J Intell Inf Syst
DOI 10.1007/s10844-013-0254-7

Cost-sensitive three-way email spam filtering

Bing Zhou · Yiyu Yao · Jigang Luo

Received: 25 July 2012 / Revised: 24 May 2013 / Accepted: 29 May 2013

Abstract Email spam filtering is typically treated as a binary classification problem that can be solved by machine learning algorithms. We argue that a three-way decision approach provides a more meaningful way to users for precautionary handling their incoming emails. Three email folders instead of two are produced in a three-way spam filtering system, a suspected folder is added to allow users make further examinations of suspicious emails, thereby reducing the chances of misclassification. Different from existing ternary email spam filtering systems, we focus on two issues that are less studied, that is, the computation of required thresholds to define the three email categories, and the interpretation of the cost-sensitive characteristics of spam filtering. Instead of supplying the thresholds based on intuitive understandings of the levels of tolerance for errors, we systematically calculate the thresholds based on decision-theoretic rough set model. A loss function is interpreted as the costs of making classification decisions. A decision is made for which the overall cost is minimum. Experimental results show that the new approach reduces the error rate of misclassifying a legitimate email to spam and demonstrates a better performance for the cost-sensitivity aspect.

Keywords Email spam filtering · Cost-sensitive learning · Ternary classification · Three-way decision · Naive Bayes classifier

B. Zhou (✉)
Department of Computer Science, Sam Houston State University,
Huntsville, TX 77341, USA
e-mail: zhou@shsu.edu

Y. Y. Yao · J. G. Luo
Department of Computer Science, University of Regina,
Regina, SK, Canada S4S 0A2

Y. Y. Yao
e-mail: yyao@cs.uregina.ca

J. G. Luo
e-mail: luo226@cs.uregina.ca

Published online: 09 June 2013

1 Introduction

Email spam filtering is a growing concern. Over the years, various anti-spam technologies and softwares have been developed. One popular approach is to treat spam filtering as a classification problem. Many classification algorithms from machine learning can be applied to automatically classify incoming emails into different categories based on the contents of emails (Cristianini and Shawe-Taylor 2000; Good 1965; Masand et al. 1992; Mitchell 1997; Pantel and Lin 1998; Sahami et al. 1998; Schapire and Singer 2000). Among these algorithms, the naive Bayes classifier has received much attention and served as a base for many open source projects and commercial products due to its simplicity, computational efficiency and good performance (Barracuda Spam Firewall 2013; Bogofilter 2013; GFI MailEssentials 2013; Rennie 1996; Robinson 2004; Yerazunis 2003). The naive Bayes classifier, along with many other classification algorithms, typically treat spam filtering as a binary classification problem, that is, the incoming email is either spam or non-spam. In reality, this simple treatment is too restrictive and could result in losing vital information by misclassifying a legitimate email to spam. For example, a user could miss an important job offer just because the email contains "congratul" (i.e., a common word in email spam filter word list) in its header. On the other hand, misclassifying a spam email to non-spam also brings unnecessary costs and waste of resources.

In a binary spam filtering system, two email folders are produced that contain legitimate and spam messages, respectively. However, it is often difficult to choose the best cut-off value (threshold) on probability for deciding spam and non-spam. A higher threshold may be in favor of high precision but low recall, a lower threshold may produce opposite results. In a three-way email spam filtering system, a pair of thresholds is used to produce three email folders, the accepted folder, the suspected folder, and the rejected folder. A user may view the accepted folder immediately, delays the processing of suspected folders, and deletes the rejected folder without viewing. This is a useful option if users are reviewing these emails under a time constraint. By adding the suspected folder, the misclassification errors from incorrect acceptance and incorrect rejection are reduced by introducing deferment errors. Thus, we can have a better precision and recall.

Ideas of ternary email spam filtering has been discussed in previous literatures. Robinson (2004) suggested to add a boundary region marked unsure to the classification results. Yih et al. (2007) suggested to call these messages that could not reasonably be considered either spam or non-spam as gray mail, and proposed four prototype methods for detecting them. Zhao and Zhang (2005) proposed a classification schema based on rough set theory to classify the incoming emails into three categories, spam, non-spam, and suspicious. Siersdorfer and Weikum (2005) introduced a framework of using restrictive methods and ensemble-based meta methods for junk elimination. In their approach, classifiers for a given topic make a ternary decision on a newly seen document: they can accept the document for the topic, reject it for the topic, or abstain if there is neither sufficiently strong evidence for acceptance nor sufficiently strong evidence for rejection. Zhou et al. (2010) proposed a three-way decision approach based on Bayesian decision theory. Ternary classifications were also used in some anti-spam applications, such as SpamBayes (http://spambayes.sourceforge.net/), Bogofilter (2013), and SpamAssassin (http://spamassassin.apache.org/). The main advantage of the ternary email spam

filtering is that it allows the possibility of indecision, i.e., do not make a yes or no decision in close cases. This is a useful option if the cost of being indecisive is not too high. The undecided cases must be re-examined by collecting additional information, thereby classifying emails with fewer errors.

There are two key issues in ternary email spam filtering. The first issue is the estimation of the legitimacy of an email. For content-based statistical filtering, the selected model generally produces a real-valued number that indicates the legitimacy of an email. For instance, the naive Bayes classifier uses posterior probability to represent the possibility of an email being legitimate given its feature descriptions. For rule-based approach, this estimation is done by evaluating strength of the matching rules through some quantitative measures, such as accuracy and coverage. The second issue is the interpretation and computation of required thresholds to define the three email categories. The existing ternary email spam filtering methods choose the thresholds fairly arbitrarily based on an intuitive understanding of the levels of tolerance for errors. For example, SpamBayes uses 0.9 to determine between spam and unsure, while 0.2 separates unsure and legitimate class. Bogofilter uses 0.99–0.45 for the unsure range.

The existing ternary email spam filtering methods focus on the first issue, that is, working towards functions that give better estimations of legitimacy of emails. After testing on the statistical-based and rule-based filtering, Graham (2002) pointed out that although the rule-based approach is easy to begin with, but it gets very hard to catch the last few percent of spam, the statistical-based filtering is a better way to stop spam. On the other hand, estimations of required thresholds have not received much attention. Little analysis has been done to determine the optimal thresholds. There is a need for inferring these thresholds from a theoretical and practical basis.

Further more, cost-sensitive classification has received much attention in recent years. In the traditional classification task, minimizing misclassification rate is used as the guideline of designing a good classifier. The misclassification rate is also called the error rate or 0/1 loss function, which assigns no loss to a correct classification, and assigns a unit loss to any error. Thus, all errors are equally costly. In real-world applications, each error has an associated cost. For instance, the cost of false positive errors (i.e., giving treatment to a patient who does not have cancer) is different from the cost of false negative errors (i.e., failing to treat a patient who has cancer). Therefore, it is important to build a cost-sensitive classifier to incorporate different types of costs into the classification process. Cost-sensitive learning is one of the challenging problems in the current stage of data mining and machine learning research (Zhou and Liu 2012). Email spam filtering is a typical cost-sensitive task (Elkan 2001; Zhou and Liu 2006, 2010). Misclassifying a legitimate email to spam is usually considered more costly than misclassifying a spam to legitimate. Such characteristics have not been explicitly reflected and made clear in the existing ternary email spam filtering methods.

In this paper, we introduce a cost-sensitive three-way decision approach to address the above issues in email spam filtering. Since the estimations of the legitimacy of an email has been extensively discussed in other papers (Robinson 2004; Siersdorfer and Weikum 2005; Yih et al. 2007; Zhao and Zhang 2005), we will concentrate on the other issues, namely, estimations of the required thresholds and characterizations of the cost-sensitive features. More specifically, we adopt the systematic method from the decision-theoretic rough set (DTRS) models to calculate a pair of thresholds based on the well established Bayesian decision theory, with the aid of more

practically operable notions such as cost, risk, benefit etc. (Yao et al. 1990). A loss function is defined to state how costly each decision is, and a final decision is to choose the one for which the overall cost is minimum. Experimental results on several benchmark datasets show that the new approach consistently performs better than the binary naive Bayes classifier, and outperforms the other two existing ternary email spam filtering methods in cost-sensitive settings.

2 Workflow of a spam filtering system

The workflow of an email spam filtering system is based on the bag of words model. Several pre-processing steps are required before information can be used by a filter. The main steps involved in a spam filter are illustrated in Fig. 1.

At the first step, the whole text of every email need to be scanned to obtain words, phrases or meta-data as tokens. The set of tokens is then represented by a format

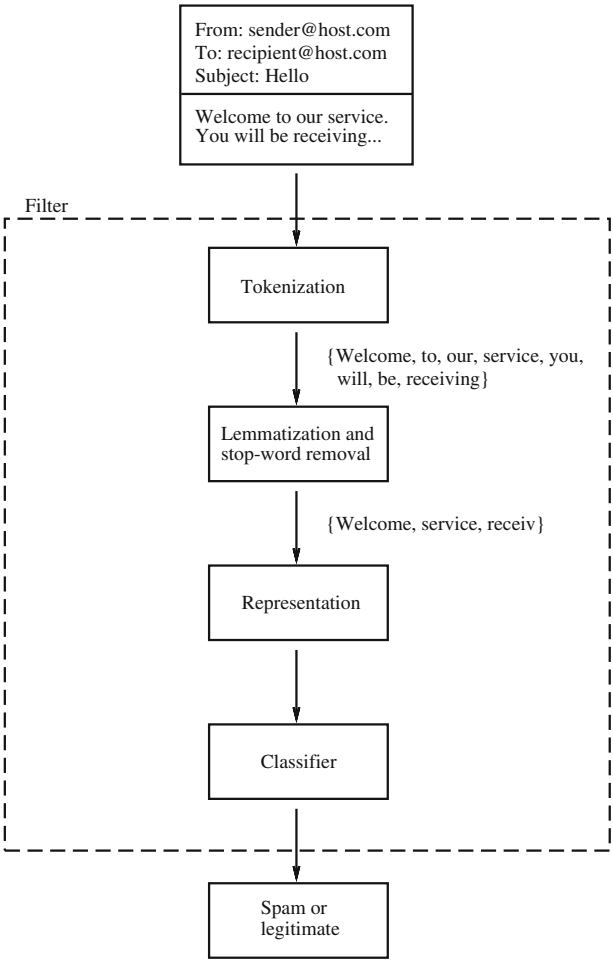


Fig. 1 An illustration of some of the main steps involved in a spam filter

(e.g., real values) required by the machine learning algorithm used. These values indicate a number of things such as the presence of a token or the frequency with which a given token occurs.

At the second step, words common to both spam and non-spam classes will be removed. For example, words, such as "to," "will" and "be," provide very little information as to the class of the email message. At the same time, a step called lemmatization (stemming) reduces words to their root forms. For example, "receiving" and "received" will be treated as "receiv" rather than two distinct words.

At the third step, feature selection is often performed to reduce the size of the set of selected tokens to ensure the performance of algorithms. There are two commonly used feature selection methods in email spam filtering. Document Frequency (DF) is the number of emails in which a token occurs. Information Gain (IG) measures the number of bits of information obtained for category prediction by knowing the presence or absence of a token in an email. It is assumed that rare tokens are non-informative for category prediction. Tokens whose DF or IG are less than some predetermined thresholds will be removed.

At the last step, after a set of emails have been scanned, they can be represented in a data table (i.e., the training set). We have a set of tokens as attributes, one of these attributes is class (e.g., legitimate or spam). Each email is a record of the collection. The attribute values are frequencies of tokens. A learning algorithm can be performed on the training set, so we can find a model (e.g., a decision tree or a set of rules) to measure the hamminess or spamminess of each email. Finally when a new email comes in, we can apply the model and predict its class.

The last step is a typical classification process in data mining and it is the part that we want to improve on.

3 Two email classification models

In this section, we review and identify issues in two email classification models within a unified formulation. Understanding these issues enables us to show clearly the contributions of this paper.

3.1 Basic formulations

Suppose each incoming email is represented by a feature vector $\mathbf{x} = (x_1, x_2, ..., x_n)$, where $x_1, x_2, ..., x_n$ represent the occurrence of the selected features in a message. Let C denote the legitimate class, and C^c denote the spam class. Based on the description of email, we can use a discriminant function $f(\mathbf{x})$ for classification. For example, in naive Bayes classifier, $f(\mathbf{x})$ is the posterior probability or its monotonic transformations (e.g., the posterior odds). In SVM method, $f(\mathbf{x})$ is the distance between the given email and the decision hyperplane. There are typically two ways to use a discriminant function for classification. One is to use the discriminant function to rank emails and let a user read through the ranked list. The other is to classify emails into several categories based on some thresholds on $f(\mathbf{x})$.

In the classic binary email spam filtering, only one threshold $\gamma \in [0, 1]$ is used to compare with normalized $f(\mathbf{x})$ in order to determine whether or not to reject an email. Two classification regions, called the positive and negative regions, are

produced for a given set of emails C (the same symbol is used to denote the corresponding class label), that is,

$$\begin{aligned} \mathrm{POS}_{(\gamma)}(C) &= \{\mathbf{x} \mid f(\mathbf{x}) \geq \gamma\}, \\ \mathrm{NEG}_{(\gamma)}(C) &= \{\mathbf{x} \mid f(\mathbf{x}) < \gamma\}. \end{aligned} \tag{1}$$

The positive region $\mathrm{POS}_{(\gamma)}(C)$ contains emails that are accepted as legitimate, and the negative region $\mathrm{NEG}_{(\gamma)}(C)$ contains emails that are rejected as spam.

For ternary email spam filtering, although the real class label of an email is only binary (i.e., C or C^c), we make a three-way decision for each incoming email. A pair of thresholds (α, β) with $0 \leq \beta < \alpha \leq 1$ is used to distinguish different value ranges of $f(\mathbf{x})$. The first threshold α determines the probability necessary for a re-examination, and the second threshold β determines the probability necessary to reject an email. The pair of thresholds produces three classification regions, called the positive, boundary, and negative regions as follows:

$$\begin{aligned} \mathrm{POS}_{(\alpha,\beta)}(C) &= \{\mathbf{x} \mid f(\mathbf{x}) \geq \alpha\}, \\ \mathrm{BND}_{(\alpha,\beta)}(C) &= \{\mathbf{x} \mid \beta < f(\mathbf{x}) < \alpha\}, \\ \mathrm{NEG}_{(\alpha,\beta)}(C) &= \{\mathbf{x} \mid f(\mathbf{x}) \leq \beta\}, \end{aligned} \tag{2}$$

We accept an email x to be a legitimate email if $f(\mathbf{x})$ is greater than or equals to α. We reject x to be a legitimate email if $f(\mathbf{x})$ is less than or equals to β. We neither accept nor reject x if $f(\mathbf{x})$ is between α and β, instead, we make a decision of deferment. The differences between binary and ternary spam filtering are demonstrated in Fig. 2.

3.2 Binary naive Bayesian spam filtering

The naive Bayesian spam filtering is a probabilistic classification technique of email filtering (Pantel and Lin 1998; Sahami et al. 1998). It is based on Bayes' theorem with naive (strong) independence assumptions (Good 1965; Mitchell 1997). The main idea is to transfer a difficult-to-estimate probability $Pr(C|\mathbf{x})$ into an easy-to-estimate probability $Pr(\mathbf{x}|C)$.

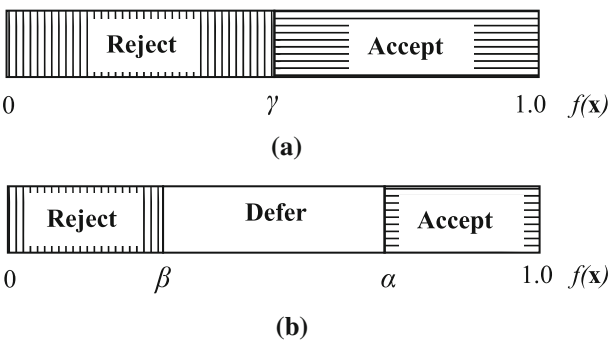


Fig. 2 **a** Binary spam filtering with single threshold. **b** Ternary spam filtering with a pair of thresholds

Based on Bayes' theorem and the theorem of total probability, the posterior probability of an email is in C given $\mathbf{x}$ is:

$$Pr(C|\mathbf{x}) = \frac{Pr(C)\,Pr(\mathbf{x}|C)}{Pr(\mathbf{x})}, \tag{3}$$

where $Pr(\mathbf{x}) = Pr(\mathbf{x}|C)\,Pr(C) + Pr(\mathbf{x}|C^c)\,Pr(C^c)$, $Pr(C)$ is the prior probability of C, and $Pr(\mathbf{x}|C)$ is the likelihood of C given $\mathbf{x}$.

The likelihood $Pr(\mathbf{x}|C)$ is a joint conditional probability $Pr(x_1, x_2, ..., x_n|C)$. In practice, it is difficult to analyze the interactions between the components of $\mathbf{x}$, especially when the number n is large. In order to solve this problem, an independence assumption is used in naive Bayes classifier which assumes that each feature x_i is conditionally independent of every other features, given the class C, this yields,

$$\begin{aligned} Pr(\mathbf{x}|C) &= Pr(x_1, x_2, ..., x_n|C) \\ &= \prod_{i=1}^{n} Pr(x_i|C), \end{aligned} \tag{4}$$

where $Pr(x_i|C)$ can be easily estimated as relative frequencies from data. Thus (3) can be rewritten as:

$$Pr(C|\mathbf{x}) = \frac{Pr(C)\prod_{i=1}^{n} Pr(x_i|C)}{Pr(\mathbf{x})}. \tag{5}$$

Similarly, the posterior probability $Pr(C^c|\mathbf{x})$ can be written as:

$$Pr(C^c|\mathbf{x}) = \frac{Pr(C^c)\prod_{i=1}^{n} Pr(x_i|C^c)}{Pr(\mathbf{x})}. \tag{6}$$

The probability $Pr(\mathbf{x})$ can be eliminated by taking the ratio of $Pr(C|\mathbf{x})$ and $Pr(C^c|\mathbf{x})$ as follows,

$$\frac{Pr(C|\mathbf{x})}{Pr(C^c|\mathbf{x})} = \prod_{i=1}^{n} \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \frac{Pr(C)}{Pr(C^c)}. \tag{7}$$

An incoming email will be classified as legitimate if $\frac{Pr(C|\mathbf{x})}{Pr(C^c|\mathbf{x})}$ (i.e., the posterior odds) exceeds a threshold, otherwise it is spam.

3.3 Related work on ternary email spam filtering

3.3.1 Robinson's approach

Given an email x described by its feature vector $\mathbf{x} = (x_1, x_2, ..., x_n)$. Robinson (2004) calculates the spamminess of each selected word based on its occurrence x_i, and then combines these individual probabilities into an overall probability for x. The per-word probability $Pr(C^c|x_i)$ is calculated as follows,

$$Pr(C^c|x_i) = \frac{(s \times p) + \left(m_{x_i} \times \frac{Pr(x_i|C^c)}{Pr(x_i|C^c)+Pr(x_i|C)}\right)}{s + m_{x_i}}, \tag{8}$$

where s is a strength value based on background information, p is an assumed probability, based on our general background information, that a word will first appear in a spam, and m_{x_i} is the number of emails received that contain x_i. In practice, the values for s and p are found through testing to optimize performance. Reasonable starting points are 1 and 0.5 for s and p, respectively.

The per-word probabilities are combined into an overall probability based on Fisher's inverse chi-square procedure (Triola 2005). The combined probability H is calculated as follows,

$$H = C^{-1}\left(-2\ln\prod_{i=1}^{n} Pr(C^c|x_i), 2n\right), \tag{9}$$

where $C^{-1}()$ is the inverse chi-square function, used to derive a probability from a chi-square-distributed random variable. H indicates the legitimacy of an given email. The combined probability S that represents the spamminess of the email is calculated by:

$$S = C^{-1}\left(-2\ln\prod_{i=1}^{n}(1 - Pr(C^c|x_i)), 2n\right). \tag{10}$$

The final probability is:

$$I = \frac{1 + H - S}{2}. \tag{11}$$

A given email is classified as spam if the value of I is near 1, is classified as legitimate if I is near 0, and is classified as uncertain when I is near 0.5.

Robinson's approach does not need naive independence assumption when combining the individual probabilities, but it is based on an assumption that a randomly chosen e-mail containing x_i would be spam in a world where half the e-mails were spam and half were ham. Whereas in reality, legitimate and spam emails are usually not equally distributed. In addition, the selection of the thresholds 1 and 0.5 is based on intuition with little analysis.

3.3.2 Rough set approach

Zhao and Zhang (2005) introduced a rough set approach for automatically learning rules to classify emails into three categories: spam, non-spam and suspicious. A set of decision rules is induced by a genetic algorithm intergraded in the rough set tool kit, Rosetta (http://rosetta.lcb.uu.se). The strength of each rule is evaluated by its accuracy. For an email $\mathbf{x}$, suppose a rule r that matches $\mathbf{x}$ is given in the form $\mathbf{x} \rightarrow C$, where the left-hand side is the conjunction of features that describes $\mathbf{x}$, and the right-hand side is the non-spam class label C. Let $RUL(\mathbf{x})$ denote the set of these types rules with non-spam as consequent that matches $\mathbf{x}$. The certainty of $\mathbf{x}$ being in the non-spam class is measured as follow,

$$Certainty_{\mathbf{x}} = \frac{\sum_{r\in RUL(\mathbf{x})} accuracy(r)}{\mid RUL(\mathbf{x}) \mid}, \tag{12}$$

where $|\cdot|$ denote the cardinality of a set, and the accuracy of r is calculated by $accuracy(r) = \frac{|C\cap\mathbf{x}|}{|\mathbf{x}|}$. A pair of thresholds α and β is used to compare with $Certainty_{\mathbf{x}}$

to define the three email categories. An email $\mathbf{x}$ is classified into the non-spam category if $Certainty_{\mathbf{x}} \geq \alpha$, is classified into the suspicious category if $\beta \leq Certainty_{\mathbf{x}} < \alpha$. Otherwise, it will be classified into the spam category.

There are a few problems with the rough set approach to spam filtering. First, when more than one rule matches a given email $\mathbf{x}$, features on the left-hand side of rules in $RUL(\mathbf{x})$ may have overlaps. Simply adding up the accuracy of each rule may cause repeated consideration of some features which will lead to biased classification results. Second, accuracy is the only measure used to evaluate the strength of a rule. It may not be able to provide reliable indications for some data sets. For example, suppose we have a data set that contains important evidences of emails being spam, but it may not contain all evidences of emails being legitimate. Even we have equally distributed positive and negative examples, it does not mean that the probability of an email being spam is 50 %. In other words, using evidence $|\mathbf{x}|$ as a denominator to measure the accuracy of a rule may mislead the classification results. Other forms of rule evaluations should also be considered. Third, the two required thresholds are arbitrarily defined with one simple constrain, that is, $\alpha \in (\frac{1}{2}, 1]$ and $\beta = 1 - \alpha$. Last, the CPU time to learn accurate rule sets from data is higher than the statistical methods. Considering that spam filtering tends to be computationally demanding, especially when the volume of emails is high, the applicability of rule-based learning algorithms to practical spam filtering systems is problematic (Cohen 1996).

3.3.3 The ensemble methods

Siersdorfer and Weikum (2005) proposed a restrictive meta method for eliminating junk documents, in which junk documents are defined as the class of documents that does not appear in the training set, but appears in the testing set. In their approach, restrictions are first made to a set of binary classifiers, these classifiers are then ensembled to make a ternary decision on a newly seen document: they can accept the document for the topic, reject it for the topic, or abstain if there is neither sufficiently evidence for acceptance nor for rejection.

The method they used to make binary classifiers restrictive is in fact to have a three-way classification, which is similar to the ideas of ternary email spam filtering. A pair of thresholds is used to compare with the return value of each classifier. More specifically, we are given a set $V = \{v_1, v_2, ..., v_k\}$ of k binary classifiers with results $R(v_j, d)$ in $\{+1, -1, 0\}$ for a document d. The value of $R(v_j, d)$ is $+1$ if the return value of v_j is above the first threshold (i.e., d is accepted for the given topic by v_j), -1 if the return value of v_j is below the second threshold (i.e., d is rejected), and 0 if the return value of v_j lies between the two thresholds (i.e., d is abstained). These results are combined into a meta result as follows:

$$Meta(d) = \begin{cases} +1 & \text{if } \sum_{j=1}^{k} R(v_j, d) \cdot weight(v_j) > \alpha \\ -1 & \text{if } \sum_{j=1}^{k} R(v_j, d) \cdot weight(v_j) < \beta \\ 0 & \text{otherwise} \end{cases}$$

where α and β are two thresholds with $\alpha > \beta$, and $weight(v_j)$ is a weight factor for each v_j. The restrictive and tunable behavior is achieved by the choice of the thresholds: we dismiss the documents where the meta result combination lies between α and β.

Siersdorfer's approach can be applied to ternary email spam filtering with little justifications. The main difference is that in ternary email spam filtering, there are only positive and negative examples even in the testing set, but we make a ternary classification on a binary scale. Thus, Siersdorfer's approach returns a 3×3 contingency table that contains the classification results, whereas the ternary classification methods returns a 2×3 contingency table, which represents different types of misclassification errors and costs. Similar ideas of the ensemble method can be found in Yih et al. (2007), where multiple email filters are performed using different disjoint subsets of the training data. An email is classified based upon the level of disagreements between these filters. However, both of these approaches did not provide evidence to show that the ensembled method is better than applying a single filter. Siersdorfer's experimental results found that for some datasets, the meta classifier outperformed the single classifier, but for some other datesets, the single classifier performed better.

3.4 Discussions

In general, the ternary email spam filtering adds a boundary region to the classification results. Those emails that can not be easily classified are moved from the positive and negative regions into the boundary region for further examination. With the deferment option, we aim to reduce the chances of a legitimate email being ignored and a spam being accepted due to misclassifications. In other words, the ternary email spam filtering reduce the acceptance or rejection errors by introducing deferment errors.

The existing ternary email spam filtering methods focus on deriving efficient functions to represent the legitimacy of email. The content-based statistical filtering, the rule-based filtering, and the combined ensemble methods are all employed in the ternary email spam filtering methods. On the other hand, the derivation of required thresholds has not received much attention. They are either provided based on intuitions (Robinson 2004; Zhao and Zhang 2005) or used as two indicators without much explanations (Siersdorfer and Weikum 2005). We argue that there is a need for inferring these thresholds from a theoretical and practical basis. Cost-sensitive evaluations have been used for email spam filtering by considering the different costs of misclassifications (Androutsopoulos et al. 2000). However, the existing ternary email spam filtering methods are still based on the standard non-cost-sensitive learning methods. It is critical to find a solution to reflect the cost-sensitive characteristics in ternary email spam filtering models.

4 A cost-sensitive three-way decision approach

A cost-sensitive three-way decision approach is introduced in this section. We adopt a monotonic transformation of the posterior probability and concentrate on the derivation of the required thresholds based on the systematic method provided in the DTRS models.

4.1 Overview of Bayesian decision theory

Bayesian decision theory is a fundamental statistical approach that makes decisions under uncertainty based on probabilities and costs associated with decisions. Following the discussions given in the book by Duda and Hart (1973), the basic ideas of the theory are reviewed.

Let $\Omega = \{w_1, \ldots, w_s\}$ be a finite set of s states and let $\mathcal{A} = \{a_1, \ldots, a_t\}$ be a finite set of t possible actions. Let $\lambda(a_i|w_j)$ denote the loss, or cost, for taking action a_i when the state is w_j. Let $Pr(w_j|\mathbf{x})$ be the posterior probability of an email being in state w_j given that the email is described by $\mathbf{x}$. For an email with description $\mathbf{x}$, suppose action a_i is taken. Since $Pr(w_j|\mathbf{x})$ is the probability that the true state is w_j given $\mathbf{x}$, the expected loss associated with taking action a_i is given by:

$$R(a_i|\mathbf{x}) = \sum_{j=1}^{s} \lambda(a_i|w_j)\, Pr(w_j|\mathbf{x}). \tag{13}$$

The quantity $R(a_i|\mathbf{x})$ is also called the conditional risk.

Given a description $\mathbf{x}$, a decision rule is a function $\tau(\mathbf{x})$ that specifies which action to take. That is, for every $\mathbf{x}$, $\tau(\mathbf{x})$ takes one of the actions, $a_1, \ldots, a_t$. The overall risk $\mathbf{R}$ is the expected loss associated with a given decision rule. Since $R(\tau(\mathbf{x})|\mathbf{x})$ is the conditional risk associated with action $\tau(\mathbf{x})$, the overall risk is defined by:

$$\mathbf{R} = \sum_{\mathbf{x}} R(\tau(\mathbf{x})|\mathbf{x})\, Pr(\mathbf{x}), \tag{14}$$

where the summation is over the set of all possible descriptions of emails. If $\tau(\mathbf{x})$ is chosen so that $R(\tau(\mathbf{x})|\mathbf{x})$ is as small as possible for every $\mathbf{x}$, the overall risk $\mathbf{R}$ is minimized. Thus, the optimal Bayesian decision procedure can be formally stated as follows. For every $\mathbf{x}$, compute the conditional risk $R(a_i|\mathbf{x})$ for $i = 1, \ldots, t$ defined by (13) and select the action for which the conditional risk is minimum. If more than one action minimizes $R(a_i|\mathbf{x})$, a tie-breaking criterion can be used.

4.2 Computing thresholds based on DTRS

With respect to a set C of emails, we can form a set of two states $\Omega = \{C, C^c\}$ indicating that an email is in C (i.e., legitimate) or not in C (i.e., spam). To derive the three classification regions, the set of actions is given by $\mathcal{A} = \{a_P, a_B, a_N\}$, where a_P, a_B, and a_N represent the three actions in classifying an email x, namely, deciding $x \in \mathrm{POS}(C)$, deciding $x \in \mathrm{BND}(C)$, and deciding $x \in \mathrm{NEG}(C)$, respectively. The loss function is given by a 3×2 matrix:

	C (P): positive	C^c (N): negative
a_P: accept	$\lambda_{PP} = \lambda(a_P\|C)$	$\lambda_{PN} = \lambda(a_P\|C^c)$
a_B: defer	$\lambda_{BP} = \lambda(a_B\|C)$	$\lambda_{BN} = \lambda(a_B\|C^c)$
a_N: reject	$\lambda_{NP} = \lambda(a_N\|C)$	$\lambda_{NN} = \lambda(a_N\|C^c)$

In the matrix, λ_{PP}, λ_{BP} and λ_{NP} denote the losses incurred for taking actions a_P, a_B and a_N, respectively, when an email belongs to C, and λ_{PN}, λ_{BN} and λ_{NN} denote

the losses incurred for taking these actions when the email does not belong to C. In particular, λ_{NP} is the loss incurred for mistakenly rejecting a legitimate email, and λ_{PN} is the loss incurred for mistakenly accepting a spam email.

The expected losses associated with taking different actions for emails with description $\mathbf{x}$ can be expressed as:

$$\begin{aligned} R(a_P|\mathbf{x}) &= \lambda_{PP} Pr(C|\mathbf{x}) + \lambda_{PN} Pr(C^c|\mathbf{x}), \\ R(a_B|\mathbf{x}) &= \lambda_{BP} Pr(C|\mathbf{x}) + \lambda_{BN} Pr(C^c|\mathbf{x}), \\ R(a_N|\mathbf{x}) &= \lambda_{NP} Pr(C|\mathbf{x}) + \lambda_{NN} Pr(C^c|\mathbf{x}). \end{aligned} \tag{15}$$

The Bayesian decision procedure suggests the following minimum-risk decision rules:

(P) If $R(a_P|\mathbf{x}) \leq R(a_B|\mathbf{x})$ and $R(a_P|\mathbf{x}) \leq R(a_N|\mathbf{x})$, decide $x \in \text{POS}(C)$;

(B) If $R(a_B|\mathbf{x}) \leq R(a_P|\mathbf{x})$ and $R(a_B|\mathbf{x}) \leq R(a_N|\mathbf{x})$, decide $x \in \text{BND}(C)$;

(N) If $R(a_N|\mathbf{x}) \leq R(a_P|\mathbf{x})$ and $R(a_N|\mathbf{x}) \leq R(a_B|\mathbf{x})$, decide $x \in \text{NEG}(C)$.

Tie-breaking criteria should be added so that each email is put into only one region.

Since $Pr(C|\mathbf{x}) + Pr(C^c|\mathbf{x}) = 1$, we can simplify the rules based only on the probabilities $Pr(C|\mathbf{x})$ and the loss function λ. Consider a special kind of loss functions with:

$$\begin{aligned} \text{(c0).}\ \ &\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}, \\ &\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}. \end{aligned} \tag{16}$$

That is, the loss of classifying an email x being in C into the positive region $\text{POS}(C)$ is less than or equal to the loss of classifying x into the boundary region $\text{BND}(C)$, and both of these losses are strictly less than the loss of classifying x into the negative region $\text{NEG}(C)$. The reverse order of losses is used for classifying an email not in C. Under condition (c0), we can simplify decision rules (P)–(N) as follows. For the rule (P), the first condition can be expressed as:

$$\begin{aligned} & R(a_P|\mathbf{x}) \leq R(a_B|\mathbf{x}) \\ & \iff \lambda_{PP} Pr(C|\mathbf{x}) + \lambda_{PN} Pr(C^c|\mathbf{x}) \\ & \quad \leq \lambda_{BP} Pr(C|\mathbf{x}) + \lambda_{BN} Pr(C^c|\mathbf{x}) \\ & \iff \lambda_{PP} Pr(C|\mathbf{x}) + \lambda_{PN}(1 - Pr(C|\mathbf{x})) \\ & \quad \leq \lambda_{BP} Pr(C|\mathbf{x}) + \lambda_{BN}(1 - Pr(C|\mathbf{x})) \\ & \iff Pr(C|\mathbf{x}) \geq \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}. \end{aligned} \tag{17}$$

Similarly, the second condition of rule (P) can be expressed as:

$$\begin{aligned} & R(a_P|\mathbf{x}) \leq R(a_N|\mathbf{x}) \\ & \iff Pr(C|\mathbf{x}) \geq \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}. \end{aligned}$$

The first condition of rule (B) is the converse of the first condition of rule (P). It follows,

$$\begin{aligned} & R(a_B|\mathbf{x}) \leq R(a_P|\mathbf{x}) \\ & \iff Pr(C|\mathbf{x}) \leq \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}. \end{aligned}$$

For the second condition of rule (B), we have:

$$\begin{aligned} & R(a_B|\mathbf{x}) \leq R(a_N|\mathbf{x}) \\ & \iff Pr(C|\mathbf{x}) \geq \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}. \end{aligned}$$

The first condition of rule (N) is the converse of the second condition of rule (P) and the second condition of rule (N) is the converse of the second condition of rule (B). It follows,

$$\begin{aligned} & R(a_N|\mathbf{x}) \leq R(a_P|\mathbf{x}) \\ & \iff Pr(C|\mathbf{x}) \leq \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}, \\ & R(a_N|\mathbf{x}) \leq R(a_B|\mathbf{x}) \\ & \iff Pr(C|\mathbf{x}) \leq \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}. \end{aligned}$$

To obtain a compact form of the decision rules, we denote the three expressions in these conditions by the following three parameters:

$$\begin{aligned} \alpha &= \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}, \\ \beta &= \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}, \\ \gamma &= \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}. \end{aligned} \tag{18}$$

The decision rules (P)–(N) can be expressed concisely as:

(P) If $Pr(C|\mathbf{x}) \geq \alpha$ and $Pr(C|\mathbf{x}) \geq \gamma$, decide $x \in \text{POS}(C)$;

(B) If $Pr(C|\mathbf{x}) \leq \alpha$ and $Pr(C|\mathbf{x}) \geq \beta$, decide $x \in \text{BND}(C)$;

(N) If $Pr(C|\mathbf{x}) \leq \beta$ and $Pr(C|\mathbf{x}) \leq \gamma$, decide $x \in \text{NEG}(C)$.

Each rule is defined by two out of the three parameters.

The conditions of rule (B) suggest that $\alpha > \beta$ may be a reasonable constraint; it will ensure a well-defined boundary region. By setting $\alpha > \beta$, namely,

$$\frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})} > \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})},$$

we obtain the following condition on the loss function:

$$\text{(c1).} \quad \frac{\lambda_{NP} - \lambda_{BP}}{\lambda_{BN} - \lambda_{NN}} > \frac{\lambda_{BP} - \lambda_{PP}}{\lambda_{PN} - \lambda_{BN}}. \tag{19}$$

The condition (c1) implies that $1 \geq \alpha > \gamma > \beta \geq 0$. In this case, after tie-breaking, the following simplified rules are obtained:

(P1) If $Pr(C|\mathbf{x}) \geq \alpha$, decide $x \in \text{POS}(C)$;

(B1) If $\beta < Pr(C|\mathbf{x}) < \alpha$, decide $x \in \text{BND}(C)$;

(N1) If $Pr(C|\mathbf{x}) \leq \beta$, decide $x \in \text{NEG}(C)$.

The parameter γ is no longer needed.

From the rules (P1), (B1), and (N1), the (α, β)-probabilistic positive, negative and boundary regions are given, respectively, by:

$$\begin{aligned} \text{POS}_{(\alpha,\beta)}(C) &= \{x \mid Pr(C|\mathbf{x}) \geq \alpha\}, \\ \text{BND}_{(\alpha,\beta)}(C) &= \{x \mid \beta < Pr(C|\mathbf{x}) < \alpha\}, \\ \text{NEG}_{(\alpha,\beta)}(C) &= \{x \mid Pr(C|\mathbf{x}) \leq \beta\}. \end{aligned} \tag{20}$$

The threshold parameters can be systematically calculated from a loss function based on the Bayesian decision theory.

4.3 Estimating probabilities

The posterior probability $Pr(C|\mathbf{x})$ in (20) is not always directly derivable from data. We need to consider alternative ways to calculate their values. Recall that in naive Bayes classifier, the posterior probability is calculated based on the Bayes' theorem, which reduces the problem of estimating $Pr(C|\mathbf{x})$ into estimating the prior probability $Pr(C)$ and the likelihood $Pr(\mathbf{x}|C)$. There are many methods to estimate likelihood from data, which makes naive Bayes classifier practically useful.

The computation of the posterior probability $Pr(C|\mathbf{x})$ in the naive Bayes classifier involves a multiplication of many likelihoods, one for each feature, which can result in a floating point underflow. In practice, the multiplication of probabilities is often converted to an addition of logarithms of probabilities. In this paper, we use a monotonic transformation of the posterior probability to construct an equivalent classifier, that is, the logit transformation defined by $\text{logit}(Pr(\cdot)) = \log(O(\cdot)) =$

$\log \frac{Pr(\cdot)}{1-Pr(\cdot)}$. A threshold on $Pr(\cdot)$ can indeed be interpreted as another threshold on $\log \frac{Pr(\cdot)}{1-Pr(\cdot)}$. For the positive region, we have:

$$\begin{aligned} Pr(C|\mathbf{x}) \geq \alpha &\iff O(C|\mathbf{x}) \geq \frac{\alpha}{1-\alpha} \\ &\iff \frac{Pr(C|\mathbf{x})}{Pr(C^c|\mathbf{x})} \geq \frac{\alpha}{1-\alpha} \\ &\iff \frac{Pr(\mathbf{x}|C)}{Pr(\mathbf{x}|C^c)} \cdot \frac{Pr(C)}{Pr(C^c)} \geq \frac{\alpha}{1-\alpha} \\ &\iff \prod_{i=1}^{n} \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \geq \frac{Pr(C^c)}{Pr(C)} \cdot \frac{\alpha}{1-\alpha} \\ &\iff \sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \\ &\qquad \geq \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{PN}-\lambda_{BN}}{\lambda_{BP}-\lambda_{PP}}. \end{aligned}$$

Similar expression can be obtained for the negative region as:

$$\sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \leq \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{BN}-\lambda_{NN}}{\lambda_{NP}-\lambda_{BP}}.$$

Thus, we can derived a new pair of thresholds α' and β' as:

$$\begin{aligned} \alpha' &= \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{PN}-\lambda_{BN}}{\lambda_{BP}-\lambda_{PP}}, \\ \beta' &= \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{BN}-\lambda_{NN}}{\lambda_{NP}-\lambda_{BP}}, \end{aligned} \tag{21}$$

where $\log \frac{Pr(C^c)}{Pr(C)}$ is independent of the description of emails, we treat it as a constant.

We can then get the (α', β')-probabilistic positive, negative and boundary regions written as:

$$\begin{aligned} \mathrm{POS}_{(\alpha',\beta')}(C) &= \left\{ x \,\middle|\, \sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \geq \alpha' \right\}, \\ \mathrm{BND}_{(\alpha',\beta')}(C) &= \left\{ x \,\middle|\, \beta' < \sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} < \alpha' \right\}, \\ \mathrm{NEG}_{(\alpha',\beta')}(C) &= \left\{ x \,\middle|\, \sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \leq \beta' \right\}. \end{aligned} \tag{22}$$

All the factors in (22) are easily derivable from data.

4.4 An example

Based on notations in Section 4.2, there are two states (classes) regarding an incoming email: C (P) denoting a Legitimate email and C^c (N) denoting a Spam.

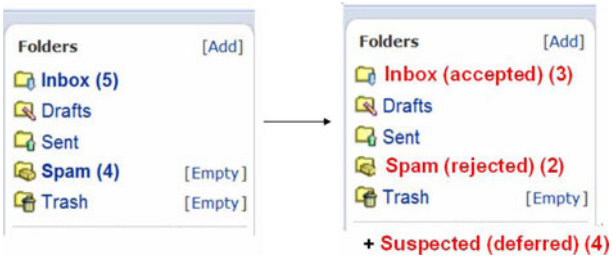


Fig. 3 From binary spam filtering to ternary spam filtering

There are three actions: a_P for accepting the email, a_B for making a deferred decision (i.e., neither accept nor reject the email due to insufficient information), and a_N for rejecting the email.

As shown in Fig. 3, three email folders are produced instead of two, the *Inbox* folder contains accepted emails, the *Spam* folder contains rejected emails, and the *Suspected* folder contains suspicious emails that need to be further examined. This is a useful option when users view their emails under a time constraint. They may view the *Inbox* folder immediately, delete the *Spam* folder without viewing, and delay the processing of *Suspected* folders. The emails in the suspected folder can be ranked based on their probability of being spam or legitimate to help the user make decisions later on.

A loss function is interpreted as the costs of taking the corresponding actions. Generally speaking, a higher cost occurs when misclassifying a legitimate email as a spam; it could result in losing vital information for a user. On the other hand, misclassifying a spam to be a legitimate email brings unnecessary costs of processing the spam. Costs also occur when a delayed decision is made. The costs depend very much on a particular user's subjective evaluation about various actions and the tolerance of different types of errors. Different users may give different values depending on how critical it is for them to process a spam in the *Inbox* folder, to delete the *Spam* folder, and to delay processing of the *Suspected* folder.

As shown in Fig. 4, there are four types of loss functions in the binary cost matrix. According to the minimum risk decision rules in Bayesian decision theory, we accept an email if the expected risk is smaller than rejecting the email, that is, $R(a_P|[x]) \leq$

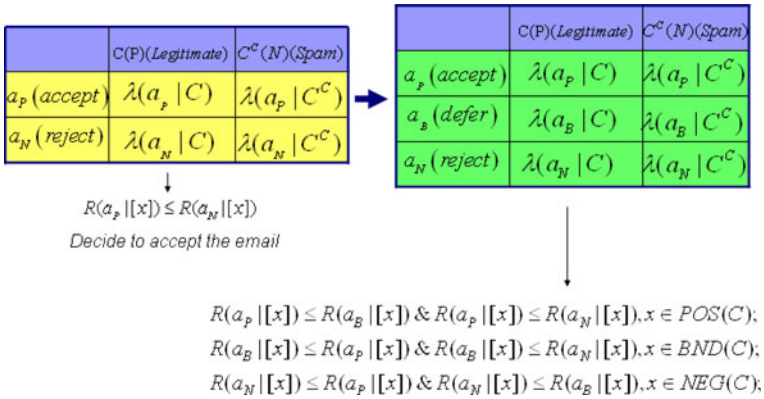


Fig. 4 From binary cost matrix to ternary cost matrix

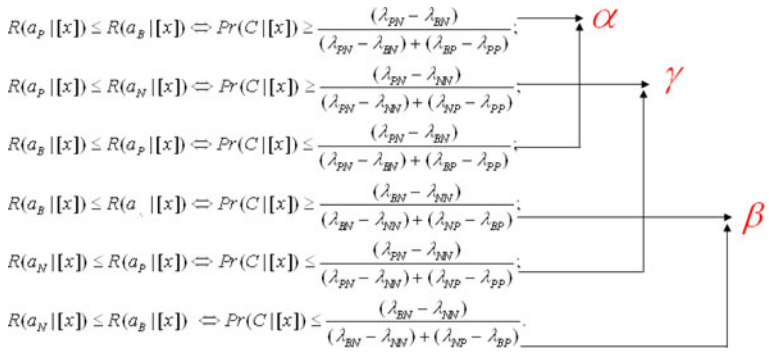


Fig. 5 Estimating thresholds

$R(a_N|[x])$. In the cost matrix for ternary classification, there are six types of loss functions. In order to make a decision, we need to do a pairwise comparison of the three expected risks.

As explained in Section 4.2, the estimation of the required thresholds α and β is illustrated in Fig. 5. It maybe reasonable to impose the constraint $\alpha > \beta$ so that the boundary region may be non-empty, and the loss incurred for deferment should be in between of accept and reject, therefore, we get $1 \geq \alpha > \gamma > \beta \geq 0$. After tie-breaking, γ is no longer needed.

In short, a three-way decision can be made based on the pairwise comparison of the three expected risks: $R(a_P|[x])$, $R(a_B|[x])$ and $R(a_N|[x])$. Each of these comparisons can be translated into the comparison between the a posteriori probability $Pr(C \mid [x])$ and a pair of thresholds α and β, where (α, β) can be calculated based on the loss functions as shown in Section 4.2, and $Pr(C \mid [x])$ can be calculated based on Bayesian inference as shown in Section 4.3.

Tables 1 and 2 give the loss functions of two users, Users 1 and 2, respectively. It can be seen that User 1 is more concerned about losing a legitimate email and at the same time about processing a spam. In comparison, User 2 is not so much concerned. The pair of thresholds α^1 and β^1 for User 1 is calculated according to (18) as:

$$\alpha^1 = \frac{\left(\lambda^1_{PN} - \lambda^1_{BN}\right)}{\left(\lambda^1_{PN} - \lambda^1_{BN}\right) + \left(\lambda^1_{BP} - \lambda^1_{PP}\right)} = \frac{10-5}{(10-5)+(5-0)} = 0.50,$$

$$\beta^1 = \frac{\left(\lambda^1_{BN} - \lambda^1_{NN}\right)}{\left(\lambda^1_{BN} - \lambda^1_{NN}\right) + \left(\lambda^1_{NP} - \lambda^1_{BP}\right)} = \frac{5-0}{(5-0)+(90-5)} = 0.06.$$

The pair of thresholds α^2 and β^2 for User 2 is calculated as:

$$\alpha^2 = \frac{\left(\lambda^2_{PN} - \lambda^2_{BN}\right)}{\left(\lambda^2_{PN} - \lambda^2_{BN}\right) + \left(\lambda^2_{BP} - \lambda^2_{PP}\right)} = \frac{8-5}{(8-5)+(5-0)} = 0.38,$$

Table 1 Loss function of User 1

	C (P)(Legitimate)	C^c (N)(Spam)
a_P(Accept)	$\lambda^1_{PP} = 0$	$\lambda^1_{PN} = 10$
a_B(Defer)	$\lambda^1_{BP} = 5$	$\lambda^1_{BN} = 5$
a_N(Reject)	$\lambda^1_{NP} = 90$	$\lambda^1_{NN} = 0$

Table 2 Loss function of User 2

	$C\ (P)$(Legitimate)	$C^c\ (N)$(Spam)
a_P(Accept)	$\lambda^2_{PP} = 0$	$\lambda^2_{PN} = 8$
a_B(Defer)	$\lambda^2_{BP} = 5$	$\lambda^2_{BN} = 5$
a_N(Reject)	$\lambda^2_{NP} = 15$	$\lambda^2_{NN} = 0$

$$\beta^2 = \frac{(\lambda^2_{BN} - \lambda^2_{NN})}{(\lambda^2_{BN} - \lambda^2_{NN}) + (\lambda^2_{NP} - \lambda^2_{BP})} = \frac{5-0}{(5-0)+(15-5)} = 0.33.$$

It follows that $\beta^1 < \beta^2 < \alpha^2 < \alpha^1$. As expected, the thresholds of User 2 are within the thresholds of User 1, which shows that User 1 is much critical than User 2 regarding both incorrect acceptance and rejection. Consequently, User 1 would have smaller accepted and rejected folders but a large deferred folder. In contract, User 2 would have larger accepted and rejected folders but a smaller deferred folder.

For a new email *Eml*7, suppose its probability of being legitimate can be calculated from the training set as shown in Fig. 6, that is, $Pr(Legitimate|[x]) = 0.3$. For User 1, *Eml*7 will be classified into the deferment folder for further examination because $0.06 < Pr(Legitimate|[x]) < 0.50$, but for User 2, *Eml*7 will be rejected because $Pr(Legitimate|[x]) \geq 0.38$. Different filtering options are tailored to meet individual requirements in terms of minimum overall cost based on our unified framework.

More specifically, users are involved in the setup of the loss functions (costs) in Tables 1 and 2. For example, in Table 1, misclassifying an legitimate email into spam is nine times more costly than misclassifying a spam email email into legitimate (90 vs 10), whereas in Table 2, misclassifying an legitimate email into spam is less than two times more costly than misclassifying a spam email email into legitimate (15 vs 8). That's because User 1 is more concerned about losing a legitimate email than User 2, so he/she can set up a higher cost value. It also makes more sense for the user to assign a loss function instead of a threshold because loss functions can be semantically

Fig. 6 An example

The training set

Email	Welcome	service	receive	class
Eml1	2	5	3	Spam
Eml2	1	2	4	Legitimate
Eml3	1	1	3	Legitimate
Eml4	8	0	2	Spam
Eml5	5	3	0	Spam
Eml6	0	1	3	Legitimate

The testing set

Email	Welcome	service	receive	class
Eml7	2	5	3	?

User1

Cost matrix	Legitimate	Spam
Accept	0	10
Defer	5	5
Reject	90	0

$\alpha = 0.50, \beta = 0.06$

User2

Cost matrix	Legitimate	Spam
Accept	0	8
Defer	5	5
Reject	15	0

$\alpha = 0.38, \beta = 0.33$

related to more practically operable notions such as cost, risk, benefit etc. When $\alpha = \beta = 0.5$, the ternary classification is the same as the binary classification systems. We can use this as the initial setup, and the user has the flexibility of choosing their own cost values which will lead to bigger or smaller positive regions (accepted folders).

5 Experiments and evaluations

5.1 Dataset preparations

Our experiments were performed on three different datasets, the PU1 corpus (Androutsopoulos et al. 2000), the Ling-Spam corpus (Androutsopoulos et al. 2000), and a spambase data set from UCI Machine Learning Repository (http://www.ics.uci.edu/mlearn/MLRepository.html). Our goal is to compare the cost-sensitive three-way decision approach with the original naive Bayesian spam filter, Robinson's approach and the rough set approach on these three different datasets.

For the PU1 corpus, we selected the emails under the *bare* folder as the dataset, there were 1,099 emails, 481 were spam, 618 were legitimate. Since the corpus was divided into ten parts, 10-fold cross validation was used with one part reserved for testing set at each repetition. After scanning the training set, we removed the tokens that appear less than 5 % and more than 95 % in all the emails, because we want to reduce the feature (attribute) space and only keep those tokens that differentiate spam and legitimate emails. Information gain was used as the feature selection method to further reduce the feature space and the top 300 attributes were selected as the feature set. For the Ling-Spam corpus, we selected the emails under the *lemm_stop* folder as the dataset, there were 2,412 legitimate emails from a linguistic mailing list and 481 spam ones collected by the author. 10-fold cross validation was used and the same feature selection strategy was used, the top 150 attributes were selected as the feature set based on their information gain. The UCI spambase data set consisted of 4,601 instances, with 1,813 instances as spam, and 2,788 instances as legitimate, each instance was described by 58 attributes. We split the data set into a training set of 3,681 instances, and a testing set of 920 instances. Entropy-MDL (Fayyad and Irani 1993) is used as the discretization method applied to both the training and testing data sets. The best-first search is used for attribute selection and 15 attributes are selected. For the rough set approach, the set of decision rules is induced by the genetic algorithm intergraded in the rough set tool kit, Rosetta (http://rosetta.lcb.uu.se).

5.2 Evaluation measures

Spam recall and *spam precision* have been used as indicators for measuring email spam filter performance (Androutsopoulos et al. 2000; Sahami et al. 1998). They are defined as:

$$spam\ recall = \frac{n_{S\to S}}{n_{S\to S} + n_{S\to L}},$$

$$spam\ precision = \frac{n_{S\to S}}{n_{S\to S} + n_{L\to S}},$$

where $n_{S\to S}$ denotes the number of emails classified as spam which truly are, $n_{L\to S}$ denotes the number of legitimate emails classified as spam, and $n_{S\to L}$ denotes the number of spam emails classified as legitimate. Since misclassifying an legitimate email to spam is more costly than misclassifying a spam to legitimate, we consider *spam precision* as the main indicator for the non-cost-sensitive evaluation, the ternary email spam filtering should increase the *spam precision* in comparison with the original binary naive Bayes classifier (i.e., $n_{L\to S}$ decreases).

For cost-sensitive evaluations, we assume that misclassifying a legitimate email as spam is w times more costly than misclassifying a spam email as legitimate (i.e., $(\frac{Pr(C|\mathbf{x})}{Pr(C^c|\mathbf{x})})^{-1}=\frac{Pr(C^c|\mathbf{x})}{Pr(C|\mathbf{x})} > w$). Three different w values ($w = 1$, $w = 3$, and $w = 9$) were used for the original naive Bayesian spam filter. Three sets of loss functions for the three-way decision approach were provided accordingly with the same cost ratios. For instance, when we use $w = 9$ for the naive Bayesian spam filter, $\lambda_{NP}/\lambda_{PN} = 9$ was used in the three-way decision approach. Three pairs of thresholds (α, β) were calculated based on these three sets of loss functions, respectively, to distinguish the three email categories in ternary email spam filtering. The *weighted accuracy*, *weighted error rate* and *total cost ratio* suggested by Androutsopoulos et al. (2000), the *cost* measure suggested by Yao (2011), and the *cost curve* suggested by Drummond and Holte (2000, 2006) were used as our cost-sensitive evaluation measures.

The *weighted accuracy* is defined as:

$$WAcc = \frac{w \cdot n_{L\to L} + n_{S\to S}}{w \cdot N_L + N_S},$$

where $n_{L\to L}$ denotes the number of emails classified as legitimate which truly are, N_L and N_S are the number of legitimate and spam emails to be classified by the spam filter. Similarly, we can get the *weighted error rate* as follows:

$$WErr = \frac{w \cdot n_{L\to S} + n_{S\to L}}{w \cdot N_L + N_S}.$$

It is important to compare the *weighted accuracy* and *weighted error rate* to a baseline approach to avoid the often high accuracy and low error rate. A baseline is defined as the case where all legitimate emails are never blocked, and all the spam emails always pass the filter. The *weighted error rate* of the baseline is defined as:

$$WErr^b = \frac{N_S}{w \cdot N_L + N_S}.$$

The *total cost ratio* is defined as:

$$TCR = \frac{WErr^b}{WErr} = \frac{N_S}{w \cdot n_{L\to S} + n_{S\to L}}.$$

Greater *TCR* indicates better performance. If cost is proportional to wasted time, *TCR* measures how much time is wasted to delete manually all spam emails when no filter is used, compared to the time wasted to delete manually any spam emails that passes the filter plus the time needed to recover from mistakenly blocked legitimate emails. The ternary email spam filtering introduce two additional types of misclassification errors besides the original incorrect acceptance and incorrect rejection, namely, deferment of positive and deferment of negative. In other words, the misclassification rate is reduced by the deferment errors. Therefore, the ternary

email spam filtering should decrease the *weighted error rate* (i.e., its numerator decreases), increase the *TCR* (i.e., its denominator decreases).

Suppose the classification results of the ternary email spam filtering are represented by the following 3×2 contingency table:

	C (P): positive	C^c (N): negative
a_P: accept	n^t_{PP}	n^t_{PN}
a_B: defer	n^t_{BP}	n^t_{BN}
a_N: reject	n^t_{NP}	n^t_{NN}

The *cost* measure for ternary classification models is defined as:

$$Cost^t = \frac{1}{U}\left[\left(\lambda_{PP}n^t_{PP} + \lambda_{PN}n^t_{PN}\right) + \left(\lambda_{BP}n^t_{BP} + \lambda_{BN}n^t_{BN}\right) + \left(\lambda_{NP}n^t_{NP} + \lambda_{NN}n^t_{NN}\right)\right],$$

where U denotes the number of examples in the testing set, λ_{PP}, λ_{PN}, λ_{BP}, λ_{BN}, λ_{NP} and λ_{NN} are the loss functions defined in Section 4.2. Assume the classification results of the binary models are represented by the following 2×2 contingency table:

	C (P): positive	C^c (N): negative
a_P: accept	n^b_{PP}	n^b_{PN}
a_N: reject	n^b_{NP}	n^b_{NN}

The *cost* measure for binary models is defined as:

$$Cost^b = \frac{1}{U}\left[\left(\lambda_{PP}n^b_{PP} + \lambda_{PN}n^b_{PN}\right) + \left(\lambda_{NP}n^b_{NP} + \lambda_{NN}n^b_{NN}\right)\right],$$

Note that n^b_{NP} and n^t_{NP} correspond to $n_{L\to S}$ in the *weighted error rate* and *TCR* measures, and n^b_{PN} and n^t_{PN} correspond to $n_{S\to L}$. We use the symbols from the original paper (Yao 2011) for the consistency with the subscripts of the loss functions. As it has been proved by the author (Yao 2011), the associated costs of a ternary classification model are always less than the binary model. The main advantage of the *cost* measure is that it takes deferment cost and error into consideration, while the other measures only consider the acceptance and rejection errors.

Cost curve is an alternative to *ROC curve* in which the expected cost of a classifier is represented explicitly (Drummond and Holte 2000, 2006). The x-axis in a cost curve is the probability-cost function for positive examples, which is defined as:

$$PCF(C) = \frac{Pr(C)\lambda_{NP}}{Pr(C)\lambda_{NP} + Pr(C^c)\lambda_{PN}}.$$

The y-axis is the expected cost normalized with respect to the cost incurred when every example is incorrectly classified, which is defined as:

$$NE(\lambda) = (1 - TP - FP) * PCF(C) + FP,$$

where TP is the true positive rate, and FP is the false positive rate. If one classifier is lower in expected cost across the whole range of the probability-cost function, it dominates the other.

Table 3 Comparison results on PU1 corpus

Thresholds	Approaches	Cost	WAcc (%)	WErr (%)	TCR	Spam	
						Precision (%)	Recall (%)
	NB	2.73	81.81	18.18	2.25	80.49	73.33
$w = 1$	Three-way	2.07	74.55	10.00	4.09	89.66	57.78
$\alpha = 0.75$	Robinson	3.72	0.90	0.00	0.00	100.00	2.22
$\beta = 0.25$	RS	3.99	50.91	19.10	2.14	58.62	37.78
	NB	3.82	88.33	11.67	1.61	85.37	77.78
$w = 3$	Three-way	2.97	83.75	7.08	2.65	90.91	66.67
$\alpha = 0.65$	Robinson	5.36	0.42	0.00	0.00	100.00	2.22
$\beta = 0.15$	RS	8.43	50.42	19.58	0.96	57.58	42.22
	NB	3.13	91.59	8.41	0.85	88.10	82.22
$w = 9$	Three-way	1.79	86.35	5.40	1.32	91.18	68.89
$\alpha = 0.55$	Robinson	2.18	0.16	0.00	0.00	100.00	2.22
$\beta = 0.05$	RS	7.54	39.21	23.17	0.31	57.89	48.89

5.3 Results and analysis

Tables 3, 4 and 5 show the evaluation results on three different datasets. We can see that Robinson's approach has better performance in non-cost-sensitive evaluations, but provides poor performance in cost-sensitive settings on all three datasets (e.g., low *WAcc* and TCR). The rough set approach produces poor results on PU1 corpus for both the cost-sensitive and non-cost-sensitive evaluations (e.g., low *spam precision* and *WAcc*, high *WErr* and *cost*). On average, the three-way decision approach provides better results than the original naive Bayes classifier, and outperforms the other two ternary spam filtering methods for the cost-sensitivity aspect. That is, lower *cost* and *weighted error*, higher *TCR*. When w increases, *spam precision* increases for both the naive Bayes classifier and the three-way decision approach, and the *weighted error* and *TCR* decrease. On the other hand, Robinson's approach and the rough set approach are not sensitive to different cost settings. Their cost-sensitive

Table 4 Comparison results on Ling-Spam corpus

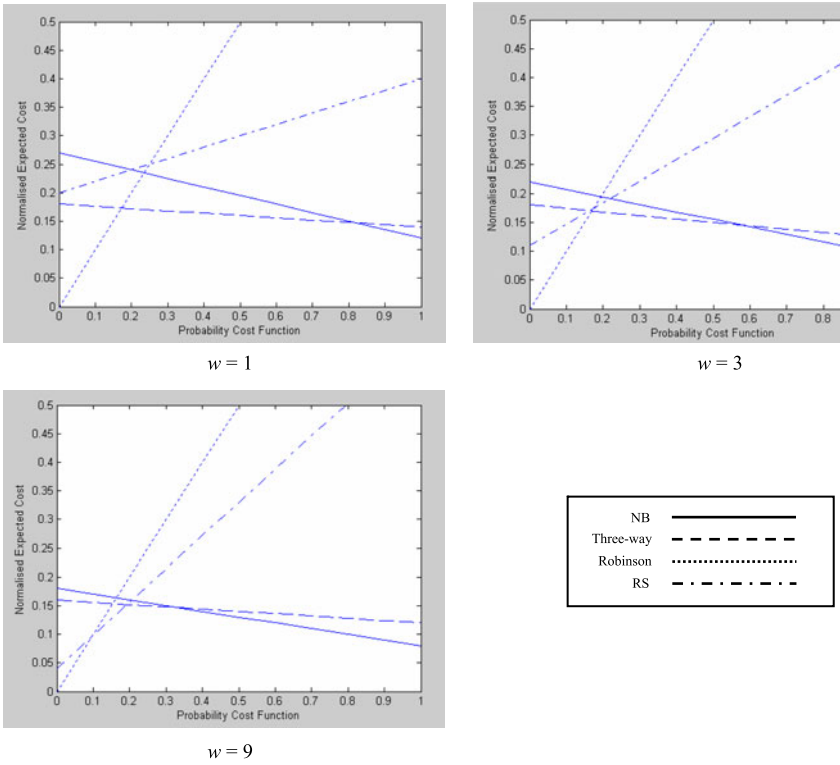
Thresholds	Approaches	Cost	WAcc (%)	WErr (%)	TCR	Spam	
						Precision (%)	Recall (%)
	NB	1.71	88.58	11.42	1.61	72.73	60.38
$w = 1$	Three-way	1.63	88.58	10.73	1.71	74.42	60.38
$\alpha = 0.75$	Robinson	3.54	6.57	0.35	53.00	95.00	35.85
$\beta = 0.25$	RS	1.99	83.39	12.11	1.51	82.35	26.42
	NB	2.80	92.90	7.10	0.98	74.42	60.38
$w = 3$	Three-way	2.53	92.51	6.31	1.10	74.42	60.38
$\alpha = 0.65$	Robinson	4.43	3.15	0.79	8.83	92.31	45.28
$\beta = 0.15$	RS	2.32	89.09	4.47	1.56	80.00	30.19
	NB	1.80	95.22	4.78	0.51	76.92	56.60
$w = 9$	Three-way	1.66	93.94	4.27	0.57	78.38	54.72
$\alpha = 0.55$	Robinson	2.28	1.29	0.83	2.94	93.33	52.83
$\beta = 0.05$	RS	1.03	88.84	1.88	1.29	85.00	32.08

Table 5 Comparison results on UCI spambase dataset

Thresholds	Approaches	Cost	WAcc (%)	WErr (%)	TCR	Spam	
						Precision (%)	Recall (%)
	NB	1.79	88.04	11.96	3.34	88.13	80.93
$w = 1$	Three-way	1.07	84.89	4.46	8.95	94.70	77.93
$\alpha = 0.75$	Robinson	0.08	97.93	0.00	0.00	100.00	98.09
$\beta = 0.25$	RS	2.29	82.61	14.57	2.74	97.38	60.76
	NB	2.11	93.63	6.37	2.84	94.70	77.93
$w = 3$	Three-way	1.72	88.75	4.05	4.48	97.56	76.29
$\alpha = 0.65$	Robinson	0.17	95.21	0.00	0.00	100.00	98.91
$\beta = 0.15$	RS	2.55	90.72	7.26	2.50	97.01	61.85
	NB	0.91	96.86	3.14	2.18	97.99	66.49
$w = 9$	Three-way	0.84	89.88	1.78	3.86	98.55	55.59
$\alpha = 0.55$	Robinson	0.56	61.19	0.00	0.00	100.00	98.91
$\beta = 0.05$	RS	1.08	94.70	3.54	1.94	97.02	62.13

evaluation results are poor. For instance, the rough set approach has higher *weighted error*, higher *cost*, and lower *TCR* comparing to other spam filters, and Robinson's approach has lower *TCR* and higher *cost* than the three-way decision approach.

Figure 7 shows the *cost curves* of four email spam filters on PU1 corpus under three different cost settings. As we can see, Robinson's approach and the rough set



$w = 1$ $w = 3$ $w = 9$

Fig. 7 Comparison results of cost curves on PU1 corpus for different cost settings

approach have high expected cost under all three settings. The three-way decision approach has lower expected cost than the naive Bayes classifier across the most range of the probability-cost function when $w = 1$ and $w = 3$, and has a close expected cost with the naive Bayes classifier when $w = 9$. Figure 8 shows the *cost curves* on Ling-Spam corpus. Robinson's approach has high expected cost under all three settings. The rough set approach dominates the other three spam filters when $w = 9$, but has higher expected costs when $w = 1$ and $w = 3$. The three-way decision approach dominates the naive Bayes classifier under all three settings. Figure 9 shows the *cost curves* on the UCI dataset. The rough set approach has high expected costs under all three settings. Robinson's approach dominates the other three spam filters when $w = 1$ and $w = 3$, but has a higher expected cost when $w = 9$. The cost-sensitive three-way decision approach has lower expected cost than the naive Bayes classifier across the most range of the probability-cost function under all three settings.

Overall, for the cost-sensitive evaluations, the three-way decision approach has lowest *cost* comparing to other three existing spam filters. It is sensitive to different cost settings and consistently performs better than other spam filters on all three datasets. Robinson's approach has better *spam precision* for the non-cost-sensitivity aspect, but provides poor performance in the cost-sensitive evaluations. Among the three ternary spam filters, the three-way decision approach is the only one that outperforms the original naive Bayesian spam filter in both the cost-sensitive and non-cost-sensitive settings on all three datasets.

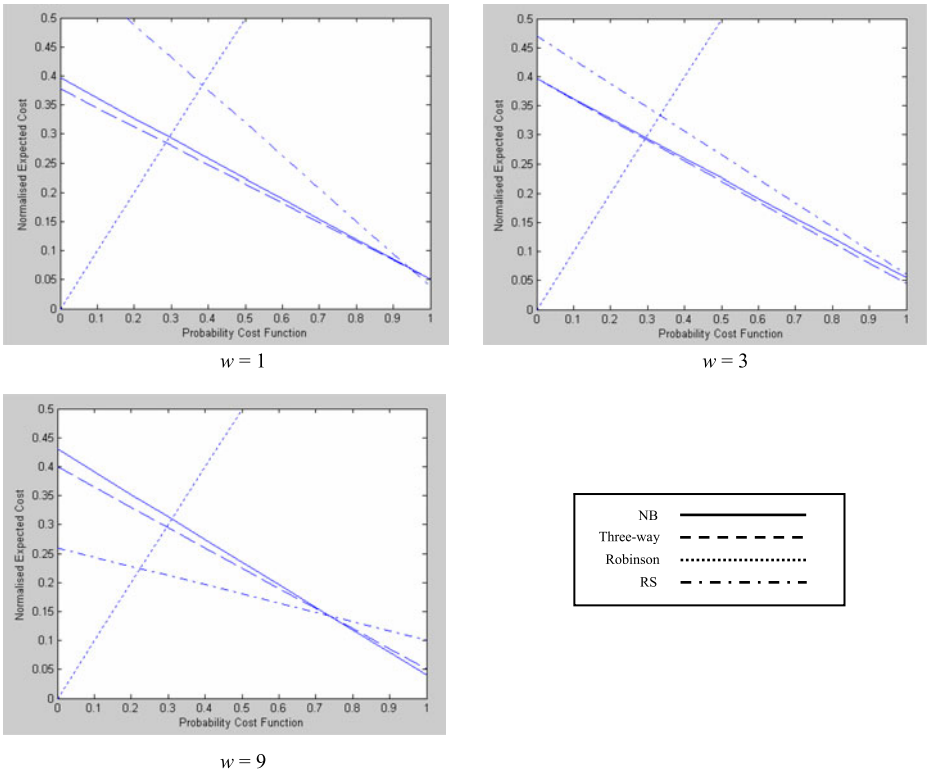


Fig. 8 Comparison results of cost curves on Ling-Spam corpus for different cost settings

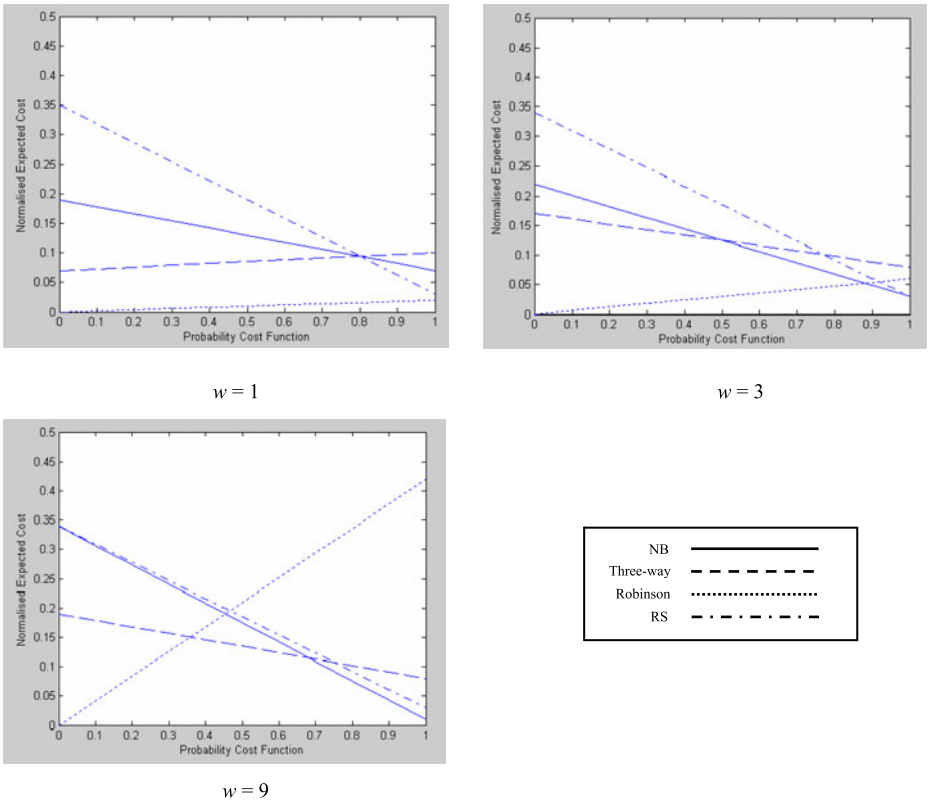


Fig. 9 Comparison results of cost curves on UCI dataset for different cost settings

Note that in these experiments, we compare four different email spam filters based on their classification accuracy and cost, not their time efficiency, since computational time is not the main concern of this paper.

6 Conclusion

A cost-sensitive three-way decision approach to email spam filtering is presented in this paper. Compared to the most commonly used binary email spam filtering, a boundary region marked unsure is added to the classification results to convert potential misclassification errors into errors of deferment. The main advantage of the three-way decision approach is that it allows the possibility of indecision, i.e., of refusing to make a decision. The undecided cases must be reexamined by collecting additional information. A pair of thresholds are used. The first threshold determines the point necessary for a re-examination, and the second threshold determines the point to reject an email. The main difference between our approach and other existing ternary email spam filtering methods is the computation of the required thresholds. Instead of supplying them based on intuitive understandings or trail and error, we provides a systematically calculation based on the decision-theoretic rough

set model. The cost associated with each decision is given by a loss function from the well established Bayesian decision theory. The cost-sensitive characteristic of email spam filtering is reflected by varying the values of loss functions. The experiment results on several benchmark datasets show that the new approach outperforms other spam filters in cost-sensitive evaluations. This three-way decision approach can also be applied to junk eliminations of other types of web documents such as web pages and blogs with slight modifications.

Acknowledgement The authors are grateful for the financial support from NSERC Canada.

References

Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G., Spyropoulos, C.D. (2000). An evaluation of naive Bayesian anti-spam filtering. In *Proc. of the workshop on machine learning in the new information age*.

Barracuda Spam Firewall (2012). From http://www.barracudanetworks.com. Accessed 25 July 2012.

Bogofilter (2012). From http://bogofilter.sourceforge.net. Accessed 25 July 2012.

Cohen, W. (1996). Learning rules that classify email. In *Advances in inductive logic programming*.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Drummond, C., & Holte, R.C. (2000). Explicitly representing expected cost: an alternative to ROC representation. In *KDD 2000* (pp. 198–207).

Drummond, C., & Holte, R.C. (2006). Cost curves: an improved method for visualizing classifier performance. *Machine Learning, 65*(1), 95–130.

Duda, R.O., & Hart, P.E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Elkan, C. (2001). The foundations of cost-senstive learning. In *Proceedings of the 17th international joint conference on artificial intelligence* (pp. 973–978).

Fayyad, U.M., & Irani, K.B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th international joint conference on artificial intelligence* (pp. 1022–1029).

GFI MailEssentials (2012). http://www.gfi.com/. Accessed 25 July 2012.

Good, I.J. (1965). *The estimation of probabilities: An essay on modern Bayesian methods*. Cambridge: MIT Press.

Graham, P. (2002). A Plan for spam. http://www.paulgraham.com/spam.html. Accessed 25 July 2012.

Masand, B., Linoff, G., Waltz, D. (1992). Classifying news stories using memory based reasoning. In *Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 59–65).

Mitchell, T. (1997). *Machine learning*. New York: McGraw Hill.

Pantel, P., & Lin, D.K. (1998). SpamCop—a spam classification & organization program. In *Proceedings of AAAI workshop on learning for text categorization* (pp. 95–98). Madison, WI.

Rennie, J. (1996). "ifile". http://people.csail.mit.edu/jrennie/ifile/. Accessed 25 July 2012.

Robinson, G. (2004). A statistical approach to the spam problem, spam detection. In *Why Chi? Motivations for the use of fishers inverse Chi-square procedure in spam classification. Handling redundancy in email token probabilities*.

Sahami, M., Dumais, S., Heckerman, D., Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In *AAAI workshop on learning for text categorization*. AAAI Technical Report WS-98-05, Madison, Wisconsin.

Schapire, E., & Singer, Y. (2000). BoosTexter: a boosting-based system for text categorization. *Machine Learning, 39*(2/3), 135–168.

Siersdorfer, S., & Weikum, G. (2005). Using restrictive classification and meta classification for junk elimination. In *Proceedings of ECIR'2005* (pp. 287–299).

Triola, M.F. (2005). *Elementary statistics*. Reading: Addison Wesley.

Yao, Y.Y. (2011). The superiority of three-way decisions in probabilistic rough set models. *Information Sciences, 181*, 1080–1096.

Yao, Y.Y., Wong, S.K.M., Lingras, P. (1990). A decision-theoretic rough set model. In Z.W. Ras, M. Zemankova, M.L. Emrich (Eds.), *Methodologies for intelligent systems* (Vol. 5, pp. 17–24). New York: North Holland.

Yerazunis, W.S. (2003). Sparse binary polynomial hashing and the CRM114 discriminator. In *Proceedings of the MIT spam conference*.

Yih, W., McCann, R., Kolcz, A. (2007). Improving spam filtering by Detecting Gray mail. In *Proceedings of the 4th conference on e-mail and anti-spam (CEAS07)*.

Zhao, W., & Zhang, Z. (2005). An email classification model based on rough set theory. In *Proceedings of the international conference on active media technology* (pp. 403–408).

Zhou, Z.H., & Liu, X.Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering, 18*(1), 63–77.

Zhou, Z.H., & Liu, X.Y. (2010). On multi-class cost-sensitive learning. *Computational Intelligence, 26*(3), 232–257.

Zhou, B., & Liu, Q.Z. (2012). A comparison study of cost-sensitive classifier evaluations. In *The 2012 international conference on brain informatics (BI'12). Lecture notes in computer science* (Vol. 7670, pp. 360–371).

Zhou, B., Yao, Y.Y., Luo, J.G. (2010). A three-way decision approach to email spam filtering. In *Proceedings of the 23th Canadian conference on artificial intelligence (AI 2010), University of Ottawa, Ontario, Canada, 31 May–2 June 2010. Lecture notes in artificial intelligence* (pp. 28–39).