2006

Single-Class Learning for Spam Filtering: An Ensemble Approach

Tsang-Hsiang Cheng
Southern Taiwan University of Technology

Chih-Ping Wei
National Tsing Hua University, cts@mail.stut.edu.tw

Single-Class Learning for Spam Filtering: An Ensemble Approach

Tsang-Hsiang Cheng
Department of Business Administration
Southern Taiwan University of Technology
Tainan, Taiwan, R.O.C.

Chih-Ping Wei
Institute of Technology Management
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
cts@mail.stut.edu.tw

Abstract

Spam, also known as Unsolicited Commercial Email (UCE), has been an increasingly annoying problem to individuals and organizations. Most of prior research formulated spam filtering as a classical text categorization task, in which training examples must include both spam emails (positive examples) and legitimate mails (negatives). However, in many spam filtering scenarios, obtaining legitimate emails for training purpose is more difficult than collecting spam and unclassified emails. Hence, it would be more appropriate to construct a classification model for spam filtering from positive (i.e., spam emails) and unlabeled instances only; i.e., training a spam filter without any legitimate emails as negative training examples. Several single-class learning techniques that include PNB and PEBL have been proposed in the literature. However, they incur fundamental limitations when applying to spam filtering. In this study, we propose and develop an ensemble approach, referred to as E2, to address the limitations of PNB and PEBL. Specifically, we follow the two-stage framework of PEBL and extend each stage with an ensemble strategy. Our empirical evaluation results on two spam-filtering corpora suggest that the proposed E2 technique exhibits more stable and reliable performance than its benchmark techniques (i.e., PNB and PEBL).

Keywords: Spam Filtering, Single-Class Learning, Ensemble Approach, Text Categorization, Learning from Positive and Unlabeled Examples, Partially Supervised Classification

1. Introduction

With the advancement and proliferation of information and networking technologies, individuals and organizations have increasingly relied on emails for communications and information sharing. While enjoying this efficient and convenient communication medium, individuals and organizations are suffering from spam emails that have increased dramatically in the past few years. Spam, also known as Unsolicited Commercial Email (UCE) and Unsolicited Bulk Email (UBE), is Internet mail that is sent to a group of recipients who have not requested it (Boykin and Roychowdhury 2005; Whitworth and Whitworth 2004; Zhang et al. 2004). These unsolicited emails not only

consume users' time and energy to identify and remove them, but also cause many annoying problems such as filling mailboxes, engulfing important personal emails, and wasting network bandwidth (Zhang et al. 2004). In some cases, spam emails may even be harmful; e.g., spam emails containing pornographic materials may be read by children (Zhang et al. 2004; Zorkadis et al. 2005).

The volume of spam emails is growing increasingly because sending e-mails has nearly no cost and spammers can obtain e-mail addresses easily via email address harvesting tools. Jupiter Research (Taylor 2003) estimates that 4.9 trillion spam emails were sent worldwide in 2003. In addition, according to Brightmail (http://www.brightmail.com), a vendor for anti-spam software, the volume of spam as a percentage of all emails rose from 8% in January 2001 to 56% in November 2003. A Ferris Research report (http://entmag.com/news/article.asp?EditorialsID=5651) estimates that spam emails cost US companies $10 billion in 2003, where the cost estimate includes loss of user productivity, consumption of information technology resources, and help desk costs. A recent study (Fallows 2003) showed that 52% of email users say spam has made them less trusting of email, and 25% say that the volume of spam has reduced their email use. To reduce the aforementioned costs that result from spam emails, effective spam filtering that automatically discriminates spam emails from legitimate email is essential to individuals and organizations.

Spam filtering has been considered as a classical text categorization task (Pantel and Lin 1998; Drucker et al. 1999; Sebastiani 2002; Weiss et al. 1999), although other approaches have also been suggested (e.g., maintaining blacklists of frequent spammers). With this formulation, given a set of training instances that are preclassified as belonging to the *spam* or *legitimate* class, a classification analysis or supervised machine learning algorithm is employed to induce a classification model, which will then be used to classify incoming emails. Common classification analysis algorithms used in the context of spam filtering include Naïve Bayes classifier (Androutsopoulos et al. 2000a; Pantel and Lin 1998; Sahami et al. 1998; Schneider 2003), Support Vector Machines (SVM) (Drucker et al. 1999; Kolcz and Alspector 2001), RIPPER rule induction (Pantel and Lin 1998; Drucker et al. 1999), Rocchio (Drucker et al. 1999), Memory-based reasoning (Cunningham et al. 2003; Sakkis 2001), AdaBoost (Carreras and Márquez 2001; Drucker et al. 1999), and maximum entropy model (Zhang and Yao 2003). While all these algorithms seem appealing, they have an explicit requirement on the set of training examples. That is, the training set must contain instances from both classes (i.e., spam and legitimate). Nevertheless, in most of real world scenarios, obtaining legitimate emails for training purpose is more difficult than collecting spam emails because individuals may be willing to contribute spam emails they have received but generally are reluctant to release their legitimate emails due to privacy concerns. In this case, it would be more appropriate to construct a classification model for spam filtering from positive (i.e., spam emails) and unlabeled instances only.

The described categorization problem is regarded as "single-class learning or classification," "learning from positive and unlabeled examples," "partially supervised classification," and "learning without negative examples" (Comité et al. 1999; Denis et al.

2002; Letouzey et al. 2000; Yu et al. 2004). Prior research has proposed several single-class learning techniques that include the Positive Naïve Bayes (PNB) technique and the Positive Example Based Learning (PEBL) technique. Let the class of positive examples be C_p and that of negatives be C_n. PNB takes as its input a training set of positive examples and a set of unlabeled documents and requires an estimate $\hat{Pr}(C_p)$ of the class prior probability of C_p (Denis et al. 2002). To determine an appropriate class for an unclassified document d_j, PNB relies on $\hat{Pr}(C_p)$ and estimates of the word probabilities $Pr(w_i|C_p)$ for each $w_i \in d_j$ to derive the probability of d_j belonging to the class C_p. Because of the unavailability of negative examples, PNB depends on the set of unlabeled instances and the estimate of the class prior probability of C_n (i.e., $\hat{Pr}(C_n) = 1-\hat{Pr}(C_p)$) to estimate $Pr(w_i|C_n)$ for each $w_i \in d_j$ and then to derive the probability of d_j belonging to the class C_n.

Yu et al. (2004) propose PEBL that adopts a two-stage strategy for learning from positive and unlabeled documents. The Mapping stage uses a rough classifier to identify a set of "strong negative" examples from the unlabeled set of documents. Subsequently, PEBL employs Support Vector Machines (SVM) in the Convergence stage to maximize margin to make a progressively better approximation of the negative class. PEBL iteratively identifies and selects for training purpose more negative examples from the unlabeled set of documents until no more negative examples can be recognized. Finally, PEBL uses the initial positive examples and the negative examples previously identified from the unlabeled documents to train a classifier that will be used for class prediction for future documents.

Although their empirical results are encouraging, these two techniques incur some inherent limitations in the context of spam filtering. As mentioned, PNB involves an estimate of $\hat{Pr}(C_p)$ whose accuracy greatly affects the classification accuracy of PNB. However, in the spam filtering application, the percentage of spam emails is highly variable over time (Taylor 2003; Denis et al. 2002). In this case, an accurate estimate of $\hat{Pr}(C_p)$ is difficult to obtain, possibly limiting the applicability of PNB for spam filtering. On the other hand, PEBL's effectiveness highly depends on the accuracy of the initial set of "strong negative" examples identified by the rough classifier. If the initial "strong negative" examples are not trustworthy, the accuracy of negative examples selected in the Convergence stage of PEBL may gradually deteriorate over iterations; hence, possibly impairing the effectiveness of PEBL.

In response, in this study, we propose an ensemble approach, referred to as E2, to address the PNB's sensitivity to the estimate of $\hat{Pr}(C_p)$ and the PEBL's susceptibility to the accuracy of the initial set of "strong negative" examples identified in the Mapping stage. Specifically, we follow the two-stage framework of PEBL and extend each stage with an ensemble strategy to provide a more reliable single-class learning technique for spam filtering. Essentially, an ensemble classifier consists of multiple classifiers (referred to as base classifiers) induced from a given set of training examples. When classifying an unseen instance, the ensemble classifier combines the predictions of the base classifiers through voting or other mechanism (Bauer and Kohavi 1999; Breiman 1996; Dietterich

2000; Hansen and Salamon 1990; Opitz and Maclin 1999). Prior empirical results have demonstrated that the ensemble classifier generally attains better classification effectiveness than any individual base classifiers do (Dietterich 2000; Dong and Han 2004; Opitz and Maclin 1999). In this vein, to improve the accuracy of the initial set of "strong negative" examples in the first stage, the proposed E2 technique adopts an ensemble approach that combines the predictions of the rough classifier of PEBL and PNB on the unlabeled examples. In addition, the second stage of the proposed E2 technique constructs an ensemble classifier that adopts SVM, Naïve Bayes, and C4.5 as its base classifiers. In this study, we empirically evaluate the proposed E2 technique with two spam-filtering corpora and include PNB and PEBL as our performance benchmarks.

The remainder of this paper is organized as follows: We review PNB and PEBL for learning from positive and unlabeled documents in Section 2. In Section 3, we depict the proposed E2 technique, including its overall process and algorithmic details. Subsequently, we describe our evaluation design and discuss some important experimental results in Section 4. Finally, we conclude in Section 5 with a summary and some future research directions.

2. Literature Review

In this section, we review PNB and PEBL, two well-known single-class learning algorithms, and depict their limitations to highlight our motivation.

2.1 Positive Naive Bayes (PNB)

Let C_p be the positive class (i.e., spam emails) and C_n be the negative class (i.e., legitimate emails). PNB takes a training set of positive examples PD and a set of unlabeled documents UD as its inputs and involves an estimate $\hat{Pr}(C_p)$ of the class prior probability of C_p (Denis 2002). PNB classifies a document d_j that consists of n words $\{w_1, \ldots, w_n\}$ with possibly multiple occurrences of a word as a member of the class by

$$PNB(d_j) = \underset{C \in \{C_p, C_n\}}{\operatorname{argmax}} \hat{Pr}(C) \prod_{i=1}^{n} \hat{Pr}(w_i|C).$$

The prior probability $\hat{Pr}(C_n)$ of the class C_n is estimated using $1-\hat{Pr}(C_p)$. Furthermore, the positive word probability $Pr(w_i|C_p)$ is estimated by the frequency that w_i occurs in all training documents for the positive class C_p (i.e., PD) divided by all word occurrences for the documents in PD. That is, $\hat{Pr}(w_i|C_p) = \frac{N(w_i, PD)}{N(PD)}$, where $N(w_i, PD)$ is the total number of times w_i occurs in the documents in PD and $N(PD)$ is the total number of word occurrences in PD. If a word w_i in the document d does not appear in any documents in C_p, $Pr(w_i|C_p)$ will become 0. To avoid such an undesired situation caused by the described estimate for $Pr(w_i|C_p)$, PNB adopts the Lidstone's law of succession to smooth the maximum likelihood estimate (Agrawal et al. 2000) and defines $\hat{Pr}(w_i|C_p) = \frac{N(w_i, PD) + \lambda}{N(PD) + \lambda \times |V|}$, where V is the number of distinct features in the training documents, $|V|$ is the cardinality of V, and $\lambda \geq 0$.

Because of the unavailability of negative training examples, the negative word probabilities are estimated from the unlabeled examples as: $Pr(w_i) = Pr(w_i|C_n)\times Pr(C_n) + Pr(w_i|C_p)\times Pr(C_p)$, where $Pr(w_i)$ is the probability that the underlying generative model creates w_i. Accordingly, the negative word probabilities can be derived as $Pr(w_i|C_n) = \frac{Pr(w_i) - Pr(w_i|C_p)\times Pr(C_p)}{1 - Pr(C_p)}$. The probability $Pr(w_i)$ can be estimated on the basis of the set of unlabeled documents by $\hat{Pr}(w_i) = \frac{N(w_i, UD)}{N(UD)}$. Thus, the estimate for negative word probabilities $\hat{Pr}(w_i|C_n)$ can be rewritten as $\hat{Pr}(w_i|C_n) = \frac{N(w_i, UD) - \hat{Pr}(w_i|C_p)\times\hat{Pr}(C_p)\times N(UD)}{(1 - \hat{Pr}(C_p))\times N(UD)}$. Similarly, based on the Lidstone's law of succession, the negative word probability is estimated as $\hat{Pr}(w_i|C_n) = \frac{(N(w_i, UD) - \hat{Pr}(w_i|C_p)\times\hat{Pr}(C_p)\times N(UD)) + \lambda}{(1 - \hat{Pr}(C_p))\times N(UD) + \lambda\times|V|}$.

Due to the unavailability of negative training examples, most of the estimates involved in PNB are derived from the estimate $\hat{Pr}(C_p)$. Thus, the effectiveness of PNB is greatly affected by the accuracy of $\hat{Pr}(C_p)$. However, in the spam filtering application, the percentage of spam emails is highly variable over time (Taylor 2003; Denis et al. 2002). In this case, an accurate estimate of $\hat{Pr}(C_p)$ is difficult to obtain and dynamic adjustment of the estimate of $\hat{Pr}(C_p)$ that conforms to the true class distribution of the current situation is even more difficult, possibly limiting the applicability of PNB for spam filtering.

2.2 Positive Example Based Learning (PEBL)

PEBL attempts to induce a classification model that can differentiate the boundary of the positive and negative classes on the basis of a training set of positive examples *PD* and a set of unlabeled documents *UD* (Yu et al. 2004). PEBL adopts a two-stage framework, including the Mapping and the Convergence stage. In the Mapping stage, PEBL employs a rough classifier that draws an initial approximation of "strong negative" examples. Specifically, PEBL first identifies "strong positive" features by comparing the frequencies of features within the positive training and unlabeled examples. For example, a feature is considered as a "strong positive" feature if it occurs in more than α% of the positive training examples but only in β% of unlabeled examples in *UD*. On the basis of the identified list of the "strong positive" features, the unlabeled documents in *UD* that do not contain any of the "strong positive" features are selected and regarded as "strong negative" examples. The remaining documents in *UD* are referred to as "plausible positive" examples.

In the Convergence stage, PEBL constructs an initial classifier based on the positive training examples and the "strong negative" examples identified in the previous stage. Subsequently, PEBL iteratively detects and includes more negative examples from the unlabeled examples using Support Vector Machines (SVM). At each iteration, PEBL

employs the classification model induced in the previous iteration to classify the current set of "plausible positive" examples into the positive or negative class. Afterward, PEBL expands the set of negative examples by incorporating the negative examples identified at this iteration and, accordingly, reconstructs a new classification model using SVM. The set of documents that are classified into the positive class at this iteration becomes the set of "plausible positive" examples for the next iteration. PEBL repeats the negative example selection and the classification model reconstruction process until PEBL cannot find any negative examples from the unlabeled examples in *UD*. As the result of the Convergence stage of PEBL, the class boundary eventually converges to the plausible boundary of the positive class in the feature space.

Evidently, the effectiveness of PEBL highly depends on the accuracy of the initial set of "strong negative" examples identified by the rough classifier in the Mapping stage. If the initial set of "strong negative" examples indeed encompasses true positive examples, the accuracy of negative examples identifies in the Convergence stage of PEBL may gradually deteriorate over iterations and the resulting effectiveness of PEBL will be degraded.

3. Design of E2

To address the aforementioned limitations of PNB and PEBL, we propose an ensemble approach, referred to as E2, for single-class learning for spam filtering. Specifically, we follow the two-stage framework of PEBL and extend each stage with an ensemble strategy. Figure 1 illustrates the overall process of the proposed E2 technique that consists of two main stages, i.e., Mapping via Ensemble and Convergence via Ensemble.

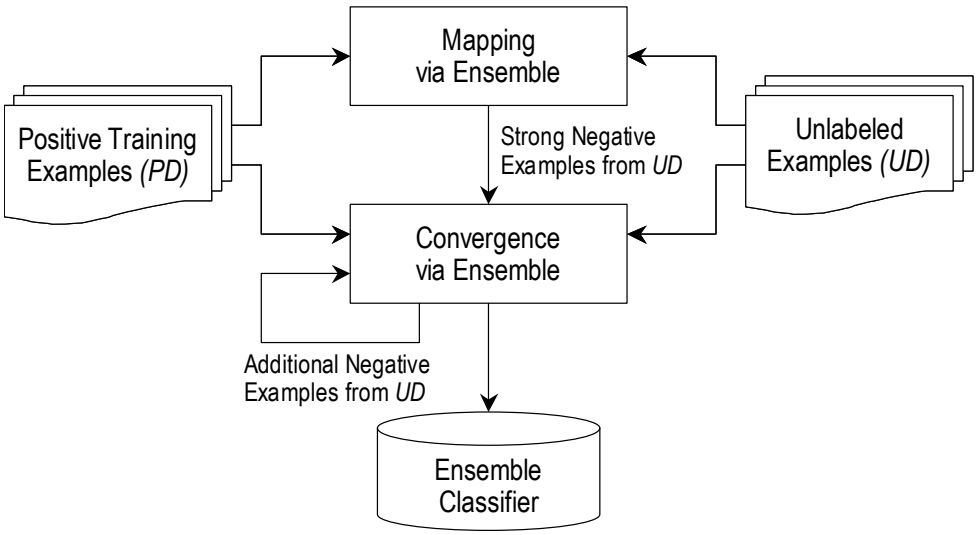


Figure 1: Overall Process of the E2 Technique

3.1 Mapping via Ensemble

The Mapping via Ensemble stage starts with feature extraction and selection. In this study, we use all words in each document in the set of positive training examples *PD* or the set of unlabeled examples *UD* as the features of the document; i.e., without feature selection. We employ the Porter stemmer (Porter 1980) and perform stemming that removes the

suffixes and prefixes of words in documents. Subsequently, we use the bag-of-words scheme for document representation. That is, each document (in *PD* or *UD*) is represented as a feature vector, where $w_1, \ldots, w_n$ are the weights of features $f_1, \ldots, f_n$ and w_i is the term frequency of f_i in the document.

Subsequently, the Mapping via Ensemble stage employs an ensemble of two classifiers for identifying strong negative examples from *UD*. Specifically, we use the rough classifier (employed by PEBL) and PNB and combine their predictions on the unlabeled examples to form the initial set of "strong negative" examples. On the basis of the positive examples in *PD*, the rough classifier of the original PEBL technique identifies a set of "strong negative" examples from *UD*. The rough classifier of PEBL first identifies "strong positive" features from positive and unlabeled examples by comparing the frequency of features within *PD* and *UD*. Specifically, in this study, two parameters are involved: positive threshold (α_{PT}) and unlabeled threshold (α_{UT}). Let $Pr(f_i, PD)$ is the probability of a feature f_i that occurs in the positive examples and $Pr(f_i, UD)$ is the probability of f_i appearing in the unlabeled examples. If $Pr(f_i, PD) > \alpha_{PT}$ and $Pr(f_i, UD) < \alpha_{UT}$, we consider f_i as a "strong positive" feature. Accordingly, we construct from *PD* and *UD* a list of "strong positive" features with respect to α_{PT} and α_{UT}. With the use of such list, an unlabeled example in *UD* that does not contain any of the "strong positive" features is identified as a "strong negative" example by the rough classifier of PEBL.

Similarly, PNB, on the basis of the estimate $\hat{Pr}(C_p)$, is employed to classify the unlabeled examples in *UD* into a subset of negative examples and a subset of positive ones. Consequently, the decision combination step of the Mapping via Ensemble stage combines the classification results of the two base classifiers (i.e., the rough classifier and PNB) and selects as "strong negatives" those unlabeled examples that are classified by both base classifiers as negative ones. The use of the consensus-based strategy could improve the accuracy of the initial set of "strong negative" examples. Thus, the PNB's sensitivity to the accuracy of the estimate of $\hat{Pr}(C_p)$ and the PEBL's susceptibility to the accuracy of the initial set of "strong negative" examples can be mitigated. As a result, the Mapping via Ensemble stage produces the initial set of "strong negative" examples (referred to as N_1) and retains the remaining unlabeled examples in the plausible positive examples (referred to as P_1).

3.2 Convergence via Ensemble

The Convergence via Ensemble stage constructs an ensemble classifier by adopting SVM, Naive Bayes, and C4.5 as its base classifiers. Initially (i.e., at the iteration 1), both the positive training examples (*PD*) and the strong negative examples (N_1) yielded from the Mapping via Ensemble stage form the training set TS_1 for the three base classifiers (i.e., $TS_1 = PD \cup N_1$). Each classifier then induces a classification model from the current training set and attempts to classify each plausible positive example in P_1. When all base classifiers suggest the decision of the negative class for a plausible positive example, this example is then considered as a negative example; thus, forming an additional set of negative examples N_2. The remaining plausible positive examples are then assigned to P_2. At the next iteration i, the training set $TS_i = TS_{i-1} \cup N_i$ from which the base classifiers are

re-trained to classify the plausible positive examples P_i into N_{i+1} and P_{i+1}. This process continues until no more negative examples can be extracted (i.e., $N_{i+1} = \varnothing$). Consequently, an ensemble classifier that consists of three base classifiers is obtained and will be used for classify any future unclassified instances.

3.3 Prediction

When receiving an unclassified document (i.e., email in this study), the proposed E2 technique uses the ensemble classifier constructed in the Convergence via Ensemble stage for classification purpose. We employ a voting scheme to arrive at an overall classification decision from individual decisions suggested by the three base classifiers. Specifically, if two or more base classifiers assign the target unclassified document to the positive class (i.e., spam), it will be considered as belonging to the positive class; otherwise, it will be assigned to the negative class (i.e., legitimate email).

4. Empirical Evaluation

This section reports our empirical evaluation of the proposed E2 technique. We highlight our evaluation design that includes the spam-filtering corpora used for evaluation purpose, the evaluation procedure and evaluation criteria, and then discuss important comparative analysis results.

4.1 Spam-filtering Corpora

Our empirical evaluation employs two public spam-filtering corpora, namely LingSpam and PU1, contributed by Androutsopoulos et al. (2000b). The LingSpam corpus contains a total of 2893 emails, where 481 (16.6%) are spam (positive) and 2412 (83.4%) are legitimate emails (negative). For the 481 spam emails, attachments, HTML tags, and duplicate spam emails received on the same day were not included. The PU1 corpus consists of 1099 emails that include 481 (43.8%) spam messages and 618 messages (56.2%) legitimate emails. The 481 spam emails were collected by the corpus author over a period of 22 months, excluding non-English emails and the duplicate spam emails received on the same day. The other 618 legitimate emails were selected from 1182 emails, which came from authors' colleagues and friends.

4.2 Evaluation Procedure and Criteria

In the study, we assume that there exist only positive and unlabeled examples for training purpose. Thus, we used 40% of spam emails in each spam-filtering corpus as the positive training examples and another 40% of spam emails and 40% of legitimate emails as the unlabeled examples. The remaining 20% of spam emails and 60% of legitimate emails in the spam corpus were used for testing purpose.

To minimize potential biases that may result from the randomized sampling process and to obtain more reliable performance estimates, we performed this validation process thirty times. The overall effectiveness of each single-class learning technique examined (including E2 and its benchmark techniques) was estimated by averaging the performance obtained from the 30 individual validation trials. We used three measures to evaluate the effectiveness of each technique under investigation, including precision (for spam), recall (for spam), and overall accuracy.

4.3 Parameter Tuning Results

We first conducted parameter-tuning experiments to determine appropriate values for the parameters involved in each technique under examination. The rough classifier in the Mapping stage of PEBL as well as in the Mapping via Ensemble stage of E2 involves the parameters α_{PT} (positive threshold for identifying "strong positive" features) and α_{UT} (unlabeled threshold). In addition, PNB, our benchmark technique that is also used as a base classifier in the Mapping via Ensemble stage of E2, involves the parameter λ for smoothing the maximum likelihood estimate of word probability given a class. Finally, we also need to determine an appropriate value for λ that is required by the Naive Bayes classifier employed as a base classifier of the Convergence via Ensemble stage in E2.

We investigated the range of values for α_{PT}, ranging from 0.1 to 0.7 in increments of 0.1. Given a specific value for α_{PT}, we also examined the range of values for α_{UT}, ranging from 0.1 to the value for α_{PT} in increments of 0.1. For both spam-filtering corpora, our tuning results suggested that when both α_{PT} and α_{UT} were set to 0.5, PEBL achieved the highest accuracy and recall rates while maintaining a satisfactorily high precision rate. Thus, we decided on 0.5 for α_{PT} and α_{UT} for subsequent experiments.

When tuning the parameter λ for PNB, we set $\hat{Pr}(C_p)$ as 0.5 and investigated different values for λ, ranging from 0.3 to 3.9 in increments of 0.3. Overall, trading off between precision and recall rates, we selected 2.7 and 2.1 for λ for the LingSpam and PU1 corpora, respectively. Because the proposed E2 technique involves the Naïve Bayes classifier in the Convergence via Ensemble stage, we need to determine an appropriate value for λ. For the parameters involved in the Mapping via Ensemble stage, we adopted the values determined previously for α_{PT} and α_{UT} for the rough classifier (i.e., 0.5 and 0.5 respectively) and λ for PNB (i.e., 2.7 for LingSpam and 2.1 for PU1). We investigated different values for λ (required by the Naïve Bayes classifier in the Convergence via Ensemble stage) ranging from 1.0 to 6.5 in increments of 0.5. Trading off between recall and precision rates, we decided on 3.0 and 1.0 for λ for the LingSpam and PU1 corpora respectively. Table 1 summarizes all parameter values determined for the three techniques examined across the two different spam-filtering corpora.

Table 1: Summary of Tuning Results

Parameters	LingSpam	PU1
α_{PT} (for PEBL and E2)	0.5	0.5
α_{UT} (for PEBL and E2)	0.5	0.5
λ (for PNB and PNB in Mapping via Ensemble stage of E2)	2.7	2.1
λ (for Naïve Bayes in Convergence via Ensemble stage of E2)	3.0	1.0

4.4 Comparative Evaluation Results

On the basis of the parameter values determined in the tuning experiments, we evaluated the performance of the proposed E2 technique and the benchmark PEBL and PNB for each spam-filtering corpus. We first set $\hat{Pr}(C_p)$ as 0.5 for PNB and then examined the

effects of different values for $\hat{Pr}(C_p)$ on the effectiveness of PNB and the proposed E2 technique. As we illustrate in Table 2, for the LingSpam corpus, PNB (i.e., 99.47%) appeared to marginally outperform E2 (i.e., 99.29%) in accuracy and both techniques in effect outperformed PEBL (i.e., 97.24%). On the other hand, E2 achieved the highest precision rate (i.e., 97.83%), whereas the precision rate attained by PEBL (i.e., 84.42%) was far worse than that by E2 or PNB (i.e., 96.42%). Finally, the recall rate of E2 recorded at 90.70% and that of PNB and PEBL was 95.19% and 68.51%, respectively. Thus, judged from recall rate, PEBL was the worst and PNB outperformed E2. Overall, for the LingSpam corpus, our proposed E2 technique outperformed PEBL in all performance metrics employed. As Table 2 shows, for the PU1 corpus, E2 achieved a higher accuracy (i.e., 95.59%) than PNB did (i.e., 93.95%). As with the LingSpam corpus, PEBL again attained the lowest accuracy (i.e., 93.00%). On the other hand, PEBL achieved the highest precision rate (i.e., 89.59%), which was higher than that attained by E2 (i.e., 84.15%) and PNB (i.e., 84.92%). However, the recall rate of E2 recorded at 96.90%, which was far better than that recorded by PNB and PEBL (i.e., 86.49% and 75.01%, respectively).

Table 2: Comparative Evaluation Results

	LingSpam			**PU1**		
	Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
E2	99.29%	97.83%	90.70%	95.59%	84.15%	96.96%
PNB	99.47%	96.42%	95.19%	93.95%	84.92%	86.49%
PEBL	97.24%	84.42%	68.51%	93.00%	89.59%	75.01%

Overall, our proposed E2 technique outperformed PEBL across the two spam-filtering corpora. The effectiveness (as measured by recall, precision, and accuracy) attained by E2 was considered comparable to that achieved by PNB for the LingSpam corpus, but better than that achieved by PNB for the PU1 corpus.

4.5 Sensitivity of E2 and PNB to $\hat{Pr}(C_p)$

Because both E2 and PNB involve an estimate of $\hat{Pr}(C_p)$, we further examine the effects of its accuracy on the classification effectiveness of PNB and E2, respectively. Specifically, we simulate various scenarios by setting different values for $\hat{Pr}(C_p)$: close to the true piror probability (i.e., 0.2 for the LingSpam corpus and 0.4 for the PU1 corpus), neutral (i.e., 0.5 that represents a scenario in which PNB's prior probability does not favor any class), and far from the true prior probability (i.e., 0.9 for both spam-filtering corpora). As Table 3 depicts, for the LingSpam, the accuracy of E2 was 99.23%, 99.29%, and 99.32% when $\hat{Pr}(C_p)$ = 0.2, 0.5, and 0.9, respectively. On the other hand, for the same corpus, PNB recorded a classification accuracy at 99.02%, 99.47%, and 98.69% when $\hat{Pr}(C_p)$ = 0.2, 0.5, and 0.9, correspondingly. These results suggested that different values for $\hat{Pr}(C_p)$ appeared to have marginal effects on accuracy for both techniques, although E2 performed slightly better than PNB in accuracy when $\hat{Pr}(C_p)$ = 0.2 and 0.9.

The precision rate of E2 was 97.74%, 97.83%, and 98.14% when $\hat{Pr}(C_p)$ = 0.2, 0.5, 0.9, respectively. These results suggested that the proposed E2 technique was stable over the range of values of $\hat{Pr}(C_p)$ examined. However, the precision rate of PNB decreased from 98.97% when $\hat{Pr}(C_p)$ = 0.2 to 86.50% when $\hat{Pr}(C_p)$ = 0.9. Such a large precision gap suggested that PNB was susceptible to $\hat{Pr}(C_p)$. Moreover, our empirical evaluations also suggested that the recall rate attained by PNB was susceptible to $\hat{Pr}(C_p)$ and that achieved by E2 was less sensitive. Specifically, the recall rate of E2 was 89.70%, 90.70%, and 90.73% when $\hat{Pr}(C_p)$ = 0.2, 0.5, 0.9, respectively. In contrast, PNB recorded a recall rate of 85.21%, 95.19%, and 94.07% when $\hat{Pr}(C_p)$ = 0.2, 0.5, and 0.9, respectively.

Table 3: Sensitivity of E2 and PNB to $\hat{Pr}(C_p)$ for LingSpam

		Close to true $Pr(C_p)$ (i.e., $\hat{Pr}(C_p)$ = 0.2)	**$\hat{Pr}(C_p)$ = 0.5**	**Far from true $Pr(C_p)$ (i.e., $\hat{Pr}(C_p)$ = 0.9)**
E2	**Accuracy**	99.23%	99.29%	99.32%
	Precision	97.74%	97.83%	98.14%
	Recall	89.70%	90.70%	90.73%
PNB	**Accuracy**	99.02%	99.47%	98.66%
	Precision	98.97%	96.42%	86.50%
	Recall	85.21%	95.19%	94.07%

Similar evaluation results were also observed with the PU1 corpus. As Table 4 shows, the accuracy of E2 was 95.65%, 95.59%, and 94.44% when $\hat{Pr}(C_p)$ = 0.4, 0.5, and 0.9, respectively, while that of PNB was 94.40%, 93.95%, and 83.99%, correspondingly. E2 outperformed PNB in accuracy across different $\hat{Pr}(C_p)$ settings. Furthermore, different values for $\hat{Pr}(C_p)$ appeared to have marginal effects on the accuracy of E2, but had remarkable effects on the accuracy of PNB. Similarly, the precision and recall rates achieved by E2 were stable over the range of values of $\hat{Pr}(C_p)$ examined. However, the precision rate of PNB decreased from 88.73% when $\hat{Pr}(C_p)$ = 0.4 to 57.54% when $\hat{Pr}(C_p)$ = 0.9, and the recall rate of PNB varied from 83.72% and 94.28% between these two extreme scenarios.

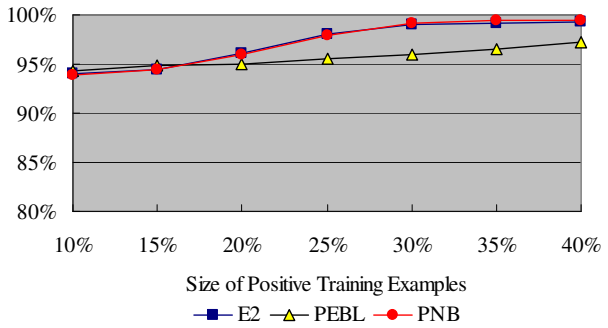
Table 4: Sensitivity of E2 and PNB to $\hat{Pr}(C_p)$ for PU1

		Close to true $Pr(C_p)$ (i.e., $\hat{Pr}(C_p) = 0.4$)	$\hat{Pr}(C_p) = 0.5$	Far from true $Pr(C_p)$ (i.e., $\hat{Pr}(C_p) = 0.9$)
E2	**Accuracy**	95.65%	95.59%	94.44%
	Precision	84.85%	84.15%	80.19%
	Recall	96.29%	96.96%	97.83%
PNB	**Accuracy**	94.40%	93.95%	83.99%
	Precision	88.73%	84.92%	57.54%
	Recall	83.72%	86.49%	94.28%

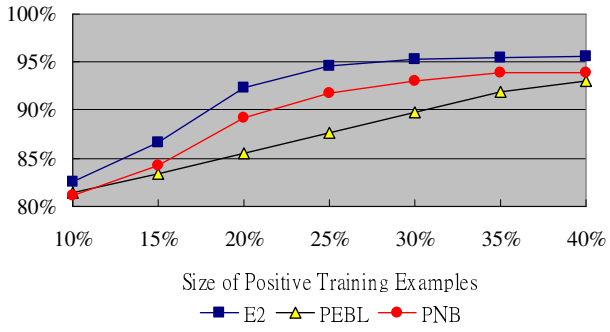
In summary, judged from all performance metrics for both spam-filtering corpora, PNB appeared to be susceptible to $\hat{Pr}(C_p)$ and E2 remained stable over the range of values of $\hat{Pr}(C_p)$ examined. Furthermore, when $\hat{Pr}(C_p)$ was close to or did not deviate too much from the true class probability of the positive class, the accuracy achieved by E2 was largely comparable to that reached by PNB. However, when $\hat{Pr}(C_p)$ was far from the true class probability of the positive class (i.e., in the 0.9 scenario), E2 achieved a noticeably higher accuracy than PNB did. Overall, these evaluation results suggested the utility of the ensemble strategy employed by E2.

4.6 Effects of Size of Positive Training Examples

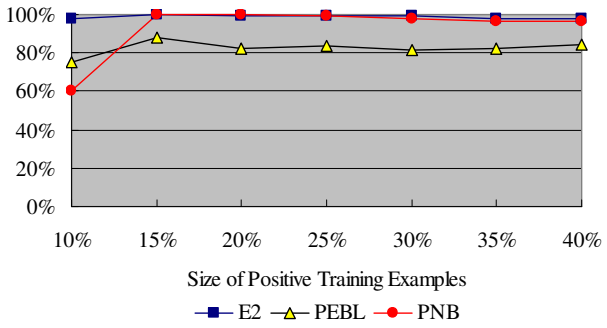
We further examined the sensitivity of different techniques to the size of positive training examples. As mentioned, in all of our previous experiments, we used 40% of spam emails in the spam corpus as the positive training examples, another 40% of spam emails and 40% of legitimate emails as the unlabeled examples for training. In this experiment, we fixed the size of unlabeled examples for training and varied the size of positive training examples ranging from 40% to 10% in decrements of 5%. Using the parameter values determined previously, the resulting evaluation results for LingSpam are shown in Figures 2(a), 2(b) and 2(c) and those for PU1 are shown in Figures 2(d), 2(e), and 2(f).



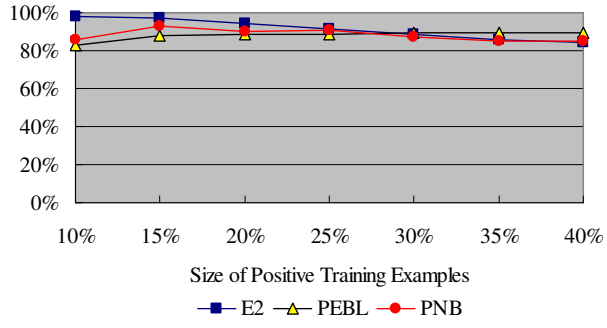
(a) Accuracy with LingSpam



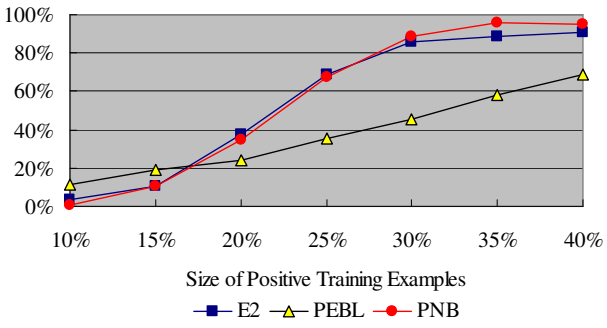
(d) Accuracy with PU1



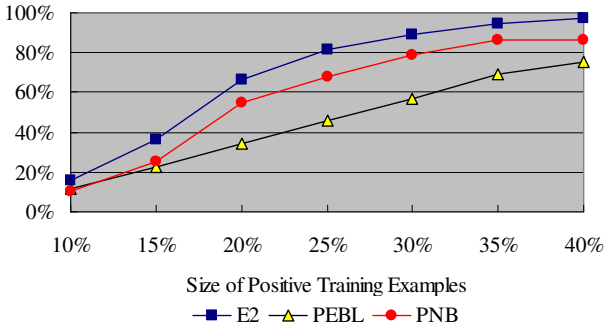
(b) Precision with LingSpam



(e) Precision with PU1



(c) Recall with LingSpam



(f) Recall with PU1

Figure 2: Effects of Size of Positive Training Examples

For the LingSpam corpus, as Figure 2(a) depicts, the accuracy of E2 was largely comparable to that of PNB over the range of sizes of positive training examples examined. The accuracy of PEBL was generally inferior to that of E2, especially when the size of positive training examples was between 20% and 40% of the LingSpam corpus. E2 remained very steady in precision rate and was superior to its benchmark techniques over the range of sizes of positive training examples examined, as Figure 2(b) shows. PNB also achieved stable precision rates. As with E2, the precision rate attained by PEBL appeared to be largely insensitive to the size of positive training examples. However, with any size of positive training examples investigated, the precision rate of PEBL was lower than that of E2. Finally, the recall of PEBL was generally inferior to that of E2 and PNB when the size of positive training examples was between 20% and 40% of the corpus, as Figure 2(c) shows

For the PU1 corpus, as Figure 2(d) depicts, the accuracy of E2 was noticeably better than that of PNB over the range of sizes of positive training examples examined. As with the LingSpam corpus, the accuracy of PEBL was generally inferior to that of E2 over the range of sizes of positive training examples. With respect to precision (as Figure 2(e) shows), E2 was superior to its benchmark techniques when the size of positive training examples was lower than 25% and was comparable to its benchmark techniques when the size of positive training examples was higher than 30%. Finally, the trend of recall rate of each technique (as Figure 2(f) shows) was similar to that of accuracy rate. The recall rates of both PNB and PEBL were inferior to that of E2 over the range of sizes of positive training examples examined.

In summary, the evaluation results on both spam-filtering corpora showed that E2 achieved better performance than PNB (measured by accuracy, precision rate, and recall rate) across most of sizes of positive training examples investigated. Furthermore, E2 outperformed PEBL in precision rate and recall rate. Such evaluation results, again, suggested the utility of the ensemble strategy employed by the proposed E2 technique.

5. Conclusion and Future Research Directions

In many spam filtering scenarios, obtaining legitimate emails for training purpose is more difficult than collecting spam and unclassified emails. Hence, it would be more appropriate to construct a classification model for spam filtering from positive (i.e., spam emails) and unlabeled instances only. Several single-class learning techniques that include PNB and PEBL have been proposed in the literature. However, they incur fundamental limitations when applying to spam filtering. In this study, we propose and develop an ensemble approach, referred to as E2, to address the PNB's sensitivity to the estimate $\hat{Pr}(C_p)$ and the PEBL's susceptibility to the accuracy of the initial set of "strong negative" examples. Specifically, we follow the two-stage framework of PEBL and extend each stage with an ensemble strategy. Our empirical evaluation results on two spam-filtering corpora suggest that the proposed E2 technique exhibits more stable and reliable performance than its benchmark techniques (i.e., PNB and PEBL).

Some additional research works related to this study might include the followings. First, even though the proposed E2 technique is less insensitive to $\hat{Pr}(C_p)$ (i.e., the estimate of the prior probability of the spam class), inaccurate estimates of $\hat{Pr}(C_p)$, to some extent, have negative effects on the precision of E2. Because the percentage of spam emails is highly variable over time, development of a dynamic adjustment mechanism for $\hat{Pr}(C_p)$ is essential to the proposed E2 technique. Second, the topic of spam emails is also changing over time. Thus, the adaptive learning ability of the single-class learning technique would be considered desirable in the spam filtering context. Third, our empirical evaluation results show that the performance of all techniques investigated is unsatisfactory when the size of positive training examples is very small. In the future research work, it would be important to enhance the effectiveness of the proposed E2 technique to address the learning from a small size of positive training examples. Fourth, in this study, we use only SVM, Naive Bayes, and C4.5 as the base classifiers in the Convergence via Ensemble stage of E2. Inclusion and empirical evaluation of different classifiers would represent an interesting future research direction. Finally, in this study, we only consider monolingual spam filtering problem. In the future, the proposed E2 technique should be extended to deal with multilingual spam filtering on the basis of positive and unlabeled training emails.

Acknowledgements

This work was supported by National Science Council of the Republic of China under the grant NSC 94-2416-H-110-018.

References

Agrawal, R., Bayardo, R., and Srikant, R., "Athena: Mining-Based Interactive Management of Text Databases," *Proceedings of the 7th International Conference on Extending Databases Technology* (*EDBT00*), 2000, pp.365-379.

Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., and Spyropoulos, C. D., "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages," *ACM Special Interest Group on Information Retrieval*, 2000a, pp.160-167.

Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G., and Spyropoulos, C. D., "An Evaluation of Naive Bayesian Anti-Spam Filtering," *Proceedings of Workshop on Machine Learning in the New Information Age*, Barcelona, Spain, 2000b.

Bauer, E. and Kohavi, R., "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," *Machine Learning*, (36), 1999, pp.105-139.

Boykin, P. O. and Roychowdhury, V. P., "Leveraging Social Networks to Fight Spam," *IEEE Computer*, (38: 4), April 2005, pp.61-68.

Breiman, L., "Stacked Regressions," *Machine Learning*, (24:1), 1996, pp.49-64.

Carreras, X. and Márquez, L., "Boosting Trees for Anti-Spam Email Filtering," *Proceedings of 4th International Conference on Recent Advances in Natural Language Processing (RANLP)*, 2001, pp.58-64.

Comité, F. De, Denis, F., Gilleron, R. and Letouzey, F., "Positive and Unlabeled Examples Help Learning," *Lecture Notes in Artificial Intelligence*, Vol. 1720, 1999, pp.219-230.

Cunningham, P., Nowlan, N., Delany, S. J. and Haahr, M., "A Case-Based Approach to Spam Filtering that Can Track Concept Drift," *Proceedings of International Conference on Case-Based Reasoning*, June 2003.

Denis, F., Gilleron, R., and Tommasi, M., "Text Classification from Positive and Unlabeled Examples," *Proceedings 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2002, pp.1927–1934.

Dietterich, T.G., "Ensemble Methods in Machine Learning," *Proceedings of the First International Workshop on Multiple Classifier Systems*, 2000, pp.1-15.

Dong, Y. S. and Han, K. S., "A Comparison of Several Ensemble Methods for Text Categorization," *Proceedings of IEEE International Conference on Services Computing*, 2004, pp.419-422.

Drucker, H., Wu, D., and Vapnik, V., "Support Vector Machines for Spam Categorization," *IEEE Transactions on Neural Networks*, (10:5), 1999, pp.1048-1054.

Fallows, D., "Spam: How It Is Hurting E-Mail and Degrading Life on the Internet," Technical Report, Pew Internet and American Life Project, October 2003, Available: http://www.pewinternet.org/reports/toc.asp?Report=102.

Hansen, L. and Salamon, P., "Neural Network Ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (12:10), 1990, pp.993-1001.

Kolcz, A. and Alspector, J., "SVM-based Filtering of Email Spam with Content-specific Misclassification Costs," *Proceedings of International Conference on Data Mining Workshop on Text Mining*, 2001

Letouzey, F., Denis, F. and Gilleron, R., "Learning from Positive and Unlabeled examples," *Proceedings of the 11th International Conference on Algorithmic Learning Theory*, 2000, pp.71-85.

Opitz, D. and Maclin, R., "Popular Ensemble Methods: An Empirical Study," *Journal of Artificial Intelligence Research*, (11), 1999, pp.169-198.

Pantel, P. and Lin, D., "SpamCop: A Spam Classification Organization Program," *Proceedings of 1998 Workshop on Learning for Text Categorization*, 1998.

Porter, M.F., "An Algorithm for Suffix Stripping," *Program*, (14:3), 1980, pp.130-137.

Sahami, M., Dumais, S., Heckerman, D. and Horvitz, E., "A Bayesian Approach to Filtering Junk E-Mail," *Proceedings of 1998 Workshop on Learning for Text Categorization*, 1998.

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C., and Stamatopoulos, P., "A Memory-Based Approach to Anti-Spam Filtering," *Information Retrieval,* (6:1), 2003, pp.49-73.

Schneider, K., "A Comparison of Event Models for Naïve Bayes Anti-Spam E-mail Filtering," *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 2003.

Sebastiani, F., "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, (34:1), March 2002, pp.1-47.

Taylor, C., "Spam's Big Bang," *Time*, June 16, 2003, p.51.

Weiss, S. M., Apte, C., Damerau, F. J., Johnson, D. E., Oles, F. J., Goetz, T., and Hampp, T., "Maximizing Text-Mining Performance," *IEEE Intelligence Systems*, (14:4), 1999, pp.63-69.

Whitworth, B. and Whitworth, E., "Spam and the Social-Technical Gap," *IEEE Computer*, (37:10), October 2004, pp.38-45.

Yu, H., Han, J. and Chang, K. C. C., "PEBL: Web Page Classification without Negative Examples," *IEEE Transaction on Knowledge and Data Engineering*, (16:1), 2004, pp.70-81.

Zhang, L. and Yao, T., "Filtering Junk Mail with A Maximum Entropy Model," *Proceedings of 20th International Conference on Computer Processing of Oriental Languages (ICCPOL 03)*, 2003, pp.446-453.

Zhang, L. Zhu, J., and Yao, T., "An Evaluation of Statistical Spam Filtering Techniques," *ACM Transactions on Asian Language Information Processing*, (3:4), December 2004, pp.243-269.

Zorkadis, V., Panayotou, M., and Karras, D. A., "Improved Spam e-Mail Filtering Based on Committee Machines and Information Theoretic Feature Extraction," *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, 2005, pp.179-184.