorial form:

$$\mathrm{E}[P(n+1)] = \begin{bmatrix} s_1+\lambda s_2 & (1-\lambda)s_2 \\ (1-\lambda)s_1 & s_2+\lambda s_1 \end{bmatrix}^T \mathrm{E}[P(n)] . \tag{8}$$

[4] $\mathrm{E}[p_1(n)]$ converges to a limit because the coefficient of the linear difference equation is λ, with $0 < \lambda < 1$.

This proves the first claim of the theorem.

The second claim of the theorem follows by solving the vectorial difference equation, and taking the limit as n is increased to infinity. The final result follows since the *only* non-unity eigenvalue of (8) is λ. □

From the analysis given above, we can derive the explicit expression for the asymptotic variance of the SLWE. We show that a small value of λ leads to fast convergence and a large variance. As opposed to this, a large value of λ implies slow convergence and a small variance.

Theorem. *Let X be a binomially distributed random variable governed by the distribution S, and $P(n)$ be the estimate of S at time 'n' obtained by (1) and (2). Then, the algebraic expression for the variance of $P(\infty)$ is* fully *determined by λ.*

Proof. Using the same notation as above, the square of p_1 at time '$n+1$' is given by:

$$\begin{aligned} p_1^2(n+1) &\leftarrow \lambda^2 p_1^2 && \text{w.p. } s_2 \\ &\leftarrow 1-2\lambda(1-p_1)+\lambda^2(1-p_1)^2 && \text{w.p. } s_1 \\ &= 1-2\lambda+2\lambda p_1+\lambda^2(1-2p_1+p_1^2) \\ &= 1-2\lambda+2\lambda p_1+\lambda^2-2\lambda^2 p_1+\lambda^2 p_1^2 . \end{aligned}$$

Using Equations (1) and (2) we can write $\mathrm{E}\left[p_1^2(n+1)|P(n)=P\right]$ as:

$$\begin{aligned} \mathrm{E}\left[p_1^2(n+1)|P(n)=P\right] &= \\ \lambda^2 p_1^2 s_2+(1-2\lambda+\lambda^2)s_1+2\lambda(1-\lambda)p_1 s_1+\lambda^2 p_1^2 s_1 &= \\ \lambda^2 p_1^2+2\lambda(1-\lambda)p_1 s_1+(1-\lambda)^2 s_1 . & \qquad (9) \end{aligned}$$

From (9), we observe that as $n\to\infty$, both $\mathrm{E}\left[p_1^2(n)\right]$ and $\mathrm{E}\left[p_1^2(n+1)\right]$ converge to $\mathrm{E}\left[p_1^2(\infty)\right]$. Thus, by gathering terms involving $\mathrm{E}\left[p_1^2(n)\right]$, (9) can be written as:

$$\mathrm{E}\left[p_1^2(\infty)\right](1-\lambda^2)=2\lambda(1-\lambda)\mathrm{E}\left[p_1(\infty)\right]s_1+(1-\lambda)^2 s_1 ,$$

which can also be expressed as:

$$\begin{aligned} \mathrm{E}\left[p_1^2(\infty)\right](1+\lambda) &= 2\lambda\mathrm{E}\left[p_1(\infty)\right]s_1+(1-\lambda)s_1 && (10) \\ &= 2\lambda s_1^2+(1-\lambda)s_1 , && (11) \end{aligned}$$

where the last equalities hold since $\mathrm{E}\left[p_1(\infty)\right]=s_1$. Thus:

$$\mathrm{E}\left[p_1^2(\infty)\right] = \frac{2\lambda s_1^2 + (1-\lambda)s_1}{1+\lambda}. \tag{12}$$

We finally compute the variance of $p_1(\infty)$ as below:

$$\begin{aligned} \mathrm{Var}[p_1(\infty)] &= \mathrm{E}[p_1^2(\infty)] - \mathrm{E}[p_1(\infty)]^2 & (13)\\ &= \frac{(1-\lambda)s_1 s_2}{1+\lambda}, & (14) \end{aligned}$$

since $s_2 = 1 - s_1$, and the theorem is thus proved. □

When $\lambda \to 1$, the variance tends to zero, implying mean square convergence. The *maximum* value of the variance is attained when $\lambda = 0$, and the *minimum* value of the variance is achieved when $\lambda = 1$. Also, when λ is close to unity, the estimates are dominated by the initial values.

Our result seems to be contradictory to our initial goal. When we motivated our problem, we were working with the notion that the environment was non-stationary. However, the results we have derived are asymptotic, and thus, are valid only as $n \to \infty$. While this could prove to be a handicap, realistically, and for all practical purposes, the convergence takes place after a relatively small values of n. Thus, if λ is even as "small" as 0.9, after 50 iterations, the variation from the asymptotic value will be of the order of 10^{-50}, because λ also determines the rate of convergence, and this occurs in a geometric manner [14]. In other words, even if the environment switches its Bernoulli parameter after 50 steps, the SLWE will be able to track this change. Observe too that we do not need to introduce or consider the use of a "sliding window".

We conclude this section by presenting the updating rule given in (1) and (2) in the context of some schemes already reported in the literature. If we assume that X can take values of '0' and '1', then the probability of '1' at time $n+1$ can be estimated as follows[5]:

$$p_1(n+1) \leftarrow \frac{n}{n+1} p_1(n) + \frac{1}{n+1} x(n+1) \tag{15}$$

This expression can be seen to be a particular case of the rule (1) and (2), where the parameter

$\lambda = 1 - \frac{1}{n+1}$. This kind of rule is typically used in stochastic approximation [9], and in some reinforcement learning schemes, such as the Q-learning [21]. What we have done is to show that when such a rule is used in estimation, the mean

[5]Note that this expression is equivalent to that of the standard sample mean estimator.

converges to the true mean (independent of the learning parameter λ), and that the variance and rate of convergence are determined only by λ. Furthermore, we have derived the explicit relationships for these dependencies.

4 Experimental Evaluation

To assess the effectiveness of using the SLWE for spam filtering, and to compare it with a widely used estimation scheme, six different public benchmark collections were used. The results are then reported and compared using standard evaluation methods.

The six publicly available benchmark corpora (Table 1) used in this study are obtained from the Enron-Spam datasets [1] which were first used in a previous study by Metsis *et al.* [11]. According to the authors, these collections were created to evaluate personal spam filters where the ham messages were obtained from messages of a single user's mailbox. Also unlike other public datasets, the order in which the messages arrived were preserved, and more importantly, the varying proportions between ham and spam messages that a user receives over time were also emulated – rendering this to be a viable testing ground for a weak estimator-based strategy. Figure 1 shows how the proportion of ham email changes after every 100 messages. Specifically, the datasets that will be used are the following:

Table 1: Enron-Spam Datasets

Dataset	Ham-Spam Ratio	Total Ham	Period	Total Spam	Period
Enron-1	3:1	3672	12/1999-01/2002	1500	12/2003-09/2005
Enron-2	3:1	4361	12/1999-03/2001	1496	06/2001-07/2005
Enron-3	3:1	4012	02/2001-02/2002	1500	08/2004-07/2005
Enron-4	1:3	1500	04/2001-02/2002	4500	12/2003-09/2005
Enron-5	1:3	1500	01/2000-05/2001	3675	06/2001-07/2005
Enron-6	1:3	1500	06/2000-03/2002	4500	08/2004-07/2005

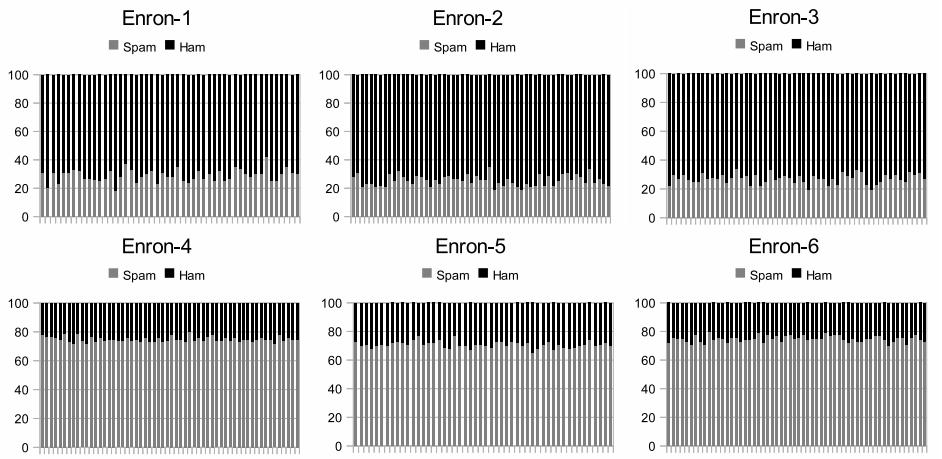

Figure 1: Proportion of ham messages in the Enron-spam dataset taken in intervals of 100 e-mails.

The datasets already come in a *preprocessed* form such that mail headers and HTML tags were removed. Moreover, the corpora do not contain spam messages written in non-Latin character sets. For the purposes of this study, further preprocessing was done using the OpenNLP API [2], which contain useful tools for accomplishing common text classification tasks. For each message, each word or token was tagged with a part of speech (POS) tagger after which an API to the WordNet dictionary [12, 6] was used to lemmatize each token. Tokens for which no lemmas could be found were retained. However, unlike most preprocessing done with text classification tasks, stop words were not removed from the messages.

For the training and classification phase, a methodology similar to what was undertaken in a previous study [11] was used. The process involved incremental updating of the filter, which closely resembles how learning-based personalized mail filters function. In this setting, only a small number of messages are available for initial training, but the filter learns as more ham and spam messages arrive and are marked by the user. On the other hand, unlike the aforementioned study, the size of the feature set used here is significantly smaller - only 200 features were used as opposed to 500 to 3,000 features. Another major difference is that although estimates are periodically updated, in the experiments mentioned here, feature selection is done only once during the initial training.

4.1 Feature Selection

All of the terms in the initial training set are considered as candidate features, and the Information Gain (IG), a common goodness of term measure in text classification, was employed for term space reduction. In a comparative study of different feature classification methods [22], it has been shown that the IG is effective in aggressive term removal without resulting in loss of classification accuracy. The IG associated with term t can be computed as:

$$\begin{aligned} G(t,c) = & - \sum_{c \in \{c_s, c_h\}} p(c) \log p(c) \\ & + p(t) \sum_{c \in \{c_s, c_h\}} p(c \mid t) \log p(c \mid t) \\ & + p(\bar{t}) \sum_{c \in \{c_s, c_h\}} p(c \mid \bar{t}) \log p(c \mid \bar{t}), \end{aligned} \tag{16}$$

where c_h and c_s refer to ham and spam classes respectively. This measures the amount of information gained about the class label by knowing the presence or absence of a term t in the message [13]. Depending on the task, terms may be selected given that their IG values are above a certain threshold. On the other hand, terms may also be ranked according to these values where the top k terms with high values are selected for classification. The latter was used here, with k=200.

4.2 Training and Classification

For classification, a multi-variate Bernoulli Naive Bayes (NB) classifier was employed. Each document d was represented as a binary vector $\overrightarrow{x} = \langle x_1, x_2, \ldots, x_k \rangle$ where $x_i \in \{1,0\}$, represents the presence or absence of the i^{th} feature in document d. As with other versions of NB classifiers, it was assumed that the occurrence of a term in a class is independent of the occurrence of other terms in the same class. A document d was classified as being spam according to the following rule:

$$\frac{p(c_s) \cdot p(\overrightarrow{x} \mid c_s)}{p(c_s) \cdot p(\overrightarrow{x} \mid c_s) + p(c_s) \cdot p(\overrightarrow{x} \mid c_h)} > T, \tag{17}$$

where the threshold T = 0.5, $p(c_s)$ and $p(c_h)$ refer to the probability of the spam and ham classes respectively, and

$$p(\overrightarrow{x} \mid c_s) = \prod_{i=1}^{k} p(t_i \mid c_s)^x \cdot (1 - p(t_i \mid c_s)^x)^{1-x}, \tag{18}$$

where $p(t_i \mid c_s)$ is the probability of the occurrence of the i^{th} feature given c_s.

Two schemes for estimating $p(t_i \mid c)$ were employed. The first used the weak SLWE estimator, and the second used a common method used with NB classifiers, i.e., maximum likelihood estimation(MLE).

For the SLWE, estimates for both $p(t \mid c_s)$ and $p(t \mid c_h)$ were maintained and depending on whether a message was classified as ham or spam, the probabilities were updated according to the following SWLE rule:

Given that a document d associated with class c_s,

$$p'(t_i \mid c_s) = \begin{cases} \lambda p(t_i \mid c_s) & \text{if } t_i \text{ does not occur in } d, \\ 1 - \lambda p(\bar{t}_i \mid c_s) & \text{if } t_i \text{ occurs in } d. \end{cases} \tag{19}$$

This procedure was followed for both spam and ham messages.

For the MLE, estimates for both $p(t \mid c_s)$ and $p(t \mid c_h)$ were also maintained. However, in order to deal with the problem of zero probability estimates due to rare words, Laplace smoothing, hence

$$p(t \mid c) = \frac{1 + N_{tc}}{2 + N_c}, \tag{20}$$

where N_{tc} is the number of emails belonging to class c that contain the term t, and N_c is the total number of emails belonging to the class c.

Since the quantity $P(c)$ is unknown, for the MLE filter, it was estimated by computing $\frac{N_c}{N}$, where N_c is the number of training messages labeled as c, where $c \in \{ham, spam\}$ and N is the total number of training messages. For the SLWE filter, weak estimation was also used in estimating $P(c)$ according to the following rule:

given a message d,

$$p'(c_s) = \begin{cases} \lambda p(c_s) & \text{if } d \text{ belongs to class } c_h, \\ 1 - \lambda p(c_h) & \text{if } d \text{ belongs to class } c_s. \end{cases} \tag{21}$$

For each dataset, the estimates were updated in intervals of k messages. Specifically the following updating procedure was followed:

1. The first k messages were used for the feature selection and for estimation of term occurrences in ham and spam emails.

2. The estimates learned from the previous step were used to classify the next k messages.

3. After classifying every k^{th} message of each interval, the estimates were updated using the true labels of the last k messages. This, in a way, emulates how users classify their emails only after several messages have arrived. Thus, the filter uses the estimates obtained from the last update to classify new messages, and only updates its estimates after the user applies the true labels.

For the SLWE, $\lambda = 0.95$ was used to estimate both $p(t \mid c)$ and $P(c)$. For both filters, the first 100 messages were used for initial training and 200 words from this set were selected using the IG as the features to be used throughout the classification process. Also, after every 100^{th} message, the true labels of the past 100 messages were given to the filters in order to update their estimates. The tests were repeated using different values for the threshold in (17).

5 Results

The following methods were used in evaluating the results. Since the the classification method used entailed the use of a threshold, ROC (Receiver Operating Characteristic) curves were used to present the performance of each filter across several threshold values.

Because the filter returns a score that is compared to the threshold to determine whether an observation should be classified as ham or spam, the scores are tested against several threshold values. This is done to determine the ham and spam misclassifications that would have resulted if a certain threshold was used. Furthermore, the following values were computed:

$$\text{True Positive Rate, } \eta_{tp} = \frac{n_{S,S}}{n_S}, \qquad \text{and} \tag{22}$$

$$\text{False Positive Rate, } \eta_{fp} = 1 - \frac{n_{H,H}}{n_H}, \tag{23}$$

where $n_{S,S}$ refers to number of spam messages correctly classified as spam, and n_S is the total number of spam messages, while $n_{H,H}$ refers to the total number of correctly classified ham message and n_H is the total number of ham messages. Figures 2 shows the ROC curves of the two filters for each of the data sets Enron-2, Enron-3 and Enron-4 respectively. To plot the curves, different values for the threshold T were used, and the resulting ham and spam recalls were recorded.

The ROC curves show how much spam can be filtered out (true negative rate or spam recall) given that the user can only tolerate a certain false positive rate. Hence, the goal is to reach the upper left corner which signifies the best performance [11]. This means that a perfect filter will tolerate 0 false positives but still

detect 100% of true positives. For example, in Figure 2 Enron-4, given that the user can only tolerate 6% of 'lost' legitimate mail, the SLWE filter can still manage to detect 94% of spam, whereas the MLE filter can only detect 75% of spam.

Observe that for datasets Enron-2, Enron-4, Enron-5 and Enron-6, the figures show that the SLWE filter has a superior performance than the MLE filter. However, for Enron-1, MLE performed better while for Enron-3, there is no big difference in the performance.

Since the values associated with the ROC curve are computed from the final results of each test, it gives us very little information on how the filters performed for each interval. Thus, we consider another approach to evaluate the per-interval performance of the filters.

The *Recall* and *Precision* metrics are commonly-used measures in information retrieval. Recall refers to the ratio of relevant retrieved documents over the total number of relevant documents. Precision, on the other hand, refers to ratio of relevant retrieved documents over all retrieved documents (24). In the context of spam filtering, either of the ham or spam classes can be considered as the relevant class. Thus, recall and precision can be computed for both ham and spam emails. 3 shows the overall recall and precision resulting from both SLWE and MLE filters.

$$precision = \frac{tp}{tp + fp} \tag{24}$$

However, according to Guzella and Caminhas [7], the false positive rate, which is a very important consideration in evaluating spam filters, cannot be directly assessed through the spam precision measure. Hence, the results were assessed by means of the Spam Recall (SR) and Ham Recall (HR). It is also a convenient way of evaluating results when a filter is incrementally updated, since the performance can be observed for each interval. The SR is equal to the true Positive rate given by (22), while the HR is $1 - \eta_{fp}$, where η_{fp} is the false positive rate given by (23).

Figure 4 shows the periodical performance in terms of Ham Recall and Spam Recall of SLWE and MLE filters tested on all six datasets. However, note that *Recall* is affected by the ratio of ham and spam emails as well as the chosen threshold. Hence, it is best to evaluate performance based on different kinds of measures.

Results show that the SLWE filter performs better than the MLE filter when conditions are dynamic, that is, when the data is characterized by changing underlying probability distributions through time. The question now is which dynamic aspects of the data affect the estimates and thus affect the performance of the classifier. Although the ratio of ham and spam emails may be considered as dynamic since in the given datasets it changes per interval, still it does not explain why in Enron-1 and Enron-3 MLE had similar or better performance than SLWE. If we look at the per-interval distribution of ham and spam emails in the datasets, we can

see that the changes in the ratio through time is similar for all the datasets.

Since the probabilities estimated are occurrence of terms given the classes where the terms that are checked are the ones that are included in the feature vector, it is very likely that there are aspects in the presence of features in the observations that are changing through time. Because of this, the occurrence of terms in the emails were observed. The term count of an email, TC is computed as:

$$TC = \sum_{i=1}^{t} x_i \tag{25}$$

where t is the number of terms in the feature vector, $x_i = \{0, 1\}$ where 1 signifies occurence of a term in an email and 0 otherwise. Moreover, the average term count TC_{ave} is the average TC of n emails.

$$TC_{ave} = \frac{1}{n} \sum_{i=1}^{n} TC_i \tag{26}$$

Figure 5 shows the plot of average TCs for every interval of 100 emails. Furthermore, Table 2 shows average term counts for every quarter in each dataset. Note that TC here is determined for the spam the ham classes. Columns 4 and 6 refer to average term count over $t = 200$ the number of terms in the feature vector.

Notice that for Enron-2, from the quarter 1 to 2 the average spam TC has shifted from 25.19 to 15.25. Figure 5 also shows that during interval 19, average spam TC suddenly dropped. Figure 4 shows that within this same interval, spam recall for MLE has dropped to 0.48 but recovered again after around 10 intervals. Similarly, the SLWE spam recall has dropped to 0.73 but recovered to 0.95 after just one interval. Keep in mind however that a stable average TC does not mean that the same features in the emails are always present. It is possible that previously present terms become absent and replaced by other terms resulting in the same average TC.

For Enron-4 from quarter 2 to 3, the average ham TC has changed from 16.81 to 4.87. Figure 5 also shows that during interval 31, average ham TC suddenly dropped. Figure 4 shows that around this time, in interval 28, ham recall for MLE has dropped to 0.67 and then dropped further to 0 in interval 30 and recovered in interval 37. The SLWE filter also experienced a drop to 0.73 in ham recall in interval 28 but unlike MLE, it has recovered to 0.88 in interval 30 and then to 1.0 in interval 31.

For Enron-5 from quarter 1 to 2, average spam TC has changed from 17.61 to 10.84. Enron-5 ham and spam recalls in Figure 4 don't show significant drops like the former cases. Notice however that although MLE seems to have better spam recall than SLWE, its ham recall performs poorly compared to the SWLE filter. Since recall is a measurement affected by the ratio of ham and spam classes as well

Table 2: Token Counts

Dataset	Qtr	spam TC_{ave}	spam $\frac{TC_{ave}}{t}$	ham TC_{ave}	ham $\frac{TC_{ave}}{t}$
Enron-1	1	12.04	0.06	14.47	0.07
	2	10.99	0.05	12.90	0.06
	3	11.31	0.06	12.84	0.06
	4	12.94	0.06	12.88	0.06
Enron-2	1	*25.19*	*0.13*	16.38	0.08
	2	*15.25*	*0.08*	16.77	0.08
	3	12.32	0.06	16.88	0.08
	4	12.78	0.06	17.58	0.09
Enron-3	1	10.82	0.05	18.63	0.09
	2	13.14	0.07	17.57	0.09
	3	14.51	0.07	22.46	0.11
	4	14.21	0.07	17.36	0.09
Enron-4	1	14.58	0.07	20.21	0.1
	2	11.78	0.06	*16.81*	*0.08*
	3	13.47	0.07	*4.87*	*0.02*
	4	14.96	0.07	4.95	0.02
Enron-5	1	*17.65*	*0.09*	18.69	0.09
	2	*10.84*	*0.05*	18.09	0.09
	3	8.28	0.04	16.75	0.08
	4	9.95	0.05	15.56	0.08
Enron-6	1	10.35	0.05	20.56	0.1
	2	11.67	0.06	21.12	0.11
	3	13.88	0.07	18.73	0.09
	4	13.33	0.07	17.75	0.09

as the threshold, it is best to also refer to the ROC curve for evaluation. Indeed, the ROC curve for Enron-5 in Figure 2 shows that SLWE performs better than MLE.

As for Enron-1, where the MLE filter has perfomed better than the SLWE filter, Table 2 shows that the average TC for both ham and spam emails are more stable compared to the other datasets. By comparing the behavior of per interval TC Enron-1 in Figure 5 to Enron-2, Enron-3 and Enron-5, it is not characterized by abrupt changes from one interval to another.

6 Conclusion

Anomaly detection in dynamic social email systems is an important issue that has invited much research attention. The question of how to adequately detect anomaly behaviors is indeed, very challenging. The state-of-the-art in anomaly detection works with the assumption that both normal and anomalous data follow data distributions that are stationary. While this is, in one sense, acceptable, it is still a limitation on the types of detections possible, and sets the boundary on what is currently solvable. In this paper, we have shown that we can apply the theory of weak estimation to anomaly detection in dynamic social environments. Particularly, we have used the Stochastic Learning Weak Estimation (SLWE) approach for spam filtering based on Naive Bayes classification. Preliminary experimental results show that the SLWE-based Multivariate Naive Bayes filter performs quite well compared to the MLE-based filter especially in environments where there are abrupt changes in the distribution of spam and ham emails. As shown in the results, by employing the SLWE, the filter was able to recover from drops in the spam detection rate faster than the MLE filter. In the future, we propose to further explore this approach and investigate its applicability in other dynamic environments.

References

[1] Enron-spam dataset. Downloaded from http://www.aueb.gr/users/ion/data/enron-spam/, 2006.

[2] Open NLP. Website: http://opennlp.sourceforge.net, November 2008.

[3] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, and C.D. Spyropoulos. An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167. ACM New York, NY, USA, 2000.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys (To Appear)*, 2009.

[5] M. Chopra, M.V. Martin, L. Rueda, and P.C.K. Hung. Toward New Paradigms to Combating Internet Child Pornography. In *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on*, pages 1012–1015, 2006.

[6] John Didion. The Java WordNet Library, 2004.

[7] T. Guzella and W. Caminhas. A review of machine learning approaches to Spam filtering. *Expert Systems With Applications (To Appear)*, 2009.

[8] JS Kong, BA Rezaei, N. Sarshar, VP Roychowdhury, and PO Boykin. Collaborative spam filtering using e-mail networks. *Computer*, 39(8):67–73, 2006.

[9] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2nd. edition, 2003.

[10] Colin McGregor. Controlling spam with spamassassin. *Linux J.*, 2007(153):9, 2007.

[11] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive bayes-which naive bayes. In *Third Conference on Email and Anti-Spam (CEAS)*, pages 125–134, 2006.

[12] A.G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[13] Dunja Mladenić, Janez Brank, Marko Grobelnik, and Natasa Milic-Frayling. Feature selection using linear classifier weights: interaction with classification models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 234–241, New York, NY, USA, 2004. ACM.

[14] K. Narendra and M. Thathachar. *Learning Automata. An Introduction*. Prentice Hall, 1989.

[15] J. Norris. *Markov Chains*. Springer Verlag, 1999.

[16] B.J. Oommen and S. Misra. A Fault-Tolerant Routing Algorithm for Mobile Ad Hoc Networks Using a Stochastic Learning-Based Weak Estimation Procedure. In *Proceedings of the 2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 31–37. IEEE Computer Society Washington, DC, USA, 2006.

[17] B.J. Oommen and L. Rueda. Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments. *Pattern Recognition*, 39(3):328–341, 2006.

[18] L. Rueda and BJ Oommen. Stochastic Automata-Based Estimators for Adaptively Compressing Files With Nonstationary Distributions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(5):1196–1200, 2006.

[19] F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[20] B. Wang, G.J.F. Jones, and W. Pan. Using online linear classifiers to filter spam emails. *Pattern Analysis & Applications*, 9(4):339–351, 2006.

[21] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, England, 1989.

[22] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[23] J. Zhan, J. Oommen, and J. Crisostmo. Anomaly Detection in Dynamic Social Email Systems. In *Proceedings of IEEE International Conference on Social Computing*. Vancouver, Canada, 2009.

[24] Le Zhang, Jingbo Zhu, and Tianshun Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269, 2004.

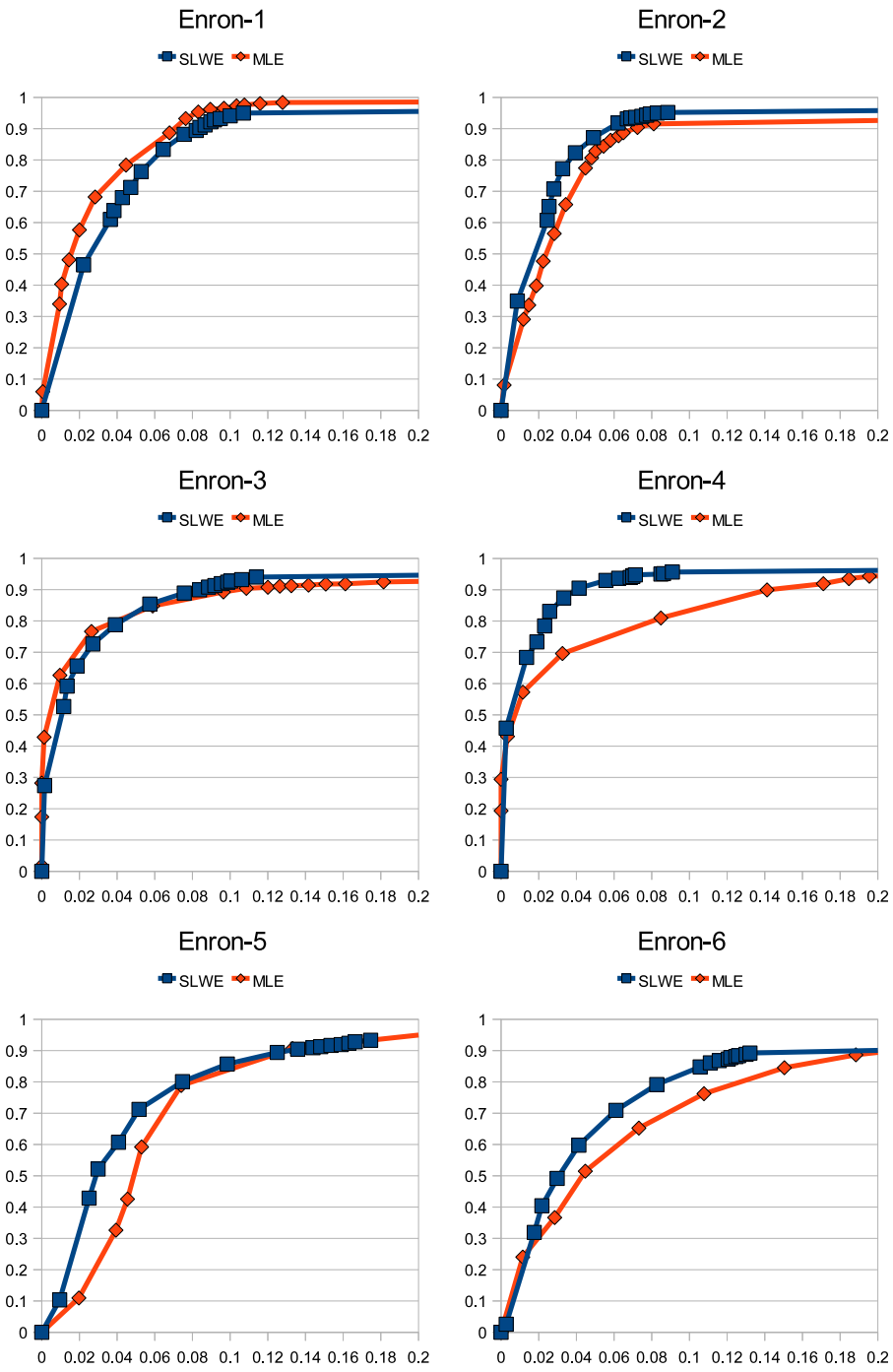

Figure 2: The ROC curves for the MLE and SLWE filters for each of the six Enron-spam datasets. The x-axes represent the false positive rate while the y-axes represent the true positive rate.

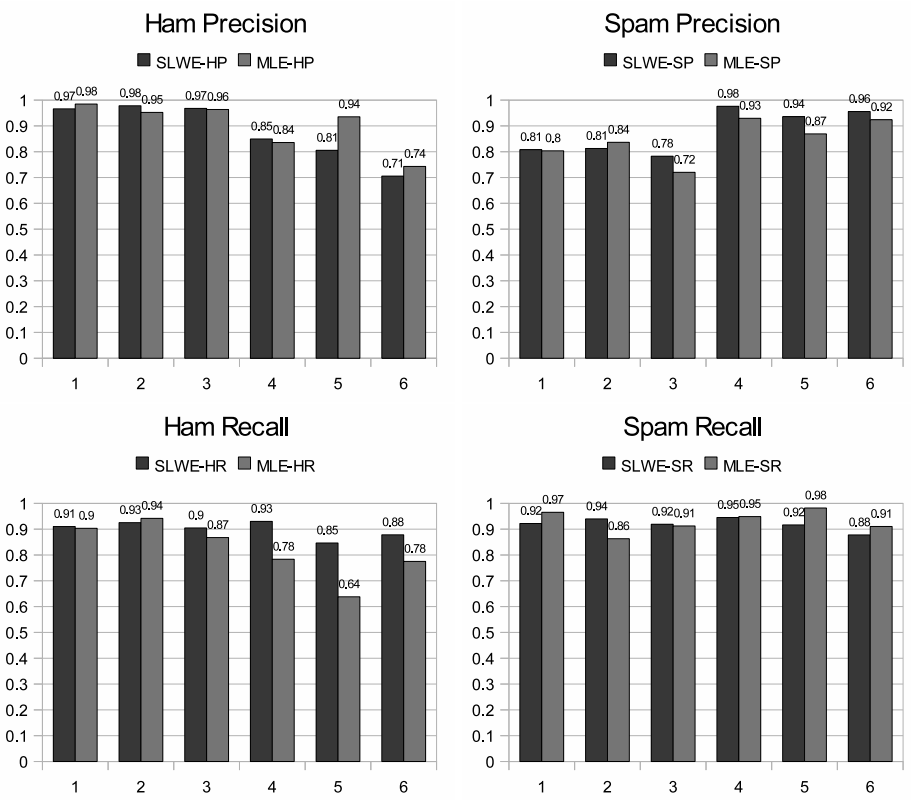

Figure 3: Overall ham and spam recall and precision measures for all six datasets.

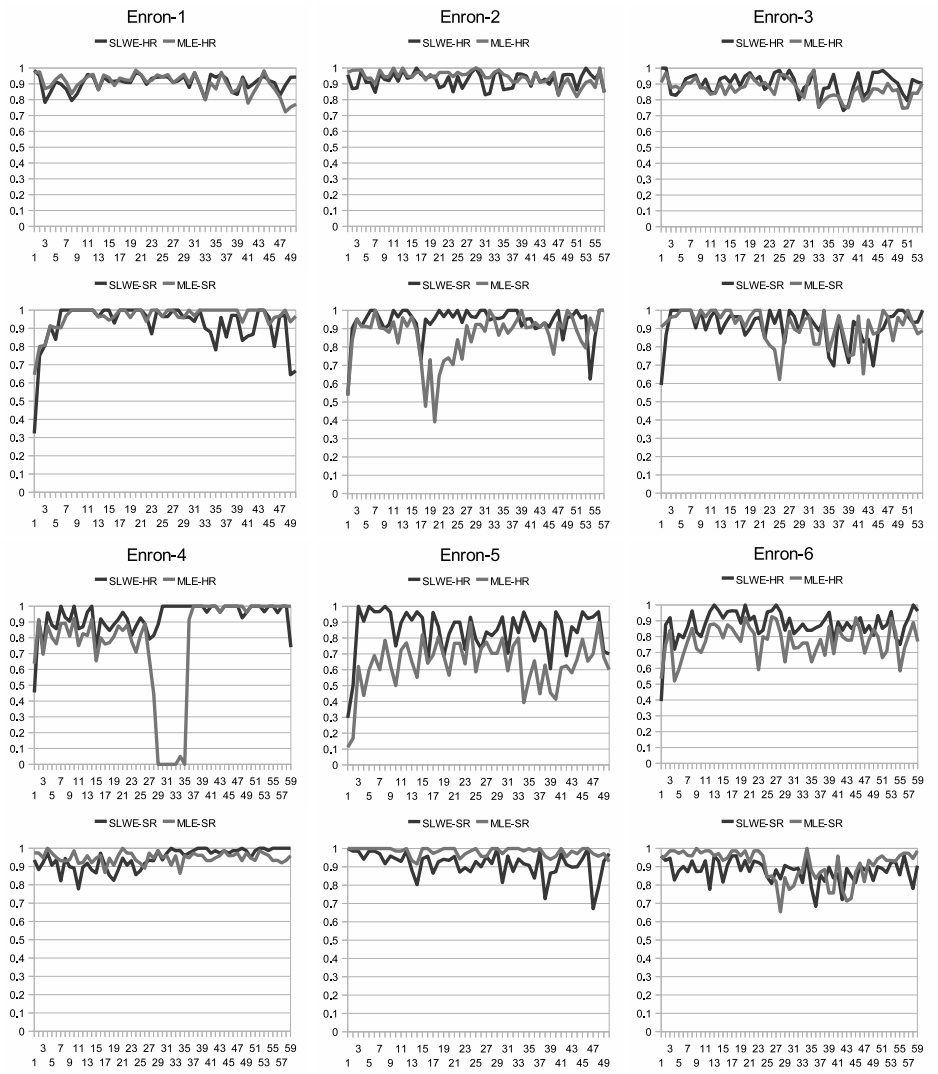

Figure 4: The Ham Recall (top) and Spam Recall (bottom) for both filters on the Enron-spam datasets. The x-axis denotes the interval after which the measurements were taken.

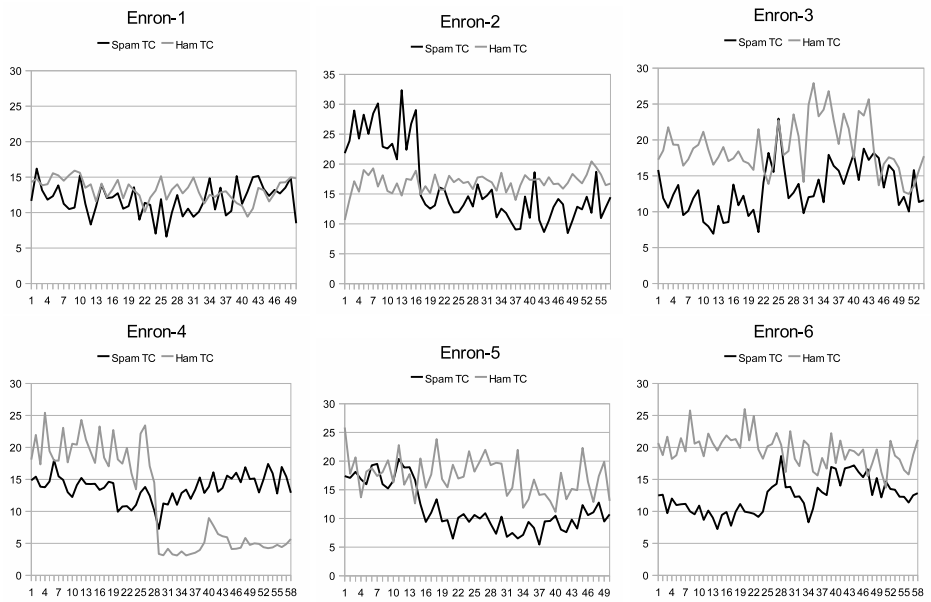

Figure 5: Average token counts (TC_{ave}) for ham and spam classes for every 100 emails.