Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 1599–1608
Expert Systems with Applications
www.elsevier.com/locate/eswa

An incremental cluster-based approach to spam filtering

Wen-Feng Hsiao [a,*], Te-Min Chang [b]

[a] Department of Information Management, National Pingtung Institute of Commerce, Taiwan
[b] Department of Information Management, National Sun Yat-sen University, Taiwan

Abstract

As email becomes a popular means for communication over the Internet, the problem of receiving unsolicited and undesired emails, called spam or junk mails, severely arises. To filter spam from legitimate emails, automatic classification approaches using text mining techniques are proposed. This kind of approaches, however, often suffers from low recall rate due to the natures of spam, skewed class distributions and concept drift. This research is thus to propose an appropriate classification approach to alleviating the problems of skewed class distributions and drifting concepts. A cluster-based classification method, called ICBC, is developed accordingly. ICBC contains two phases. In the first phase, it clusters emails in each given class into several groups, and an equal number of features (keywords) are extracted from each group to manifest the features in the minority class. In the second phase, we capacitate ICBC with an incremental learning mechanism that can adapt itself to accommodate the changes of the environment in a fast and low-cost manner. Three experiments are conducted to evaluate the performance of ICBC. The results show that ICBC can effectively deal with the issues of skewed and changing class distributions, and its incremental learning can also reduce the cost of re-training. The feasibility of the proposed approach is thus justified.

Keywords: Email classification; Skewed class distribution; Concept drift; Incremental learning

1. Introduction

With the rapid growth of the Internet, email has become one of the most common media for us to distribute information. More and more people depend on it to communicate because of its properties of convenience, no restrictions in time and location, prompt delivery, and low cost. Some people, however, abuse it to spread large amount of information ranging from ads of drug, easy money, porn, to political promotion. As a result, our mailboxes are usually filled with unsolicited emails (spam or junk mails). Spam filtering becomes one of the essential issues to most companies, governments, or even individual users.

To resolve the spam problem, traditional filters employed simple and straightforward methods, called filtering by rules, which classify spam emails by matching particular email fields (e.g. sender, recipient, date, subject, and attachment) with certain keywords. Although certain amount of spam can be filtered out, more complex spam (for example, those that can hide their own identities, including sending machines and/or senders) can survive easily with the method applied.

More recent and advanced methods rely more on the content analyses of email bodies to deal with complex spam. This kind of methods extracts features (keywords) from both legitimate and spam emails, and then builds a classification model using techniques from text mining. However, their main problem is the poor performance in the recall rate resulting from the natures of spam itself (Fawcett, 2003), including skewed class distributions and concept drift. In the case of skewed class distributions, the number of spam emails received is far more than that

* Corresponding author. Tel.: +886 8 7238700x6201.
E-mail addresses: wfhsiao@mail.npic.edu.tw (W.-F. Hsiao), temin@mail.nsysu.edu.tw (T.-M. Chang).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.01.018

of legitimate ones so that the class distributions of legitimate and spam are largely uneven. This often leads to a poor recall rate for the legitimate emails (the minority class). For the concept drift problem, users' preferences may change with time or the topics of spam (or legitimate emails) may vary according to the fashionable trends. Therefore, the recall rates for both spam and legitimate emails can be low due to the changes. The learned structure of a classifier should adapt itself to accommodate these changes to classify new emails correctly.

The purposes of this research are thus to propose an appropriate classification method to alleviate the above-mentioned problems, and to improve the effectiveness of spam filtering accordingly. We develop an adaptive cluster-based classification method, called incremental clustering-based classification (ICBC), for such purposes. Basically, we deal with the class-skewed problem by first clustering emails in each class (spam and legitimate emails) into some coherent subsets. An equal number of keywords are then extracted from each subset. By doing so, rare but important keywords of minority classes can be extracted. To deal with the problem of concept drift, we capacitate ICBC with an inherent incremental learning mechanism. With the incremental learning capability, ICBC not only can personalize users' email filtering preferences, but also can adjust itself over time for the changing environments, and thus can improve the classification accuracy under concept drift in a fast and low-cost fashion.

The rest of this paper is organized as follows. The related research works on special spam natures, adaptive learning, cluster analysis, and spam filtering are reviewed in Section 2. The details of the ICBC method are elaborated in Section 3, followed by the presentation of experiments and results for evaluating the effectiveness of ICBC. Finally, the conclusions and future works are discussed in Section 5.

2. Literature review

2.1. Special spam natures

Emails with spam often exhibit the phenomenon of skewed class distributions. Namely, we receive spam emails far more than legitimate ones in our daily life. In such cases where the class distributions of the data are highly skewed, typical classifiers have difficulty in correctly predicting the classes with few data (minority classes) (Monard & Batista, 2002). Minority classes, however, may even deserve our attention more, for example, customer churn, credit card fraud, insurance fraud, and rare disorders. The cost of not identifying these minority classes correctly is often prohibitive. Typical approaches to deal with the problem of skewed class distributions can be classified into three categories; the method of under-sampling for reducing majority data (Hart, 1968), the method of over-sampling for expanding minority data (Honda, Motizuki, Ho, & Okumura, 1997) and the method of multi-classifier committee (Chan, Fan, Prodromidis, & Stolfo, 1999). The under-sampling approach is to decrease the data of the majority class whereas the over-sampling approach is to increase the data of the minority class. The former may encounter the criticism of not fully making use of all the data, and the latter may induce noise. In contrast, the multi-classifier committee approach proportionately partitions data into several subsets and generate multiple classifiers by training individual subsets. The final prediction of the class is an aggregate from the outputs of all classifiers.

On the other hand, concept drift refers to the varying concepts over time. A classifier predicts the class of an unknown example using the rules (boundaries) that are induced from training examples to discriminate the classes. If the learned concepts (the discrimination boundaries) are not changed, the classifier's performance is about the same. However, when the concepts drift as time flies by, the original classifier cannot perform well unless it keeps learning the emerging concepts and modifying its learnt structure accordingly. Spam emails are a good example of concept drift as users' preferences change frequently and the topics of spam also change with contemporary trend. Based on the extent of the drift, three kinds of concept drift are usually discussed in the literature: sudden drift, moderate drift and slow drift (Stanley, 2003). Based on the class distributions, on the other hand, Forman (2006) classified the concept drift into three types: shifting class distribution, shifting subclass distribution, and fickle concept drift. To deal with concept drift, the classifiers should possess incremental learning capability that adapts itself to the environmental changes.

2.2. Adaptive learning in text categorization

To improve the accuracy of a classifier, Wu, Pang, and Liu (2002) proposed a refinement approach to solving the problem of the misfit between data and models. Their method did not modify the classification algorithm itself; instead, it employed an incremental refinement procedure to deal with the model misfit problem. The concept is illustrated in Fig. 1. It is an iterative process with two major steps. The first step is to train the classifier given a data

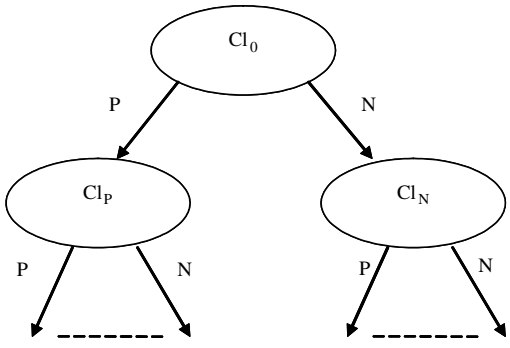

Fig. 1. Concept illustration of the refinement approach (Source: Wu et al., p. 210).

set. Since the classification result based on the trained classifier is usually not perfect, the second step is to split the given data set based on the classifier's current prediction results. The process then iterates to the first step to train a corresponding classifier for each split. It goes on repeatedly until a stop criterion is met (the *F*-measure in this case). Note that in this way, the data set in each split becomes purer and purer, which implies the concept represented at a more descending node would be more consistent.

ICBC is motivated by Wu et al.'s work, which employed an adaptive refinement procedure to partition the data into sub-clusters until the concept represented within a cluster is consistent. However, unlike their computational effort to repeatedly train the classifier, ICBC clusters the data simply by using clustering techniques.

2.3. Cluster analysis

The clustering techniques can usually be divided into two categories: partitional and hierarchical techniques (Jain, Murty, & Flynn, 1999). Partitional clustering usually employs a measure (e.g. squared error) and attempts to optimize this measure (e.g. minimize the square error). The procedure is iterated several times until a stop criterion is met.

Hierarchical clustering can be further divided into agglomerative and divisive clustering techniques. For agglomerative clustering, each data point is initially viewed as a cluster. Clusters are gradually combined together according to their similarities (with the minimal distance criterion). On the other hand, divisive clustering starts from a big cluster containing all the data points, followed by separating the cluster(s) iteratively. The measures employed to calculate the distances between clusters include single, complete, and average links.

(1) Single link: the distance between two clusters can be obtained by finding the minimal distances between them, i.e., the distance between the two closest data points in the two clusters, respectively.
(2) Complete link: the distance between two clusters is to calculate the maximal distances between them, i.e., the distance between the two farthest data points in the two clusters, respectively.
(3) Average link: the distance between two clusters is the mean distance between all possible pairs of data points in the two clusters, respectively.

2.4. Spam filtering research

With text-mining techniques adopted, a spam filter is usually established through the following steps. First, users manually classify their emails into legitimate and spam emails. A feature vector (keywords) is then formed based on TFIDF (term frequency × inverse document frequency) or its variants. Finally, a classifier is constructed using a classification algorithm that makes use of the feature vector to best discriminate the two categories (legitimate and spam).

Payne and Edward (1997) employed a rule-based induction method to automatically learn the classification rules. Their system, called "Magi", was a kind of agent program. It learned how to deal with new incoming emails through observing its user's behaviors (e.g. forwarding, deleting, and storing) toward received emails. Magi used two thresholds, predictive threshold and confidence threshold, as the criteria for firing different level of delegated actions. The predictive threshold was used to determine whether Magi should provide its suggested action to the user for reference or not. On the other hand, the confidence threshold was set at a higher level for the system to automatically execute the action according to its prediction (e.g. forwarding, deleting, or storing).

Bayes probabilistic model is another method that is commonly used in document classification (Lian, 2002). Sahami, Dumais, and Horvitz (1998) used Naïve Bayes to analyze spam. Androutsopoulos, Koutsias, Chandrinos, and Spyropoulos (2000) employed Naïve Bayes to construct email classifiers in spam filtering. Owing to its simple formulation and satisfactory performance, Naïve Bayes classifiers were often used as the benchmark for comparison purpose (Zhang & Oles, 2001). Naïve Bayes, however, assumes that each document can only belong to one category (one-to-one correspondence) and the occurrence between individual terms is mutually independent (Nigam, McCallum, Thrun, & Mitchell, 2000), which are not realistic in the real world context. Therefore, recent researches of Bayesian Network tend to relax those assumptions.

In addition, Drucker, Wu, and Vapnik (1999) employed support vector machine (SVM) to filter spam emails. Their results were compared to those of Ripper, Rocchio, and boosting decision tree. They found that with binary expression in the feature vector of emails, SVM performs best. Overall, SVM was compatible with boosting decision tree but with shorter computational time.

Luo and Zincir-Heywood (2005) designed an SOM-based sequence analysis (SBSA) system for spam filtering. The first part of SBSA was to represent document with a two-level hierarchical SOM architecture, and the second part was to employ a sequence-based *k* nearest neighbor (KNN) classifier for classification. SBSA was compared to the Naïve Bayes classifier and showed superior performance.

Delany and Cunningham (2006) proposed a spam filter system, ECUE, which employed instance-based learning technique to track concept drift. By learning from misclassification emails (either spam or legitimate emails), ECUE could personalize to the specifics of users' preferences, and adapt to the changing natures of spam. However, with its adoption of *k* nearest neighbor algorithm, ECUE had to re-perform the feature selection process and rebuild the instance-base every time the learning mechanism was triggered by the misclassification of emails. Our proposed approach, ICBC, is compatible with ECUE on concept drift with a major exception that ICBC changes

cluster structure incrementally rather than rebuild the instance base every time ECUE is triggered to learn.

3. Proposed approach

The main purpose of ICBC is to handle the problem of skewed class distributions, and of concept drift. It basically includes two phases, each of which deals with the mentioned two problems, respectively. Details of the phases, the classifier training phase and the incremental learning phase, are described as follows.

3.1. Classifier training phase

The class-skewed problem is annoying because classes with more data (majority classes) usually dominate those with less data (minority classes). The consequence is that the classifiers give more weights on the majority classes to maintain higher prediction accuracy. To relieve this problem, one should find a way that can yield comparable weights to different classes while maintaining reasonable prediction accuracy. By inspecting the procedure of document classification task, we realize that a particular step that makes minority classes dominated is the feature selection. The feature selection, commonly based on a variant of TFIDF (see, for example, Lee & Lee, 2006), is to extract a keyword vector that can discriminate different classes. Since it is done for the whole training document set, chances for keywords to be selected from the majority classes are higher than from the minority classes.

To overcome this problem, we propose ICBC, which bases itself on cluster analysis (as illustrated in Fig. 2). For each given class, ICBC groups the emails into several clusters (sub-concepts), and extract an equal number of keywords from each cluster. When a new email comes, ICBC compares the distances between this new email and the keyword vector extracted from each cluster, and assigns it to the cluster that has the minimal distance. By doing so, ICBC can effectively avoid the problem that the keywords in minority classes are usually insignificant to be selected such that documents from minority classes are often misclassified to majority classes.

The purpose of the classifier training phase is to resolve the problem of skewed class distributions. The details of this phase are shown in Fig. 3. Just like instance-based learning, the cluster-based ICBC simply performs feature selection task in this phase without training a classification model. ICBC first employs a stop word list (McCallum, 2002) and the Porter stemming algorithm (1980) to remove possible noisy data and reduce the dimension of feature space. For each given class, ICBC applies TFIDF to extract keywords for clustering purpose. The clustering method adopted here is hierarchical with complete-linkage distance measure. Hierarchical, instead of partitional, clustering is used because the latter suffers from the instability that causes from different initial cluster centers and unknown number of clusters. The hierarchical clustering is also contributive to incremental learning, as explained later in the second phase. After the clustering is done, we re-extract an equal number of keywords from each cluster. The criterion to extract keywords now is TF without IDF since our aim is to find features that represent the cluster rather than those that have higher discrimination ability. Finally, the feature vector is determined for each cluster.

To classify a new email, ICBC adopts the similarity comparison procedure as shown in Fig. 4. The essential key issue here is to select the cluster that most matches the new email, and assign the email the class that the matching cluster belongs to. It is referred to as using unsupervised learning to facilitate the supervised learning task. The similarity is calculated by comparing the keyword vector of each cluster to the new email. The more keywords match, the more similar the email to the cluster. Finally, the cluster that has the largest number of matched key-

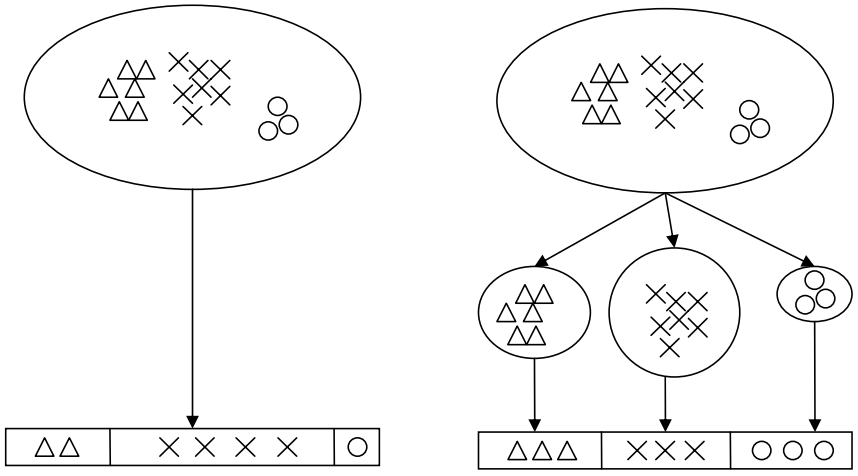

Fig. 2. The conceptual comparison of feature selection between TFIDF and ICBC.

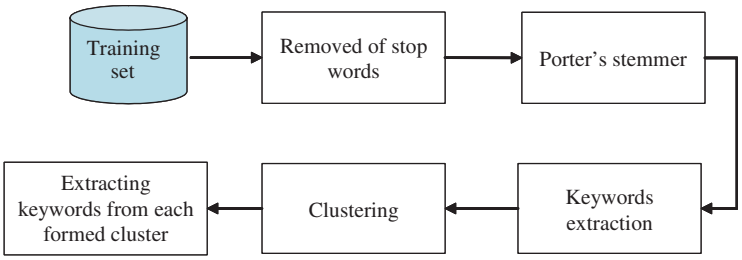

Fig. 3. The procedure of training.

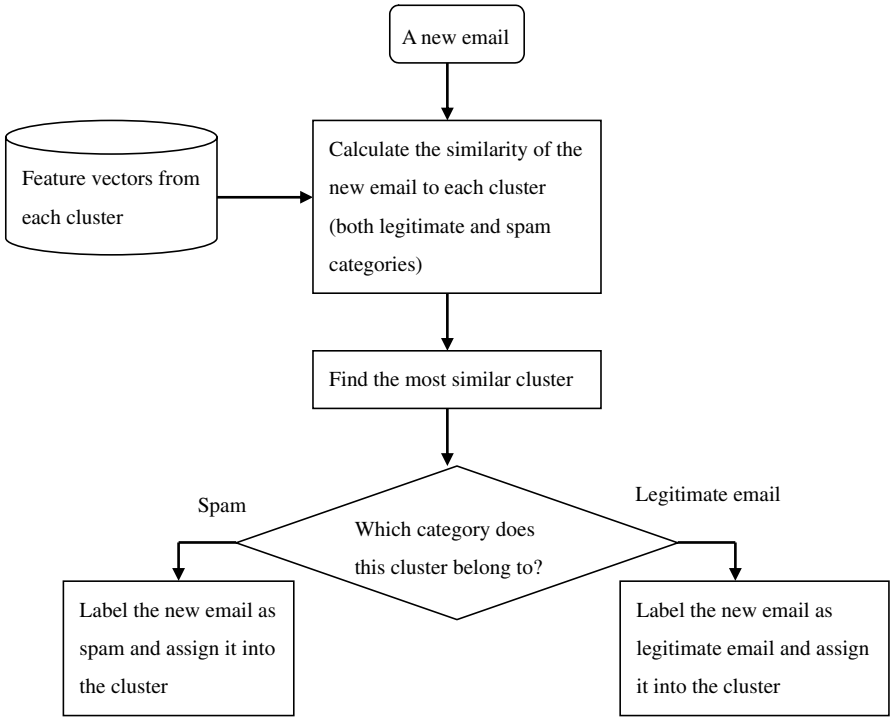

Fig. 4. The procedure to classify an unlabeled email.

words is selected as the most similar cluster, and the class of the new email is then determined.

3.2. Incremental learning phase

The purpose of the incremental learning phase is to impose the incremental learning ability on ICBC. ICBC automatically triggers the incremental learning mechanism when it detects topic changes in spam (or legitimate emails), or when the users correct misclassified emails. In the former situation, ICBC will adapt itself by modifying the existent classification knowledge accordingly, whereas in the latter situation, ICBC quickly learns to avoid the same mistakes and personalizes the users' email filtering preferences.

The incremental learning ability of ICBC lies in its cluster structure of keywords. With new emails coming, the structure can be adjusted to accommodate new information. The procedure of this phase is illustrated in Fig. 5. With the same classification scheme for a new email in the previous phase, the incremental learning occurs when ICBC detects changes in its concepts (sub-clusters), or when users modify misclassified emails.

As we classify a new email and store it to the corresponding cluster, we first consider whether the topic change in spam or legitimate emails is significant or not by checking the split condition of the cluster. If the coherence of the cluster no longer remains, the cluster will be automatically split into two or more sub-clusters, which denotes the generation of new topics. Fig. 6 illustrates the splitting process. Be it the original cluster or the split clusters, we re-extract the keyword vector of the cluster(s) to adapt to the changes. In addition, the cluster is split one at a time, if any, as we deal with the slow concept drift situation (Stanley, 2003) in incremental learning.

Furthermore, users may trigger the incremental learning by modifying the classification results of the new emails. When the modification occurs, the misclassified email will

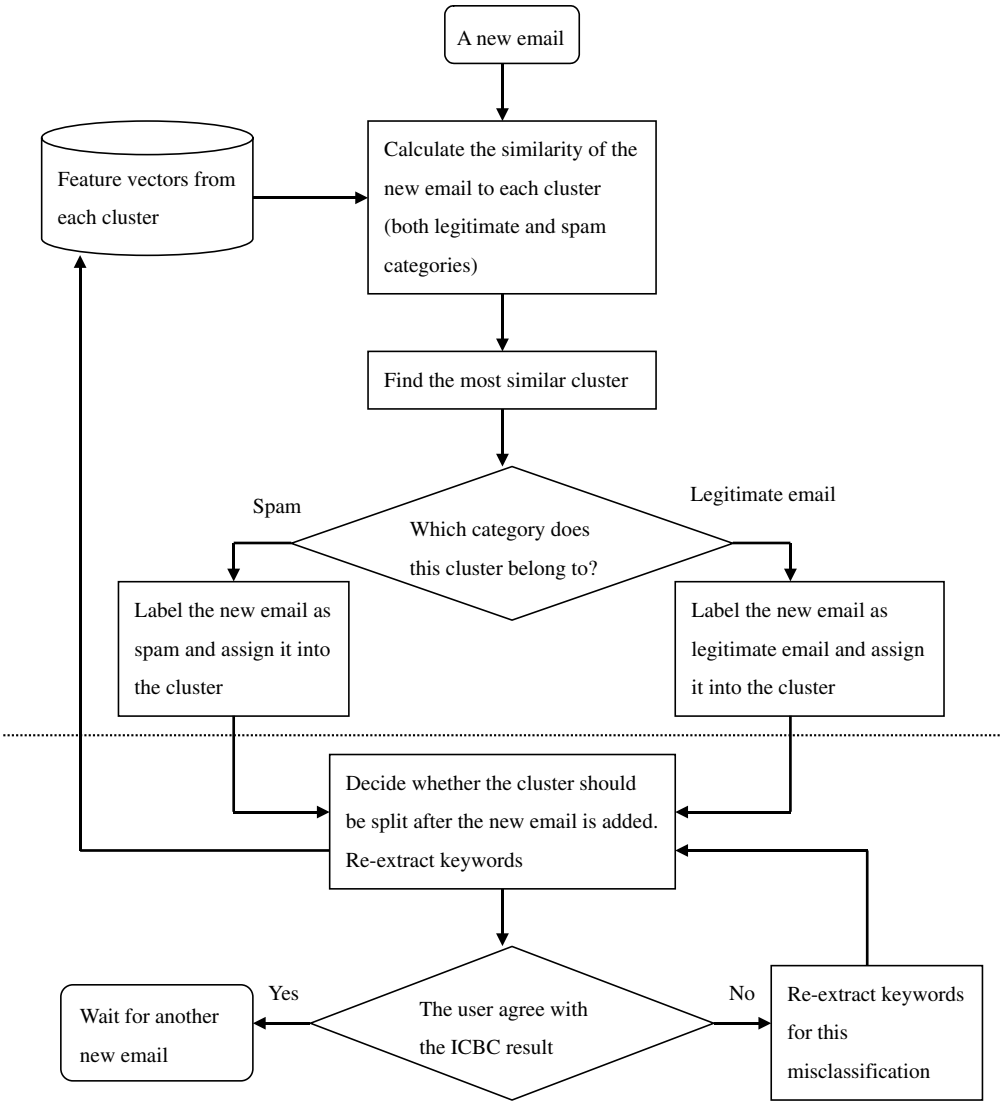

Fig. 5. The procedure of incremental learning in ICBC.

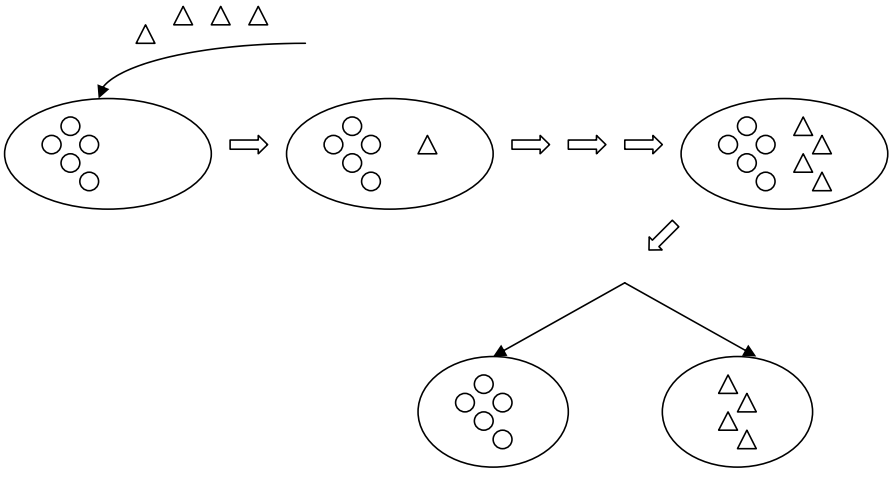

Fig. 6. The illustration of cluster splitting process.

be taken out of where it was stored and re-placed into the most matching cluster in the correct class. Again, for the matching cluster, we need to decide whether it should be split or not. Keywords of the misplaced cluster and the

matching cluster, or the split clusters are re-extracted to adapt to the changes.

Note that ICBC changes the cluster structure before users correct the classification results. The main reason is that users usually process their emails in a batch mode, rather than one-by-one. For the efficiency purpose, ICBC will automatically activate possible structure change before users' correction actions.

4. Experiments and results

In this section we conduct three experiments to illustrate the feasibility of ICBC. We first explain our experimental design, followed by detailing the three experiments.

4.1. Experimental design

We design three experiments to evaluate ICBC. Their main objectives are listed in Table 1.

4.1.1. Data collection

To conduct the experiments, two data sets are collected. The first data set, used for Experiments I and II, is from Spamassassin's[1] non-spam email dataset and E.M. Canada's[2] spam email dataset. The non-spam emails from Spamassassin, with a total of 8624 records, serve as legitimate emails. The emails filtered from E.M. Canada with time frame from January 2002 to December 2003, with a total of 15,213 records, serve as spam. The second data set, used for Experiment III, is from Yahoo newsgroups. Six subjects (eight sub-subjects), dated from March 15, 2005 to March 21, 2005, are collected with statistics shown in Table 2.

4.1.2. Performance measure

To evaluate the performance of the proposed classifier, we employ the measures that are widely used in email classification. A confusion matrix can be formed according to the matching situations between the predicted classes by the classifier and their actual classes (as shown in Table 3).

In this matrix, A represents the number that the classifier correctly classifies the legitimate mails; D is the number that the classifier correctly rejects the spam emails. B is the number that the classifier incorrectly classifies spam emails as legitimate ones, while C is the number that the classifier incorrectly rejects legitimate emails as spam. The common evaluation measures include Precision, Recall, F-measure, and Accuracy. Their corresponding definitions are as follows:

$$\text{Positive Precision} = \frac{A}{A+B} \tag{1}$$

$$\text{Negative Precision} = \frac{D}{C+D} \tag{2}$$

$$\text{Positive Recall} = \frac{A}{A+C} \tag{3}$$

$$\text{Negative Recall} = \frac{D}{B+D} \tag{4}$$

$$F\text{-measure} = \frac{2pr}{p+r} \tag{5}$$

$$\text{Accuracy} = \frac{A+D}{A+B+C+D} \tag{6}$$

Positive precision represents the percentage of real legitimate in the predicted legitimate emails. Negative precision represents the percentage of real spam in the predicted spam. Positive recall is defined as the percentage of the true legitimate emails that are correctly predicted by the classifier. Negative recall is defined as the percentage of the true spam emails that are correctly predicted by the classifier. F-measure is the harmonic average of (positive or negative) precision and recall. Accuracy is the percentage of all emails that are correctly classified by the classifier.

4.1.3. Benchmark comparison

In the experiments, we employ the performance of the k nearest neighbors (KNN) approach (adopted in ECUE, Delany & Cunningham, 2006) as the benchmark for comparison. KNN is a typical kind of instance-based learning methods, i.e., it simply performs feature selection task in the training phase without really training a classification model. When testing, it employs distance (dissimilarity) or similarity measures to find k instances in the training set that are most similar to the test data, and then aggregate the results of these k instances to be the prediction result for the test data.

With no explicit knowledge structure during training, KNN is insignificantly influenced by the class-skewed problem. On the other hand, KNN does not have incremental learning mechanism. Whenever there are new coming instances, it simply stores them as the training data and re-perform feature selection task. When testing, it performs similarity matching of the test data with the entire training data. The computational cost of re-training, i.e., repetitive feature selection tasks, is expensive though.

4.2. Experiment I

In Experiment I, under the condition of no significant skewed class distribution and of no concept drift, we examine ICBC's performance using the whole collected data from Spamassassin and E.M. Canada datasets. We randomly select 20% of the 8648 Spamassassin dataset and 20% of the 15,213 E.M. Canada dataset as the training set. The remaining 80% of each dataset is served as test data. The result is shown in Table 4.

From Table 4, we first observe that both ICBC and KNN perform well under the large number of data

[1] Spamassassin email dataset: http://spamassassin.apache.org/publiccorpus.

[2] E.M. Canada spam email dataset: http://www.em.ca/~bruceg/spam.

Table 1
Objectives of the experiments

	Objectives
Experiment I	To examine the performance of ICBC under the condition of no significant class-skewed problem and of no concept drift
Experiment II	To demonstrate the performance of ICBC under the situation of skewed class distributions
Experiment III	To justify the incremental learning capability of ICBC under the situation of a user's preference changes

Table 2
Statistics of collected data in Experiment III

Subject	Sub-subject	Number	Subject	Sub-subject	Number
Sport	Basketball	50	Entertainment	Movie	60
	Football	20	Health	Weight loss	20
Business	Economy	50	Technology	Macintosh	20
	Stock market	60	Weather	Weather	20

Table 3
A confusion matrix in email classification

		Actual	
		Legitimate	Spam
Predicted by the classifier	Legitimate	A	B
	Spam	C	D

Table 4
The classification result in Experiment I

		Precision	Recall	*F*-Measure	Accuracy
ICBC	Legitimate	0.93177	0.92302	0.92738	0.94730
	Spam	0.95762	0.96100	0.95931	
KNN	Legitimate	0.97538	0.92461	0.94931	0.96416
	Spam	0.95927	0.98657	0.97273	

employed. Furthermore, it shows that ICBC performs slightly worse than KNN. The reason is that ICBC calculates distances by comparing test data with the feature vector of each cluster, while KNN compares test data with all training data. KNN, therefore, results in more accurate performance by enumerative comparison.

Nevertheless, KNN suffers from its slowness in enumerative comparison. The time for classifying a new instance grows as the number of instances that KNN stores increases. To illustrate, we perform a simple experiment that increases the number of training instances from 10 to 100, with an increment of 10, and fixes the number of testing instances to 200, and compute the computational time for ICBC with that for KNN. The result is shown in Fig. 7.

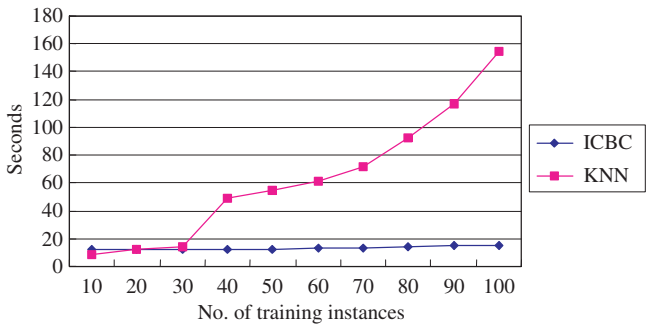

Fig. 7. The comparison of computational time between ICBC and KNN.

From Fig. 7 it is obvious that the time for KNN grows as the number of training instances increases. On the contrary, the time for ICBC seems to be steadily constant. ICBC largely decreases the influence of the number of training instances on the prediction speed.

4.3. Experiment II

Experiment II is to examine ICBC on dealing with the issue of skewed class distributions. Traditional classifiers tend to prefer the majority classes and result in poor recall rates for minority classes. This problem may not even be salient if one employs the overall prediction accuracy as the performance measure, since the errors resulted from the minority classes would not have major impact on the overall prediction accuracy. To better account for the class-skewed problem, we employ the positive recall as the evaluation criterion in this experiment.

To deliberately make significant skewed class distributions, we select five training data sets, each of which consists of 200 spam and 10, 20, 30, 40, or 50 legitimate emails. The test set consisting of another 50 legitimate emails is used to evaluate the performance under each situation.

The result is shown in Fig. 8. Overall, the positive recall rates for both ICBC and KNN deteriorate as the skewed class distribution phenomena become more significant. More importantly, ICBC outperforms KNN on all of the five cases. Originally, KNN should not be influenced by the problem of skewed class distributions. However, due to the global feature selection mechanism in text categorization, KNN still cannot avoid the feature selection problem in minority classes. In contrast, ICBC selects equal number of features within clusters. This result illustrates

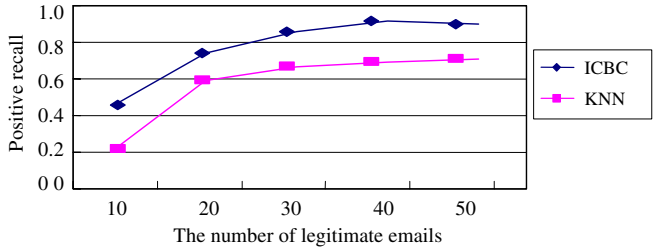

Fig. 8. The result of classification under skewed class distributions.

the effectiveness of the clustering strategy of ICBC on dealing with the issue of skewed class distributions.

4.4. Experiment III

The objective of Experiment III is to evaluate the incremental learning capability of ICBC. Therefore, we deliberately create a scenario with significant changes of a user's preference, a typical example of concept drift. The scenario is as follows. Suppose that an online user subscribes the Yahoo newsgroups and receives electronic news accordingly. At first, four subjects of news (Weather, Football, Weight Loss, and Macintosh) are sent to the user who considers emails of the former two subjects as legitimate and the latter two as spam. An initial classifier is trained according to the user's original preferences. Gradually, another two subjects of news (Movie and Stock Market) are also sent to the user who begins to read the emails of Movie subject (legitimate) but still ignore the emails of Stock Market subject (spam). In this case, the classifier should be able to adapt itself to the new situation and make correct prediction. Abiding by this description, the data in this experiment are manipulated as shown in Table 5. The procedure to conduct this experiment is further explained as follows:

(1) The initial classifier is built by a training data set that consists of 20 emails of Weather subject and 20 emails of Football subject as legitimate emails, and 20 emails of Weight Loss subject and 20 emails of Macintosh subject as spam.

Table 5
Statistics of data manipulated in Experiment III

Class		Subject	Number of emails
Legitimate		Weather	20
		Football	20
Spam		Weight loss	20
		Macintosh	20
Incremental learning test data	Legitimate	Movie	30
	Spam	Stock market	30
Incremental learning validation data	Legitimate	Movie	30
	Spam	Stock market	30

Table 6
Recall rates from ICBC and KNN under incremental learning

Size of test set	ICBC			KNN		
	Movie	Stock	*F*-Measure	Movie	Stock	*F*-Measure
0[a]	0.64000	0.74286	0.69143	0.46154	0.74074	0.60114
5	0.92857	0.93750	0.93304	0.70833	0.93548	0.82191
10	0.96774	0.96552	0.96663	0.93750	0.92857	0.93304
15	0.96774	0.96552	0.96663	0.92308	0.90909	0.91608
20	0.96774	0.96552	0.96663	0.85714	0.80000	0.82857
25	0.96774	0.96552	0.96663	0.88235	0.84615	0.86425
30	0.95082	0.94915	0.94999	0.85714	0.80000	0.82857

[a] Represents the initial framework without incremental learning.

(2) Movie and Stock are selected as the two targeted subjects emerging gradually, with emails of Movie subject as the new legitimate emails and emails of Stock subject as the new spam.
(3) To examine the effects of the incremental learning, we apply 30 emails of Movie subject and 30 emails of Stock subject as the incremental learning test data, in an increment of 5 emails fed into the classifier each time. We hold the rest 30 emails of Movie subject and the rest 30 emails of Stock subject as the validation data to evaluate the adapted classifier.

The incremental learning results (recall rates) of ICBC and KNN are shown in Table 6. We observe that ICBC does not detect the concept drift at first because of few test data inputs. Nonetheless, it quickly adapts itself to the concept drift with more test data, and improves the recall rate accordingly. On the contrary, KNN may learn to adapt to the new situations, but in a much slow and unstable manner.

We further plot the *F*-measure results from ICBC and KNN in Fig. 9 for visual comparison. It is clear that our ICBC outperforms KNN in the incremental learning process. In addition, KNN uses all the available data to re-perform the feature selection task each time, while ICBC performs incremental learning and selects features within those varied clusters solely, and thus reduce the cost of re-training.

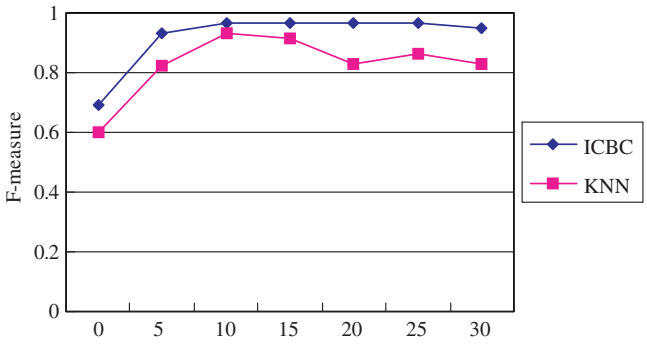

Fig. 9. The *F*-measure comparison between ICBC and KNN.

5. Concluding remarks

With the prevailing of Internet, email has become an indispensable part of our daily life for communication. However, because of the prevalence of spam emails, how to detect and filter spam become a quite essential issue. In this study, we propose an incremental cluster-based classification approach, ICBC, to improving the effectiveness of spam filtering.

ICBC is a classification method based on cluster analysis. It first clusters concepts into several sub-concepts, and extract equal-sized representative features (keywords) from each sub-concept. With this procedure, concepts in minority classes can have equal-sized features as those in majority classes, which lessens the class-skewed problem. At the same time, the cluster structure of ICBC can be adaptively changed, which imposes ICBC the capability of incremental learning. ICBC can accommodate changes in a quick and low-cost manner, and relieve the problem of concept drift.

Three experiments are conducted to investigate the performance of ICBC. The result of Experiment I shows that, when the problem of skewed class distributions is not critical, the performance of ICBC is comparable to that of KNN but its computational time is far less than that of KNN. In Experiment II, ICBC outperforms KNN when the problem of skewed classes is critical. The result shows that ICBC can deal with the issue of skewed class distributions effectively. In addition, the result of Experiment III shows that the incremental learning of ICBC can accommodate the concept drift, and thus reduce the prediction error. On the contrary, KNN, without incremental learning, has slow-adapting and unstable performance and high re-training cost.

To continue our research, we consider several potential future works. First, the initial cluster structure of ICBC has critical impact on the classification accuracy of its later modifications. A future research work is to consider approaches that can improve the initial cluster structure. WordNet or Ontology, for example, may be used to enforce the links between concepts and result in a more accurate initial clustering. Second, ICBC solely relies on the content analysis of emails without considering other useful information. A hybrid approach that includes content analysis and useful email information (e.g. headers) may improve the effectiveness in spam filtering. Finally, it is worth implementing ICBC as a spam filtering system applicable in the real world.

References

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, D. (2000). Learning to filter spam e-mail: a comparison of a naive bayesian and a memory-based approach. The 4th PKDD's workshop on machine learning and textual information access.

Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems, 14*(6), 67–74.

Delany, S. J., & Cunningham, P. (2006). ECUE: a spam filter that uses machine learning to track concept drift. Technical Report, Computer Science Department, The University of Dublin. Available from https://www.cs.tcd.ie/publications/tech-reports/reports.06/TCD-CS-2006-05.pdf.

Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks, 10*(5), 1048–1054.

Fawcett, T. (2003). In vivo spam filtering: a challenge problem for KDD. *ACM SIGKDD Explorations Newsletter, 5*(2), 140–148.

Forman, G. (2006). Tackling concept drift by temporal inductive transfer. *SIGIR '06 ACM*.

Hart, P. E. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory, IT-14*, 515–516.

Honda, T., Motizuki, H., Ho, T. B., & Okumura, M. (1997). Generating decision trees from an unbalanced data set. In Maarten vanSomeren., Gerhard Widmer (Eds.), *Poster papers presented at the 9th European conference on machine learning (ECML)* (pp. 68–77).

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys, 31*(3), 264–323.

Lee, C., & Lee, G. G. (2006). Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing and Management, 42*(1), 155–165.

Lian, Y. (2002). E-mail filtering, Master thesis. University of Sheffield, Department of Advanced Software Engineering.

Luo, X., & Zincir-Heywood, N. (2005). Comparison of a SOM based sequence analysis system and naive bayesian classifier for spam filtering. In *Proceedings of IEEE international joint conference on neural networks – IJCNN '05* (Vol. 4, pp. 2571 – 2576).

McCallum, A. (2002). Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering. Available from http://www.cs.cmu.edu/~mccallum/bow/.

Monard, M. C., & Batista, G. E. A. P. A. (2002). Learning with skewed class distributions. In *Advances in logic, artificial intelligence and robotics (LAPTEC'02)*.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*, 103–134.

Payne, T. R., & Edward, P. (1997). Interface agents that learn: an investigation of learning issues in a mail agent interface. *Applied Artificial Intelligence*, 1–32.

Porter, M. (1980). An algorithm for suffix stripping. Program. *Automated Library and Information Systems, 4*(3), 130–137.

Sahami, M., Dumais, S., Heckerman D., & Horvitz, E., (1998). A Bayesian approach to filtering junk e-mail. AAAI-98 workshop on learning for text categorization.

Stanley, K. O. (2003). Learning concept drift with a committee of decision trees. Department of Computer Sciences Technical Report AI-03-302.

Wu, H., Pang, T. H., Liu, B., & Li., X. (2002). A refinement approach to handling model misfit in text categorization. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 23–26).

Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval, 4*(1), 5–31.