Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 497354, 11 pages
http://dx.doi.org/10.1155/2014/497354

Research Article

A Dynamic Ensemble Framework for Mining Textual Streams with Class Imbalance

Ge Song and Yunming Ye

Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China

Correspondence should be addressed to Ge Song; carroll0708@qq.com

Received 6 December 2013; Accepted 19 February 2014; Published 10 April 2014

Academic Editors: Z. Chen and F. Yu

Textual stream classification has become a realistic and challenging issue since large-scale, high-dimensional, and non-stationary streams with class imbalance have been widely used in various real-life applications. According to the characters of textual streams, it is technically difficult to deal with the classification of textual stream, especially in imbalanced environment. In this paper, we propose a new ensemble framework, clustering forest, for learning from the textual imbalanced stream with concept drift (CFIM). The CFIM is based on ensemble learning by integrating a set of clustering trees (CTs). An adaptive selection method, which flexibly chooses the useful CTs by the property of the stream, is presented in CFIM. In particular, to deal with the problem of class imbalance, we collect and reuse both rare-class instances and misclassified instances from the historical chunks. Compared to most existing approaches, it is worth pointing out that our approach assumes that both majority class and rareclass may suffer from concept drift. Thus the distribution of resampled instances is similar to the current concept. The effectiveness of CFIM is examined in five real-world textual streams under an imbalanced nonstationary environment. Experimental results demonstrate that CFIM achieves better performance than four state-of-the-art ensemble models.

1. Introduction

Delivering emails and forums, providing chatting rooms, and publishing blogs are making textual streams dynamic and large scale. It is a big challenge for the textual stream classification because of many interesting characteristics of a textual stream. First, a textual stream with large-scale data instances is of high-dimensional and imbalanced distribution. High-dimensional feature distribution is a persistent problem in general data classification contexts. In addition, skewed class distributions can be seen in many data stream applications, such as credit card transactions and intrusion detection. In these cases, the instances in some classes (rare classes), which gain more attention and are of more importance in most of the real-world application, are much more difficult to be gathered than the other classes. In traditional existing classification models, imbalanced distribution may cause performance degradation since building these models mainly relies on instances in majority classes, ignoring the importance of the rare class. Therefore, most existing data stream classification methods fail to tackle the textual stream due to both its high-dimensional feature space [1] and imbalanced class distribution. Second, concept drift may occur in the period of time. This property requires the current classifier to fit up-to-date concepts. For instance, the customers' buying preferences and patterns may change with time. This drives us to develop an adaptive model to meet the current buying preferences and patterns of customers.

Ensemble methods, which are regarded as promising methods to deal with textual streams, aim at integrating several individual submodels to form a final prediction [2]. It is easy to obtain training set for each submodel since a stream can naturally be divided into several individual chunks. In this paper, we propose a new ensemble model, clustering forest, to classify textual stream with imbalanced and drifting stream (CFIM). CFIM aims at integrating a number of clustering trees (CTs) by an efficient and adaptive method. It is worth noting that CT, a clustering-based classification algorithm [3], is suitable for large high-dimensional sparse textual data [3]. Besides the current chunk for training the submodels, we collect another two new subsets, rare-class subset and misclassified subset, to tackle the imbalanced

problem. These two new subsets are aggregated from the historical chunks with the same feature distribution. It is worth pointing out that compared with other existing imbalanced stream classification approaches, which assume that the distribution of rare class is not changed [4], the assumption of our model is that all the instances in the stream may suffer from concept drift and change their feature distribution. A suitable number of CTs are combined automatically to accomplish the optimal prediction result. During the ensemble process, we propose an adaptive selection method. A dynamic threshold is used to measure whether the CT fits a new concept. The threshold is defined according to the accuracy weight of a CT rather than using the random prediction accuracy by most of the existing models [5, 6]. The performance of CFIM is experimented on several textual stream datasets on Massive Online Analysis (MOA) platform [7]. Compared with other four state-of-art ensemble approaches, experimental results show that CFIM delivers the promising performance regarding the average accuracy, the plotting accuracy, and the kappa statistic.

The previous works related to our study are reviewed in Section 2; we then present the overall framework of CFIM in Section 3. In Section 4, we propose the sampling method to form training set. The adaptive selection method and voting method are proposed in Section 5. Experimental results are analyzed in Section 6. In Section 7, we summarize the whole paper and give suggestions on future work.

2. Related Work

2.1. Textual Stream with Class Imbalance. In various real-life applications, textual stream is characterized by having the high-dimensional feature space, a large number of instances with imbalanced distribution and the concept drift. Many researches of ensemble learning focus on textual stream classification with concept drift [6, 8–11]. For example, the Conceptual Clustering and Prediction (CCP) [11] framework has been proposed to classify the textual stream with recurring concept. Incremental subclassifiers in CCP were constructed and updated when new training instances arrived. A new subclassifier was built when a new concept was detected. Thus one concept corresponded to only one incremental subclassifier. In this sense, it was not an ensemble approach in the period of one concept. Moreover, a semisupervised approach, PU Learning by Extracting Likely Positive and Negative Microclusters (LELC) [9], aimed at dealing with positive and unlabeled textual stream problems. The extension research of LELC, Voted LELC [12], was then presented. This method allocated a voting weight to each representative instance. The voting weight represented the belongingness of an instance towards its corresponding class. Then Voted-LELC combined all SVM-based subclassifiers (built by voted representative instances) into global prediction result using the AWE ensemble model. To overcome some drawbacks of AWE, this method applied active learning to classify unknown textual instances. Similar approach to one-class classification of textual streams was proposed by Zhang et al. [10].

Besides the concept drift, many researches aimed at overcoming another issue, class imbalance. A framework [13, 14] based on collecting rare-class examples has been proposed. All the instances in the rare class should be gathered into the training chunk, while the instances belonging to the majority class were sampled into the training set according to a distribution radio. This approach implicitly assumed that the distribution of rare class is not drifting over time [4]. Likewise, Chen and He [15] have proposed SERA algorithm, which selected the "best" rare-class instances according to the Mahalanobis distance, instead of using all old instances in some algorithms, such as [13, 14]. However, the algorithm may not be able to track drift in rare-class instances. Another research based on [13, 14] has been proposed by Lichtenwalter and Chawla [16]. In their paper, besides collecting rare-class instances, they also collected the misclassified majority class instances. In addition, Learn++.NIE (for learning in nonstationary and imbalanced environments) [4, 17, 18] has been proposed to deal with class imbalance. Compared with the original Learn++.NSE algorithm, a step using the bagging algorithm was added to handle the imbalance problem.

2.2. Features of Our Approach. Our new ensemble approach, CFIM is proposed to handle textual streams with class imbalance and concept drift. CFIM is different with other traditional ensemble models from the following aspects.

(i) According to the adaptive selection method, CFIM changes the number of selected submodels along with the variation of the up-to-date concept. It is more flexible to select suitable submodels in comparison to many existing methods with the fixed number of selecting the submodels.

(ii) We accumulate rare-class instances and misclassified instances from the historical chunks with the same feature distribution to better define the boundary between the classes. It is worth mentioning that our CFIM assumes that both majority class instances and rare-class instances may suffer from concept drift. Therefore, the instances in rare-class subset and misclassified subset should be sampled from the historical chunks, whose distributions are as similar as the latest chunk. Moreover, based on the sample method in our CFIM, the majority class instances should be sampled at a proper radio to form balanced training set.

3. Clustering Forest for Imbalanced and Drifting Stream (CFIM)

In this paper, we propose a new ensemble model, CFIM, for imbalanced and drifting textual stream. The framework of our model is shown in Figure 1. CFIM includes four steps:

(i) Resample step, which is to form the training chunk by combining the new chunk with misclassified subset E and rare-class subset R according to sample method.

(ii) Training step, which is to train the submodels, CTs.

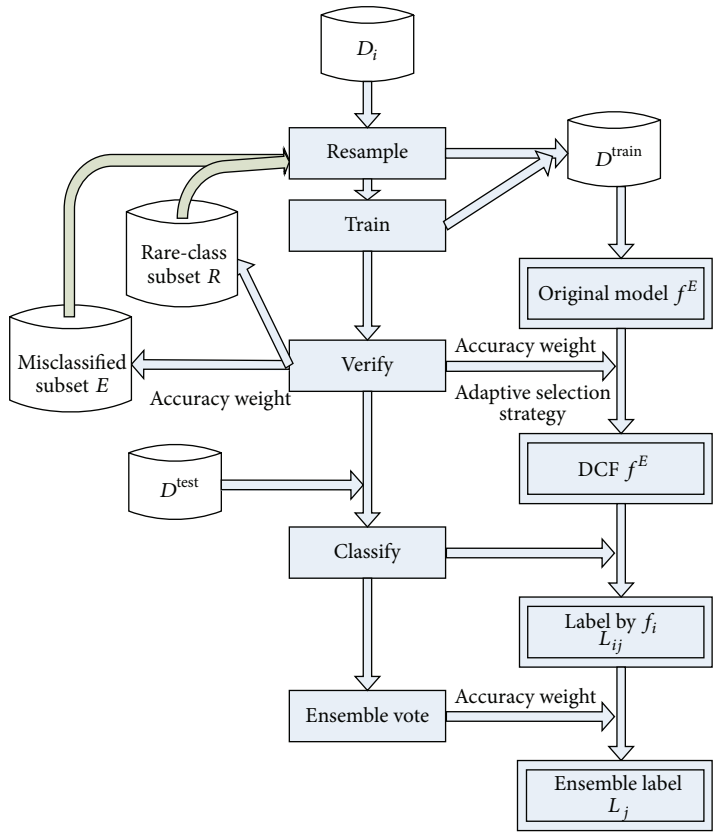

Figure 1: Framework of clustering forest (CFIM).

(iii) Verifying step, which is to adaptively select the CTs and form the misclassified subset E and rare-class subset R.

(iv) Testing step, which is to predict the labels of the incoming testing instances.

The verifying step consists of three substeps.

(1) Compute the accuracy weight of each CT.

(2) Adaptively select the CTs to integrate the optimal CF.

(3) Form the misclassified subset E and rare-class subset R by accuracy weight.

The testing step consists of two substeps.

(1) Predict the labels of incoming testing instances by each CT in CFIM.

(2) Integrate the CTs and obtain the global prediction according to accuracy weight.

We would like to clarify the framework of CFIM as follows (the details of CFIM are shown in Algorithm 1).

(i) Form the training set. The training set consists of new training chunk D_t, the misclassified subset E_t, and the rare-class subset R_t. We would like to define the radio p^R which is the size of rare-class subset as a proportion of the training chunk D_t^{train}, $p^R = |R|/|D_t^{\text{train}}|$. The threshold p_θ^R is introduced to control the number of majority class instances. At the tth time stamp, if $p_t^R < p_\theta^R$, that means the number of rare class is too small to influence the construction of submodel. So all the instances in rare class should be added in the training set at this time stamp. Meanwhile, we should undersample the incoming chunk D_t and select several $((1 - p_\theta^R)|D_t|)$ majority class instances to balance the number of rare class and majority class.

(ii) Build CTs to integrate the original clustering forest (CF). Since the submodels have time limited effectiveness, $M_{\max}$ should be defined to ensure the time efficacy and to control the scale of our ensemble model. At the tth time stamp, a new CT should be built with the training set based on Step 1. Meanwhile, the "old enough" CTs should not be discarded.

(iii) Adaptively select the CTs. An accuracy weight of each CT is computed. We then select the CTs based on an adaptive selection method to achieve the optimal CF.

(iv) Constitute the misclassified subset E and rare-class subset R. According to accuracy weight, we select the misclassified instances and rare-class instances from

Output: Class label of the testing instance x
Input: D_t: The data chunk at each time stamp
M: The number of CTs in CF$\left|f^E\right|$
$M_{\max}$: The maximum number of CTs selected in the clustering forest $\left|f^E\right|_{\max}$
f^E: The current ensemble model
E_t: Misclassified subset at the t-th time stamp
R_t: Rare-class subset at the t-th time stamp.
P_t^R: The Rare-class subset as a proportion of the training chunk; p_θ^R: The threshold of p^R
(1) at the t-th time stamp
(2) built the training set by Algorithm 2
(2) create a new CT f_t using D_t^{train}
(3) if $t < M_{\max}$
(4) $f^E \leftarrow f^E \cup f_t$
(5) Endif
(6) if $t > M_{\max}$
(7) $f^E \leftarrow f_{t-M_{\max}+1} \cup \cdots \cup f_t$
(8) Endif
(9) compute the accuracy weight of each clustering tree
(10) UPDATE(f^E) based on the adaptive selection method
(11) obtain the misclassified subset E_t and the rare-class subset R_t
(12) for each testing instance x do:
(13) PREDICT(x) by the voting method.
(14) Endfor.

Algorithm 1: Clustering forest.

the old chunks, whose distributions are similar to the testing chunk.

(v) Classify incoming instances by each CT.

(vi) Vote and obtain the ensemble prediction label. For each testing instance x, we obtain the ensemble prediction result by using a voting method, which is based on the accuracy weight.

Our proposed CFIM contains the following key methods.

(i) Sample method.

We should collect the misclassified instances and rare-class instances from the "old" chunks which have the similar distribution as the current chunk. According to accuracy weight, we choose the "old" but "useful" chunks to help the training. Misclassified instances are used to reinforce learning of the submodel and construct the strong classifiers. Rare-class instances are used to balance the size of majority class and the rare class. If the size of the rare-class set is too small, we should resample the instants in current chunk to reduce the scale of majority class. P_θ^R is defined as the threshold to ensure the rare-class instances are the proportion of the current chunk. It is especially interesting in our sample method that we choose the chunks with the same distribution as the current chunk to form the above two sets according to accuracy weight in the concept drift environment. Therefore, this method makes full use of the "old" chunks but discards the "useless" chunks.

(ii) Adaptive selection method.

A new accuracy weight for selection of CTs has been defined to dynamically select "good enough" experts (i.e., CTs) into the optimal CF. Unlike most existing models (e.g., AWE and AUE) where the number of submodels is fixed, the number of CTs in our strategy is varied according to the changes of concepts (see Section 5.2).

(iii) Voting method.

According to the accuracy weight, we utilize the historical information of the training data to ensure the ensemble prediction capabilities of the submodels.

4. Sampling Method

In this section, we propose the sample method of our CFIM. According to sample method, we combine the instances from misclassified instances E_t, rare-class instances R_t, and the current chunk D_t into the training chunk D_t^{train} at each time stamp. The key procedures of sampling method are to build the misclassified set and rare-class set. The detailed information is shown in Algorithm 2 and Figure 2.

4.1. Misclassified Set. To obtain the higher accuracy from the CFIM, we should incrementally emphasize distinguishable ability of the submodels. The instances that previous models misclassified contain more information to enhance the classification accuracy than the accurately classified instances. Therefore, we collect all the instances which are given the false labels into the misclassified set E_t and add E_t into

Output: The optimal CF
Input: D_t: The data chunk at each time stamp
M: The number of CTs
L: the prediction of $\mathbf{x}$
E_t: Misclassified subset at the t-th time stamp
R_t: Rare-class subset at the t-th time stamp.
y_R: The label of rare-class
(1) At the t-th time stamp:
(2) Obtain the f^E by Algorithm 1.
(3) For all $x_{kj} \in D_k$, where D_k is the training chunk of f_k $\left(f_k \in f^E\right)$;
(4) Obtain the label L_{kj} by f^E
(5) If $L_{kj} \neq y_{kj}$:
(6) $E_t \leftarrow E_t \cup x_{kj}$;
(7) Endif
(8) If $y_{kj} \in y_R$
(9) $R_t \leftarrow R_t \cup x_{kj}$;
(10) Endif.
(11) Endfor.
(12) if $p_t^R < p_\theta^R$:
(13) $D_t \leftarrow \text{RESAMPLE}(D_t)$ at the percentage $(1 - p_\theta^R)$
(14) $D_t^{\text{train}} \leftarrow D_t \cup E_t \cup R_t$;
(15) Else
(16) $D_t^{\text{train}} \leftarrow D_t \cup E_t \cup R_t$;
(17) Endif.
(18) Create a new CT f_t using D_t^{train} relying on Algorithm 1;
(19) UPDATE(f^E)

Algorithm 2: Sample method.

the current training chunk D_t^{train}. It is worth pointing out that according to accuracy weight in CFIM, all the misclassified instances in E_t are considered to have the similar distribution as the current chunk. If a historical submodel obtains higher accuracy weight than the threshold of accuracy weight ($\omega_i > \omega_\theta$), its corresponding chunk D_i is considered to have the similar distribution as the current chunk. So we accumulate the instances that are misclassified by f^E in data chunk D_i. According to E_t, most informative data should be gathered to build the new submodel. Meanwhile, the data related to "old" concept should be discarded.

4.2. Rare-Class Set. The number of rare-class instances in the most recent chunk is far from sufficient to train a model with high accuracy in the imbalanced stream environment. The traditional model, ignoring the imbalanced problem, would perform poorly on the rare class. All rare-class instances in the previous chunks associating with the current concept should be accumulated into the training set. In particular, if the proportion of the rare-class instances in the training is too small to build the high-accuracy submodel ($p_t^R < p_\theta^R$), we should undersample the majority class instances from the current chunk D_t to balance the class distribution. Figures 2(a) and 2(b) show the process of forming the training set at the recent time stamp under two situations ($p_t^R > p_\theta^R$ and $p_t^R < p_\theta^R$), respectively.

5. Adaptive Selection Method and Voting Method

In this section, we outline another two important methods of our CFIM framework: an adaptive selection method and voting method.

5.1. Adaptive Selection Method. Adaptive selection method in our framework seeks to select suitable CTs to organize the optimal CFIM. The CTs related to the current concept are considered as the useful CTs to predict the testing instances. The number of CTs increases when the concept is not changed. However, if the concept drift is detected, most of the old CTs do not represent the new concept, so they are not useful to classify the new testing instances. Only the "useful" CTs can participate in classification.

Algorithm 3 illustrates the adaptive selection method in CFIM. We estimate the accuracy weight ω_i of all the CTs by the new chunk. We then use all the selected CTs ($\omega_i > \omega_\theta$) to classify the testing instance. More remarkably, our method is based on the assumption that the latest CT always participates in prediction.

The following equation is used to compute the accuracy weight:

$$\omega_i = \frac{\varphi_i}{\sum_{i=1}^{M} (\varphi_i)}, \tag{1}$$

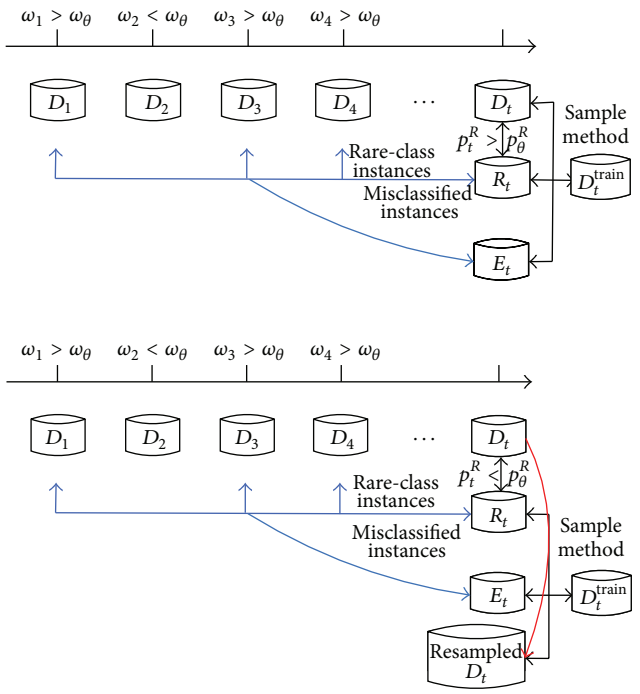

Figure 2: Process of forming the training set.

where M is the total number of CTs, φ_i is the accuracy of each CT. We define the threshold ω_θ which is used to decide whether each CT is suitable or not. The threshold ω_θ is given by

$$\omega_\theta = \begin{cases} \max\left(\min \varphi_i - \tau, \frac{1}{2}\right) & \overline{\varphi} - \min \varphi_i \le \varepsilon \\ \max\left(\overline{\varphi}, \frac{1}{2}\right) & \overline{\varphi} - \min \varphi_i > \varepsilon. \end{cases} \quad (2)$$

Here, the concept drift is detected by the difference between the average accuracy $\overline{\varphi}$ and the minimum accuracy $\min \varphi_i$ ($\overline{\varphi} - \min \varphi$) in the original CF. If concept drift does not occur (i.e., $\overline{\varphi} - \min \varphi_i \le \varepsilon$), each CT takes part in classifying the incoming instances. The parameter τ is the minimum value to ensure that all selected CTs can participate in the ensemble model. Otherwise, if the concept drift is detected (i.e., $\overline{\varphi} - \min \varphi_i > \varepsilon$), only some "useful" trees with their accuracies higher than $\max(\overline{\varphi}, 1/2)$ can participate in classification.

5.2. Voting Method. We may construct a voting method related to the accuracy weight to integrate results of subclassifiers into an ensemble result. We may use the accuracy weight for each CT as the final voting weight. That is,

$$v_i\left(\mathbf{x}_j\right) = \omega_i, \quad (3)$$

where ω_i is the accuracy weight of each CT.

The ensemble prediction label L_j of the testing instants $\mathbf{x}_j$ can be set to the maximum value of ensemble function $f^E(\mathbf{x}_j)$.

6. Experiments

6.1. Datasets. At present, lack of benchmark datasets for textual stream with imbalance class is the problem of classification performance evaluation. In fact, as collecting large-scale real-world textual streams needs to consume huge time and manpower to manually label a mass of instances, it is hard to construct such textual streams. Thus, in order to evaluate the performance of CFIM, we have compiled five imbalanced stream datasets with concept drift. The detailed description is seen in Table 1.

The Spam Assassin Collection (Spam for short) [11] is used as the real-world textual stream. This collection consists of 9,324 instances in Spam class and ham class, and each instance contains 500 attributes. The imbalanced radio of this corpus is 3 : 1. All the instances are arranged in the arrival order. The previous studies [11] show that the attributes of emails gradually change over time.

We construct a new large-scale textual stream (called Spam-Enron stream for short) by the Spam Assassin Collection and Enron Email Dataset with 33,702 instances. This dataset is chronologically collected with 17,157 instances belonging to the Spam class, while other 16,545 ones belonging to the ham class. We use the sigmoid function to formulate a weighted integration of two real-world streams in order to characterize the target concepts in a concept drift environment [7]. Based on this process, we construct the Spam-Enron stream with 100,000 instances, each of which contains 2,044 attributes.

The third stream is collected by the Spam Assassin Collection and a benchmark dataset, Reuters. We sample the instances which belong to the first class and the seventh class in Reuters in order to construct the imbalanced stream (1 : 10). According to the sigmoid function, we create a new stream (called Reuters-Spam stream) with 10,000 instances, each of which contains 19,433 attributes.

Table 1 shows that the Spam Assassin stream and Spam-Enron stream are two imbalanced streams where the proportion of two classes is approximately 3 : 1. We add another imbalanced stream with 1,187 instances in one class and 6,937 instances in the other one to evaluate whether CFIM is suitable for the imbalanced stream. This stream (Spam1 for short) is generated from the Spam Assassin stream by sampling a certain number of instances in the Spam class. The ratio of the number of instances in two classes is approximate to 6 : 1 in the imbalanced Spam Assassin stream.

Similarly, another imbalanced textual stream generated from Spam Assassin stream is collected. This textual stream (Spam2 for short) contains 6,920 instances in one class and 280 instances in another one. The ratio of the number of instances in two classes is approximated to 25 : 1.

6.2. Evaluation Procedure. The evaluation procedure is used to arrange the training instances and the testing instances. The basic assumption is that the testing chunk is basically similar to the most recent chunk in the stable period, while the distribution of the testing chunk is changed when a concept drifts. Interleaved chunk method [19] is considered as the evaluation procedure in our paper.

Output: The optimal CF
Input: D_t: The data chunk at each time stamp
M: The number of CTs in CF
(1) At the t-th time stamp:
(2) Obtain current original clustering forest f^E relying on Algorithm 1;
(3) For $f_i \in f^E$ do:
(4) Estimate the accuracy of f_i using D_t;
(5) Compute the accuracy weight ω_i of f_i with (1);
(6) If $\omega_i > \omega_\theta$:
(7) REMOVE(f_i);
UPDATE(f^E);
(8) Endif

Algorithm 3: Adaptive selection method.

Table 1: The properties of textual streams.

	The number of instances			Imbalanced radio	Size of attribute	Size of each chunk
	Spam	Legitimate	Total			
Spam Assassin	2,387	6,937	9,324	1 : 3	500	300
Enron Email	17,157	16,545	33,702	1 : 1	1,545	—
Spam-Enron	25,000	75,000	100,000	1 : 3	2,044	500
Spam1	1,187	6,937	8,100	1 : 6	500	300
Spam2	280	6,920	7,200	1 : 25	500	300
Reuters-Spam	956	9,044	10,000	1 : 10	19,433	500

Interleaved chunk combines Holdout and Interleaved Test-Then-Train evaluation methods [19]. The procedure is as follows.

(i) Collect incoming samples to form a data chunk.

(ii) Test this data chunk to evaluate the latest model.

(iii) Train this data chunk again to update the classification model.

The Spam Assassin stream, Spam1 stream, and Spam2 stream use the interleaved chunk procedure by collecting 300 instances as a textual chunk for one time stamp in a proper sequence. Therefore, Spam Assassin stream, Spam1 stream, and Spam2 stream include 30 time stamps, 26 time stamps, and 23 time stamps, respectively. Similarly, Spam-Enron stream and Reuters-Spam stream, which collect 500 instances as a chunk, include 200 time stamps and 19 time stamps, respectively.

6.3. Evaluation Measures. Two important evaluation measures regarding accuracy are used in our paper. To evaluate the overall performance of a learning algorithm, we employ average accuracy. To represent the "accuracy" at each time stamp in an evolving environment, the plotting accuracy is of great importance. The calculations of the average accuracy and the plotting accuracy can be found in [20].

However, the average accuracy and plotting accuracy are not suitable for evaluating the performance of classifiers for an imbalanced data set. The number of instances in the small classes is too small to apparently have effect on the percentage of accuracy. An alternative measure to evaluate the classification error in an imbalanced environment is the kappa statistic [21], which is used to compute the agreement between observed proportion and the expected proportion. The detailed calculation method for kappa statistic is shown in [21].

6.4. Compared Models. For comparing the performance, our model has been compared with four state-of-the-art ensemble models, which are much related to our work. These models are Accuracy Weighted Ensemble (AWE) [5], Accuracy Updated Ensemble (AUE) [22], Leveraging Bag (LB) [23], and OzaBagAdwin (OZA) [2]. AWE as a popular method uses the weighted ensemble classifier; AUE extends AWE by using incremental submodels and updating them according to the prediction accuracy; LB combines the Adwin algorithm and the Leveraging Bagging algorithms by using Random Output Codes (ROC). OZA detects and estimates the change by the Adwin algorithm for providing the ensemble weights. In our experiment, we choose Random Tree (RT), Random Forest (RF), libSVM (SVM for short) as basic learners. Since all these three basic learners are used in AUE, AWE, LB, and OZA, respectively, 12 comparative methods take part in the experiments. All these tested algorithms are implemented in the Massive Online Analysis (MOA) platform [7]. We assign the cost function of SVM $c = 1$. The maximum number of subclassifiers in all tested ensemble models to $M_{\max} = 15$ in the Spam-Enron stream and to $M_{\max} = 5$ in the other streams.

Table 2: Average accuracy of different algorithms on all streams.

	Spam	Spam-Enron	Spam1	Spam2	Reuters-Spam
CFIM	**88.68**	**88.37**	**91.78**	**93.67**	**88.70**
AUE-RT	**81.73**	**81.37**	**82.24**	88.26	80.40
AUE-RF	**86.59**	**87.28**	**89.75**	**89.68**	**85.72**
AUE-SVM	**85.33**	**85.86**	**85.94**	**89.33**	**85.74**
AWE-RT	77.24	75.71	70.69	46.74	78.31
AWE-RF	**82.45**	78.91	75.54	57.84	84.94
AWE-SVM	**81.11**	72.50	71.38	54.52	**85.27**
LB-RT	64.68	74.65	73.42	87.92	85.26
LB-RF	63.47	79.30	**84.27**	**88.70**	**85.27**
LB-SVM	76.09	72.68	50.69	**88.66**	85.27
OZA-RT	54.34	80.74	75.19	74.59	85.27
OZA-RF	67.80	**81.56**	79.39	74.07	85.27
OZA-SVM	70.88	72.50	78.28	74.33	85.27

Table 3: Kappa statistic of different algorithms on all streams.

	Spam	Spam-Enron	Spam1	Spam2	Reuters-Spam
CFIM	**70.54**	**62.66**	**62.31**	**32.81**	**3.89**
AUE-RT	**55.07**	43.67	**50.92**	**31.86**	**3.21**
AUE-RF	**65.64**	**55.96**	**62.91**	**29.85**	**0.38**
AUE-SVM	**62.41**	41.77	**51.37**	**23.12**	0.00
AWE-RT	47.05	42.13	32.43	4.90	**3.24**
AWE-RF	**58.00**	**55.45**	42.36	3.94	**1.88**
AWE-SVM	54.77	35.46	33.21	3.03	0.00
LB-RT	27.50	49.10	39.36	**17.70**	0.01
LB-RF	24.99	**61.44**	**60.54**	0.79	0.00
LB-SVM	45.29	46.78	16.30	0.00	0.00
OZA-RT	14.70	**56.09**	43.89	6.09	0.00
OZA-RF	31.44	**58.76**	54.45	5.92	0.00
OZA-SVM	36.42	53.38	47.32	6.29	0.00

6.5. Results. We experiment in 5 textual streams with concept drift under an imbalanced environment. The measures in the comparative experiment involve averaged accuracy, plotting accuracy, and kappa statistic. We outline the experimental results of different approaches regarding the average accuracy for all streams in Table 2. As observed from Table 2, CFIM achieves the highest average accuracy in comparison to all the algorithms in all the streams. Moreover, in five real-world streams under an imbalanced environment, average kappa statistic of the tested approaches in Table 3 shows that CFIM keeps the higher performance in terms of kappa statistic in most streams. We analyze the detailed experimental results in the five textual streams in the following paragraphs.

In Spam stream, the average accuracy produced by each algorithm is summarized in Table 2. Compared with AUE-RF, which is the second best algorithm, CFIM achieves 2.09% accuracy improvement. As seen in Figure 3, the curve of plotting accuracy about CFIM is always above the other two curves related to the algorithms with the second and the third highest average accuracy, respectively. At the 3rd time stamp, a large change may have occurred. The curves of all three algorithms tend to gradual increase and are relatively stable after the 3rd time stamp.

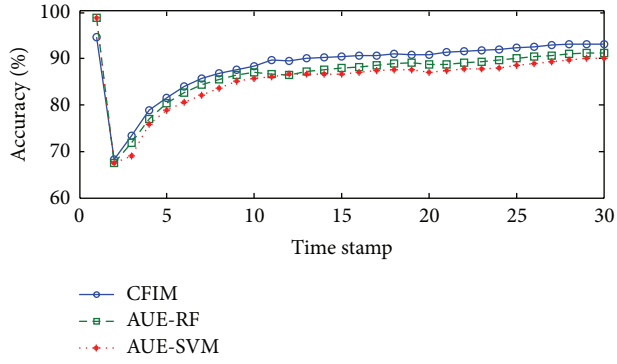

Figure 3: Plotting accuracies in the Spam Assassin stream.

The average accuracy of CFIM in Table 2 is higher than other comparative algorithms in Spam-Enron dataset. Figure 4 describes the best three algorithms' (CFIM, AUE-RF,

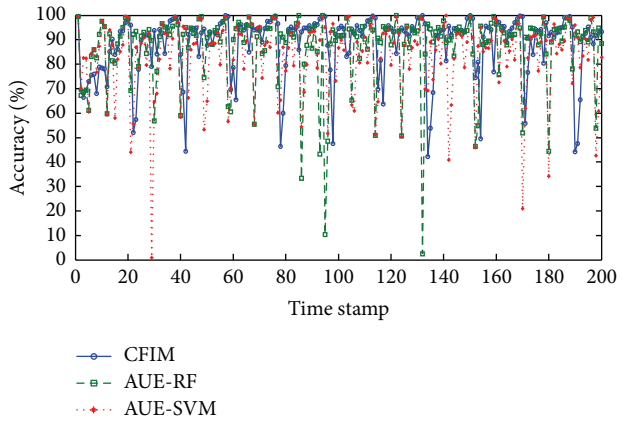

Figure 4: Plotting accuracies in the Spam-Enron stream.

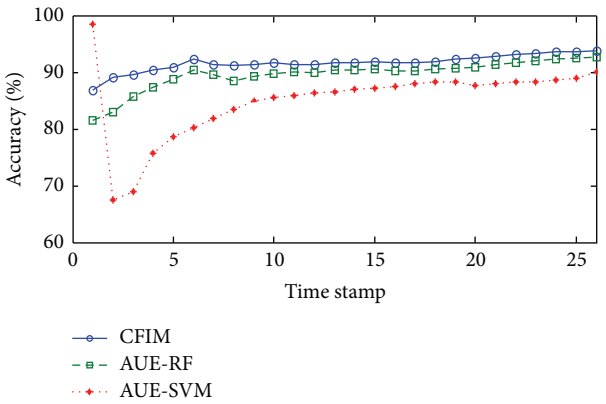

Figure 5: Plotting accuracies in the Spam1 stream.

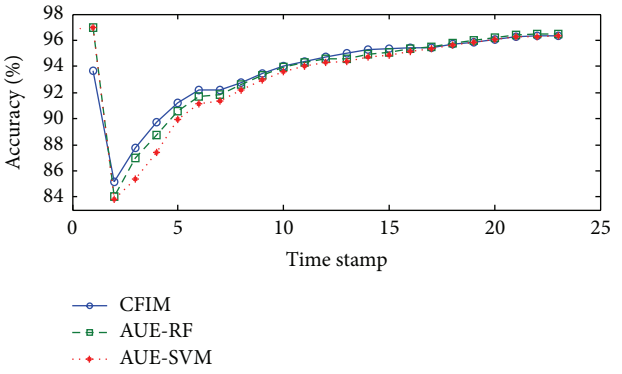

Figure 6: Plotting accuracies in the Spam2 stream.

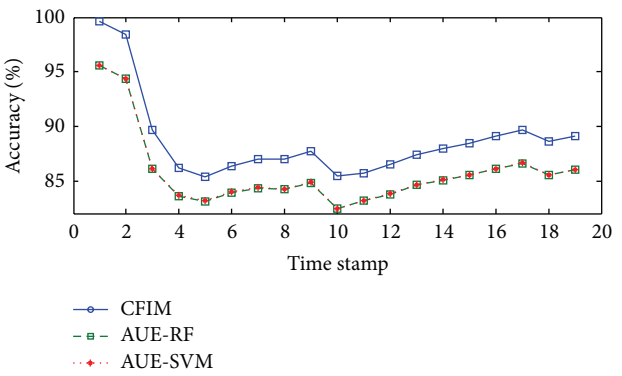

Figure 7: Plotting accuracies in the Reuters-Spam stream.

and AUE-SVM) plotting accuracies for Spam-Enron dataset. It is worth pointing out that CFIM accomplishes the better performance compared with the other two algorithms, as the fluctuation of accuracy curve in CFIM is the smallest. For example, plotting accuracy curve of AUE-RF fluctuates in a range of (16.6%, 100%) while the curve of CFIM is limited in the range of (49%, 100%). When a concept changes drastically, curves of AUE-SVM and AUE-RF fall down even below the level of CFIM's curve. In the periods of rebuilding concept, all of the three algorithms rebuild the model and tend to be stable. CFIM with a quick substitution of components creates a quick response for new concept, but the plotting accuracy of AUE-SVM grows slowly. This indicates that AUE-SVM is difficult to react to new concepts, especially after a sudden concept drift.

The average accuracy in the Spam1 stream is shown in Table 2. CFIM gives the best result (91.78%), while AUE-RF (89.75%) is the second best method. The average accuracy of CFIM is 5.84% higher than the third best algorithm, AUE-SVM. The plotting accuracies of three algorithms in the Spam1 stream in Figure 5 show that the curve of CFIM is always above the other two curves. In comparison to the curve of AUE-SVM with the dramatical decrement at the 2nd time stamp, the trends of curves produced by CFIM and AUE-RF seem to be similar. The plotting accuracy is increasing during the first six time stamps. After a slight drop at the 8th time stamp, the curves increase steadily from the 8th time stamp to the 26th time stamp.

Likewise, owing to both the highest accuracy and the quick response to sudden drifts, CFIM delivers the best performance in the Spam2 stream as presented in Table 2 and Figure 6. After the accuracies of all the models decrease at the 2th time stamp, all the models tend to ascend their accuracies by rebuilding themselves.

High-dimensional stream with concept drift is generated from Reuters. As shown in Table 2, CFIM gains the highest average accuracy (88.70%) over all the compared algorithms; AUE-SVM and AUE-RF are the second best algorithms by achieving similar accuracy. Figure 7 describes the classification results of the Reuters-Spam stream. CFIM shows a strong ability to deal with the concept drift by rebuilding the model quickly with high plotting accuracy.

The kappa statistic results are shown in Table 3. CFIM obtains the highest kappa statistic (70.54%) of all 13 algorithms in the Spam stream. AUE-RF with 65.64% kappa statistic and AUE-SVM with 62.41% kappa statistic are ranked in the second and the third place. Meanwhile, in the Spam1 stream, CFIM, AUE-RF, and LB-RF achieve the highest level of kappa statistic. Though CFIM outperforms the other 12 algorithms with respect to average accuracy, it gains slightly lower (0.6%) kappa statistic than AUE-RF but gains 1.77%

higher than LB-RF in the Spam1 stream. Compared with the 12 tested algorithms in the Spam-Enron stream in terms of kappa statistic, CFIM gains the highest performance (62.66%). The similar experimental results are observed in Spam2 stream and Reuters-Spam stream. According to three measures (average accuracy, plotting accuracy, and kappa statistic) to evaluate three imbalanced streams, we can observe that CFIM accomplishes promising performance compared with the other 12 tested algorithms.

7. Conclusion

In this paper, we present a new ensemble approach, CFIM, to handle the textual stream classification with class imbalance. This work involves the following aspects.

(i) We design an adaptive selection method to select the useful CTs and design a voting method depending on the accuracy weight to obtain the ensemble prediction result.

(ii) We construct the training set by sampling the instances from rare-class subset, misclassified subset, and the incoming chunks at a proper radio.

(iii) Experiments on real-world streams are carried out to evaluate the performances of the CFIM, AUE, AWE, LB, and OZA ensemble models based on three evaluation metrics: average accuracy, plotting accuracy, and kappa statistic. The experimental results illustrate that our CFIM approach is more effective than other tested algorithms in most of the streams.

In the future work, we further extend our work in several aspects. First, it is interesting to improve our proposed algorithm by redesigning the structure of CT for multilabel stream classification. Second, we will extend our approach to semisupervised stream classification. Third, we plan to improve our work to design a dynamic streaming clustering tree by an incremental method in order to save the running time.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported in part by NSFC under Grant no. 61073195, Shenzhen Science and Technology Program under Grant no. CXY201107010163A, and Shenzhen Strategic Emerging Industries Program under Grant nos. 0020JCYJ20130329142551746 and JCYJ20120613135329670.

References

[1] M. Scholz and R. Klinkenberg, "An ensemble classifier for drifting concepts," in *Proceedings of the 2nd International Workshop on Knowledge Discovery in Data Streams*, pp. 53–64, 2005.

[2] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 139–147, July 2009.

[3] Y. Li, E. Hung, and K. Chung, "A subspace decision cluster classifier for text classification," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12475–12482, 2011.

[4] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012.

[5] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 226–235, August 2003.

[6] I. Katakis, G. Tsoumakas, and I. Vlahavas, "On the utility of incremental feature selection for the classification of textual data streams," in *Knowledge Discovery from Data Streams*, vol. 3746 of *Lecture Notes in Computer Science*, pp. 338–348, Springer, Berlin, Germany, 2005.

[7] A. Bifet, G. Holmes, B. Pfahringer et al., "MoA: massive online analysis, a framework for stream classification and clustering," *Journal of Machine Learning Research*, vol. 22, pp. 44–50, 2010.

[8] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, pp. 784–793, August 2007.

[9] X.-L. Li, P. S. Yu, B. Liu, and S.-K. Ng, "Positive unlabeled learning for data stream classification," in *Proceedings of the 9th SIAM International Conference on Data Mining 2009 (SDM '09)*, pp. 256–267, May 2009.

[10] Y. Zhang, X. Li, and M. Orlowska, "One-class classification of text streams with concept drift," in *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDM '08)*, pp. 116–125, December 2008.

[11] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking recurring contexts using ensemble classifiers: an application to email filtering," *Knowledge and Information Systems*, vol. 22, no. 3, pp. 371–391, 2010.

[12] B. Liu, Y. Xiao, L. Cao, and P. S. Yu, "Vote-based LELC for positive and unlabeled textual data streams," in *Proceedings of the 10th IEEE International Conference on Data Mining Workshops (ICDMW '10)*, pp. 951–958, December 2010.

[13] J. Gao, B. Ding, W. Fan, J. Han, and P. S. Yu, "Classifying data streams with skewed class distributions and concept drifts," *IEEE Internet Computing*, vol. 12, no. 6, pp. 37–49, 2008.

[14] J. Gao, W. Fan, J. Han, and P. S. Yu, "A general framework for mining concept-drifting data streams with skewed distributions," in *Proceedings of the 7th SIAM International Conference on Data Mining*, pp. 3–14, April 2007.

[15] S. Chen and H. He, "SERA: selectively recursive approach towards nonstationary imbalanced stream data mining," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '09)*, pp. 522–529, New York, NY, USA, June 2009.

[16] R. N. Lichtenwalter and N. V. Chawla, "Adaptive methods for classification in arbitrarily imbalanced and drifting data streams," in *New Frontiers in Applied Data Mining*, vol. 5669 of *Lecture Notes in Computer Science*, pp. 53–75, Springer, Berlin, Germany, 2010.

[17] G. Ditzler and R. Polikar, "An ensemble based incremental learning framework for concept drift and class imbalance," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '10)*, pp. 1–8, Barcelona, Spain, July 2010.

[18] G. Ditzler, R. Polikar, and N. Chawla, "An incremental learning algorithm for non-stationary environments and class imbalance," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 2997–3000, Istanbul, Turkey, August 2010.

[19] A. Bifet and R. Kirkby, "Data stream mining a practical approach," 2009.

[20] P. Lindstrom, S. J. Delany, and B. M. Namee, "Handling concept drift in a text data stream constrained by high labelling cost," in *Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference (FLAIRS '12)*, pp. 32–37, AAAI Press, May 2010.

[21] R. Crouch, R. M. Kaplan, T. H. King, and S. Riezler, "A comparison of evaluation metrics for a broad-coverage stochastic parser," in *Proceedings of the LREC Beyond PARSEVAL workshop*, pp. 67–74, 2002.

[22] D. Brzeziński and J. Stefanowski, "Accuracy updated ensemble for data streams with concept drift," in *Hybrid Artificial Intelligent Systems*, vol. 6679 of *Lecture Notes in Computer Science*, pp. 155–163, Springer, Berlin, Germany, 2011.

[23] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases*, vol. 6321 of *Lecture Notes in Computer Science*, pp. 135–150, Springer, Berlin, Germany, 2010.