Available online at www.ijpe-online.com

International Journal of Performability Engineering

Vol. 13, No. 6, October 2017, pp. 945-955
DOI: 10.23940/ijpe.17.06.p15.945955

A Novel Ensemble Classification for Data Streams with Class Imbalance and Concept Drift

Yange Sun[a,b], Zhihai Wang[a,*], Hongtao Li[a], Yao Li[a]

[a]*School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China*
[b]*School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China*

Abstract

The processing of streaming data implies new requirements concerning restrictive processing time, limited amount of memory and one scan of incoming instances. One of the biggest challenges facing data stream learning is to deal with concept drift, i.e., the underlying distribution of the data may be evolving over time. Most of the approaches in the literature are under the hypothesis that the distribution of classes is balance. Unfortunately, the class imbalance issue is common in the real-world. And the imbalance issue further increases the difficulty of solving the concept drift problem. Motivated by this challenge, a novel ensemble classification for mining imbalanced streaming data is proposed to overcome both issues simultaneously. The algorithm utilizes the under-sampling and over-sampling techniques to balance the positive and negative instances. Moreover, dynamic weighting strategy was adopted to deal with concept drift. The experimental results on synthetic and real datasets demonstrate that our proposed method performs better than competitive algorithms, especially in situations where there exist concept drift and class imbalance.

Keywords: big data; data streams; ensemble classification; class imbalance; concept drift

(Submitted on July 25, 2017; Revised on August 30, 2017; Accepted on September 15, 2017)

(This paper was presented at the Third International Symposium on System and Software Reliability.)

1. Introduction

In recent years, with the technological advance, a lot of applications generate continuously arriving data, known as data streams [6,25]. Examples of such applications include sensor networks, spam filtering systems, traffic control, and intrusion detection. Data streams, which can be considered as one of the primary sources of what is called big data, arrive continuously with high speed [9].

There are two major challenges for these fields: First, due to the dynamic characteristics of data streams, the underlying distribution of the data may change dynamically over time, known as concept drift [10,12,22,24]. Concept drift leads to a drastic drop in classification performances. When there is a drift in data, the classification models should adapt to the changes quickly and accurately. Examples of real-world concept drifts include content changes in unwanted emails in spam categorization or evolving customer preferences [23]. Second, streaming data also has the problem of class imbalance, i. e., the number of negative instances is larger than other classes, which further increases the difficulty of solving the concept drift.

According to their speed, concept drifts have been divided into two types: sudden drifts and gradual drifts [12]. Sudden concept drift is characterized by large amounts of change between the underlying class distribution and the incoming instances in a relatively short amount of time. While gradual concept drift is featured by a large amount of time to witness a significant change in differences between the underlying class distribution and the incoming instances. Most of the existing methods just

* Corresponding author.
E-mail address: 13112074@bjtu.edu.cn.

deal with one of the two types. In fact, data streams in the real-world usually contain more than one type of concept drift. Thus, being able to track and adapt to various kinds of concept drift instantly is highly expected from a better classifier.

Concept drift has become a popular research topic over the last decade and many algorithms have been developed [12, 19]. The methods proposed for dealing with concept drifts can be organized into three groups: (i) window-based approaches, (ii) weight-based approaches, and (iii) ensemble classifiers. Among them, ensemble methods are suitable for dealing with concept drift.

There are several popular methods of handling class imbalance [4,15,17,21,26]. Some of the more popular approaches to learn class imbalance occur at a data or algorithmic level. These proposals can be categorized into four groups: (i) data level techniques, (ii) algorithm level approaches, (iii) cost-sensitive methods, and (iv) ensemble classifiers. Data level techniques concentrate on modifying the training set to make it suitable for a standard learning algorithm, such as random over/under-sampling. The best known algorithm is SMOTE (Synthetic Minority Over-sampling Technique). Algorithm level approaches seek to adapt existing algorithms to imbalanced datasets and bias them towards favoring the minority class. Cost-sensitive methods integrate both algorithm and data level approaches to incorporate different misclassification costs for each class in the learning phase. Finally, ensemble classifiers are widely used for learning class imbalance by combining multiple classifiers, sampling, and cost-sensitive learning. Out of these new solutions, ensembles are one of the most promising directions.

In the past decade, the issue of class imbalance has been studied extensively in the field of static classification. Unfortunately, the field of non-stationary learning of imbalanced data has received relatively less attention. Data streams with class imbalance are common in real-world applications such as intrusion detection in networks, fault diagnosis and credit card fraud detection etc.

The issue of learning from non-stationary data streams with imbalanced class distribution manifests largely into two sub-problems: (i) How can concept drifts be handled? (ii) How can imbalanced class distributions issue be managed? An algorithm should thus be carefully designed in a way that two issues could be effectively solved simultaneously.

In order to meet the above challenges, an efficient ensemble scheme is devised to address the joint issue. The key contribution of our algorithm is threefold:

(1) To solve the concept drift problem, we tailored ensemble learning to handle concept drift by continuously update the weights of the ensemble model.

(2) To address the class imbalance issue, the over-sampling technology was first adopted to increase the positive instances, and then the under-sampling technology was used to remove negative instances in order to achieve a balance between positive and negative instances.

(3) To enhance the performance of the ensemble, the weight of the classifier is updated based on the classification accuracy and classification error of the latest data block. At the same time, the contribution of the classifier to the accuracy of the integrated classification is taken into consideration during the elimination process of the classifiers.

The proposed algorithm was evaluated on a variety of datasets, and a comprehensive comparison study of was presented. The results demonstrated that our method obtains better performance than the competitive algorithms, especially under non-stationary data streams with imbalanced class environments.

The remainder of this paper is organized as follows. Section 2 retrospects the related work. In Section 3, the basic intuition behind our approach is discussed in detail. The algorithm is later analyzed and experimentally evaluated on real and synthetic datasets in Section 4. Finally, some conclusions are drawn and future researches are discussed in Section 5.

2. Related work

In this section, some relevant concepts are to be introduced first, and then some other previous work will be summarized.

2.1. Basic concepts and notation

Definition 1 (Data streams) A data stream is an infinite sequence of training instances:

$$S = \left\{\left(x_1, y_1\right), \left(x_2, y_2\right), \ldots, \left(x_t, y_t\right), \ldots\right\}$$

Each instance is a pair (x_t, y_t), where x_t is a d-dimensional vector arriving at the time stamp t, and y_t is the class label of x_t.

Definition 2 (Concept) The term of concept in data stream classification refers to the whole distribution of the problem at a certain point time, being characterized by the joint probability $P(x_t, y_t)$.

Definition 3 (Concept drift) Concept drift can be formally defined as any scenario where the posterior probability changes over time, i. e., $P_t(x_i, y_i) \neq P_{t+1}(x_i, y_i)$ [12].

2.2. Class imbalance machine learning in data streams

Dealing with unequal cardinalities of different classes is one of the contemporary challenges in batch learning. It has been more studied in this static framework and many new algorithms have already been introduced, however, they need to be generalized for data stream environments.

Gao et.al [13] proposed a general framework which summarizing the classification models on class imbalanced issue. In the algorithm, the incoming data block is separated into two parts, one represents the positive class instances (P) and another represents the negative class instances (Q). Generally, the size of P is smaller than the Q in the class imbalanced stream data. In order to take all them into account, choose some positive instances from P and get a subset of the Q then merge them together, then use all these instances to train a classifier, which usually has a better performance. Chen and He [5] proposed a novel algorithm, named SERA (Selectively Recursive Approach), which used a similar measure to select the previous minority instances. SERA is similar to the proposal of Gao et al.. It selects the "best" n minority class according to the Mahalanobis distance. Then it uses all majority class instances and uses bagging to build an ensemble of classifiers. Recently, Ditzler and Polikar [7] presented Learn^{++}.NIE, which is extension of Learn^{++}.NSE algorithm for address the imbalanced problem in non-stationary environments. Learn^{++}.NIE employs bagging rather than a single base learner, so it is considered that the algorithm can reduce error and create each bag on a less imbalanced dataset using under-sampling techniques. Finally, Mirza et al. [18] introduced a method of integrating extreme learning machine (ELM), which can be used both data and data blocks to solve class imbalances, but not considering concept drift issue.

Note that imbalanced data streams may not be characterized only by an approximately fixed class imbalance ratio over time. The relationships between classes may also be no longer permanent in evolving imbalanced streams. A more complex scenario is possible where the imbalance ratio and the notion of a minority class may change over time. It becomes even more complex when multi-class problems are being considered. The problem of class imbalance is often associated with concept drift in the data stream. However, most of the above algorithms are only for one of the problems, and the joint issue is not fully considered.

2.3. Ensemble classifiers for data streams with concept drift

Ensemble method provides a natural mechanism to update a knowledge base by adding, removing or updating classifiers [14]. Most ensemble-based algorithms adapt to concept drift by adding new or removing old classifiers to build an ensemble with incoming instances. Block-based ensembles have been designed to work in the environments where instances arrive in portions, called chunks or blocks. They periodically evaluate their components and substitute the weakest ensemble member with a new (candidate) classifier after each block of instances.

The first of such block-based ensembles was the Streaming Ensemble Algorithm (SEA) [20], which used a heuristic replacement strategy based on accuracy and diversity. Accuracy Weighted Ensemble (AWE) is a generic framework for mining concept drifting data streams [25]. The main idea is to train classifiers from sequential blocks of training instances. Each classifier is weighted and only the top K classifiers are kept. And the final output is based on the decision made by the weighted votes of the classifiers. The Accuracy Updated Ensemble (AUE1) [2], which incrementally trains its component classifiers after every processed block. Results obtained by AUE2 [3] suggested that by incremental learning of periodically weighted ensemble members one could preserve good reactions to gradual changes, while reducing the block size problem and, therefore, improving accuracy on suddenly changing streams.

3. Proposed Method

In this section, the general framework of block-based ensemble will be presented first, and then a novel sampling method will be introduced. Our proposed algorithm will be demonstrated in detail lastly.

3.1. General framework of Block-based ensemble

Let S represents incoming data stream, which is partitioned into sequential blocks, B_1, B_2, …, B_n, and there are *d* instances in every data block. When a block Bi is arriving, the weights of the base classifiers $C_i \in \varepsilon$ are calculated according to a classifier quality measure $Q(\cdot)$, which usually treated as a weighting function. The function acted an important role in many classification algorithms, how to get the function is based on the algorithm being analyzed. Such as AWE algorithm, which calculates weights based on the mean square error of the base classifiers. However, SEA updates the weights of the ensemble members based on accuracy and diversity. Besides, there still is a component reweighting, which is built from data block B_j as a candidate classifier and added to the ensemble if the number of the components is less than the maximum size *k*. We assume the ensemble's size is *k*, while there are *k* components in the ensemble and the candidate classifier has better performance in some measures than at least one member, then the candidate will replace the weakest ensemble member.

Algorithm 1: General framework of Block-based ensemble
Input: *S*: data stream, *d*: the size of block, *k*: the number of the base classifiers, $Q(\cdot)$: classifier quality measure.
Output: ε : ensemble of *k* weighted classifiers

```
01: begin
02:   for Bi∈S do
03:   train a candidate classifier on Bi ;
04:   calculate the weight of C' the using Q(·);
05:   weight all the classifiers in the ensemble using Q(·) and Bi;
06:     if |ε|<k then
07:     ε ← ε∪{C'};
08:     else if ∃i:Q(C')>Q(Ci) then
09:     replace weakest ensemble member with C' ;
10:     end if
11:   end for
12: end.
```

3.2. A novel sampling method

The method instances the data before classifying, uses the over-sampling method to increase the positive instances, and uses the under-sampling method to remove the negative instances. By taking both advantage of SMOTE and Tomek Links to achieve the purpose of increasing the positive class of samples, while also reducing the purpose of negative samples. For simplicity, this approach only considers two-class case, which can be extended to multi-class.

Algorithm 2: The Sampling Algorithm
Input: *T*: the number of positive instances , *N*%: over-sampling rate, *q*: the number of the nearest neighbors;
Output: new synthetic instances

```
01: begin
02:   numattrs;
03:   Instance[][];
04:   newindex;
05:   Synthetic[][];
06:   for i ← 1 to T
07:   nnarray ← find the q Nearest Neighbor of ith positive instance;
08:   generate new instances (N, i, nnarray);
09:   end for
10:   find out all Tomek link;
11:   delete negative instance of each Tomek link;
12: end.
```

3.3. Our method

To address the issue of learning from data streams with concept drift and imbalanced class, this paper introduced an Ensemble Classifier for Imbalanced Streaming Data (ECISD), which is based on integrated learning to overcome conceptual concept drift and class imbalance in data streams.

3.3.1. Ensemble Weighting

The weight of the classifier is updated dynamically according to the classification accuracy and the total cost of misclassification on the latest block. Hence, we introduce a cost matrix.

Table 1. Cost Matrix

	p	n
Y	0 (True Positives)	*FPcost* (False Positives)
N	*FNcost* (False Negatives)	0 (True Negatives)

where FPcost represents the cost of misclassifying negative classes for positive classes, FNcost means the cost of misclassifying positive classes for negative classes. In general, *FPcost* < *FNcost*.

Let $J_{y,y'}(x)$ represents the cost of instance *x* with real class *y* is classified y' : the total cost of the misclassification of the classifier C_j on B_i can be computed using equation (1):

$$ct_{ij} = \frac{1}{|B_i|} \sum_{\{x,y\} \in B_i} \sum_{\gamma} J_{y,\gamma}(x) \cdot f_{\gamma}^{j}(x) \tag{1}$$

The weight w_{ij} of C_j on the data block B_i is calculated according to equation (2):

$$w_{ij} = \frac{1}{ct_{ij} + MSE_{ij} - MSE_r + \alpha} \tag{2}$$

where α is a very small positive number to avoid the case of denominator is 0. Finally, the weights of all base classifiers need to be normalized.

3.3.2. Ensemble Pruning Strategy

In ECISD, when a new classifier is created, the original base classifier in the integrated classifier is evaluated. The overall correct rate of the integrated classifier is shown in equation (3):

$$P_{\varepsilon}(B_i) = \frac{\sum_{l=1}^{n} G_{\varepsilon}(x_l)}{N} \tag{3}$$

where x_l is an instance of data block B_i, $G_{\varepsilon}(x_l)$ is defined as equation (4):

$$G_{\varepsilon}(x_l) = \begin{cases} 1 & \text{correct} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

After removing C_j from ε , the classification accuracy of ensemble classifier is calculated using equation (5):

$$P_{\varepsilon\text{-}C_j}(B_i) = \frac{\sum_{l=1}^{n} G_{\varepsilon\text{-}C_j}(x_l)}{N} \tag{5}$$

The contribution of C_j to ε on data block Bi can be calculated using following equation:

$$Con_{\varepsilon}(j) = P_{\varepsilon}(B_i) - P_{\varepsilon\text{-}C_j}(B_i) \tag{6}$$

If the value of $Con_{\varepsilon}(j)$ is negative, showing that the base classifier reduces the overall classification accuracy. Otherwise, it indicates that the base classifier can improve the overall classification accuracy. When a new classifier is added, the base classifier is deleted if its Con value is less than zero.

ECISD trains a new classifier on each block of instances, using dynamically weighted majority voting scheme for the final decision of class label. To be specific, ECISD is done in two steps: sampling step and ensemble learning step. The procedure of ECISD is depicted in Figure 1, and the pseudo-code of ECISD algorithm is presented in Algorithm 3.

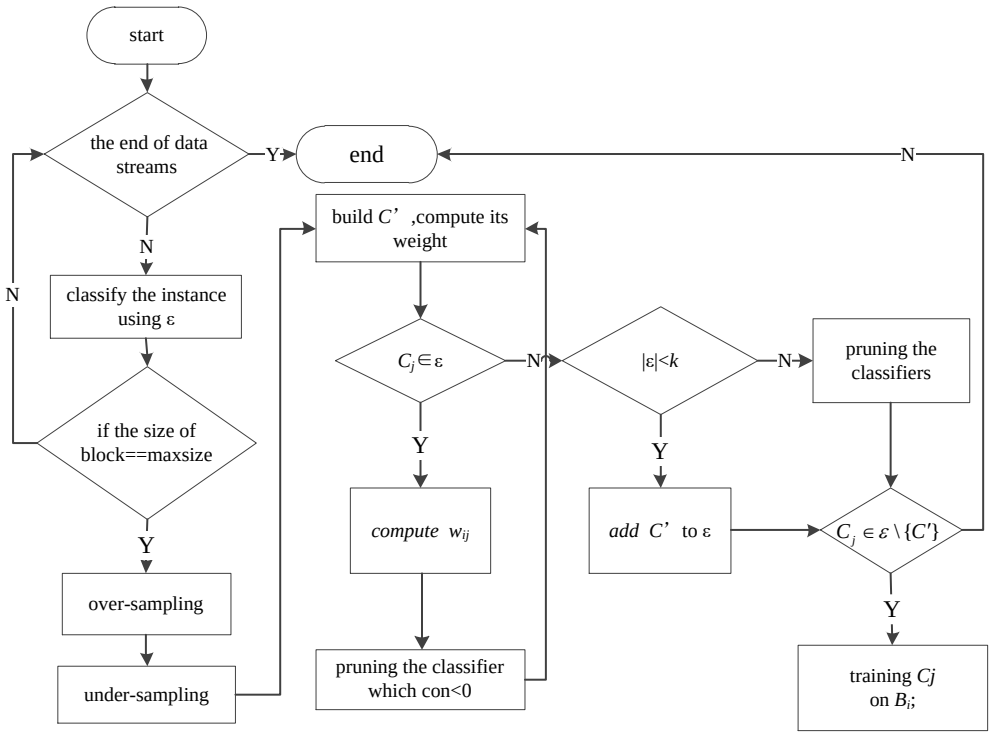

Figure 1. The procedure of ECISD

Algorithm 3: ECISD

Input: data streams S, max number of classifier k;
Output: a pool of classifiers $\mathcal{E}$;

```
01: begin
02:   for B_i∈S do
03:     C' ← new component classifier built on B_i
04:     w_C' ← 1/(MSE_r + α)
05:    for C_j ∈ E do
06:       compute MSE_ij, MSE_r, ct_ij ;
07:       w_ij ← 1/(ct_ij + MSE_ij − MSE_r + α)
08:       compute Con_ε(j)
09:       if Con_ε(j) < 0 then
10:         delete C_j;
11:       end if
12:     end for
13:     if |ε|< k then
14:       E ← ε ∪ {C'}
15:       else
16:      substitute least accurate classifier in E with C' ;
17:     end if
```

18: **for** $C_j \in \varepsilon \setminus \{C'\}$ **do**
19: incrementally train classifier C_j on B_i;
20: **end for**
21: **end for**
22: **end.**

3.4. Experimental Evaluation

The experiments are implemented in Java with help of Massive Online Analysis (MOA) (Version: MOA Release 2016.4) [1]. MOA is a framework implemented in Java for stream classification, regression and clustering.

3.4.1. Datasets

In our experiments, we adopted three synthetic and two real-world datasets. Synthetic datasets have several advantages: they are easier to reproduce and bear low cost of storage and transmission, and, most importantly, synthetic datasets provide an advantage of knowing the ground truth. For instance, we can know where exactly concept drift happens and what type of drift.

The STAGGER Concepts are boolean functions of three attributes encoding objects: size (small, medium, and large), shape (circle, triangle, and rectangle), and colour (red, blue, and green). STAGGER dataset was generated using STAGGER Generator in MOA. We used it to generate a stream of 1, 000, 000 instances with two sudden concept drifts.

The SEA dataset consists of three attributes, where only two are a relevant attributes. The points of the dataset are divided into four blocks with different concepts. In each block, the classification is done using $f_1 + f_2 \leq \Theta$, where f_1 and f_2 represent the first two attributes and Θ is a threshold value. The most frequent values are 9, 8, 7 and 9.5 for the data blocks. The SEA datasets contains 1, 000, 000 instances with sudden drifts reappearing every 250, 000 instances, and 10% of noise.

The Rotating spiral is a synthetic dataset consisting of four rotating spirals, two for each class [16]. Available from http://users.rowan.edu/~polikar/research/ NIE_data.

The Covertype dataset comes from UCI archive. It contains 581, 012 instances, 54 attributes, and no missing values. The goal is to predict the forest cover type based on cartographic variables.

Spam dataset is a real-world data stream that is chronologically ordered to represent the evolution of spam messages over time [11]. It consists of 9, 324 instances with 850 attributes. Spam messages is about 20% of the dataset. Hence it is a class imbalance dataset, and like many real datasets, a lack of knowledge about the concept drift cause and type.

3.4.2. Results and Discussion

To evaluate the effectiveness of the proposed algorithm, we adopted the prequential evaluation method [19]. This way the classifier is tested against all stream instances before seeing them. The experiments were conducted on 3.0 GHz Pentium PC machines with 16 GB of memory.

G-mean is the geometric mean of the recall of abnormal class and that of normal class. And it is often applied on class-imbalanced data streams to avoid the bias of the overall accuracy. Formally, *G*-mean is as follows.

$$G-mean=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}} \tag{7}$$

All the tested ensembles used k =15 component classifiers. Hoeffding Tree with default MOA parameters was chosen as base learner in all tested methods.

3.4.3. Parameter Sensitiveness

First, we study the effect of block size on our algorithm. The experimental design consisted of the independent variable: block size. The size of block was varied from 500 to 2000 to see how it affects the average *G*-mean of the algorithms. Table 2 presents the average *G*-mean of ECISD on different datasets while using different block sizes.

Table 2. Average *G*-mean using different block sizes

Datasets	Block Size						
	500	750	1000	1250	1500	1750	2000
STAGGER	0.7	0.76	0.82	0.89	0.84	0.80	0.72
SEA	0.89	0.92	0.98	0.97	0.95	0.90	0.87
Rotating Spiral	0.67	0.70	0.78	0.76	0.77	0.70	0.68
Covertype	0.69	0.74	0.72	0.65	0.64	0.66	0.61
Spam	0.53	0.59	0.63	0.62	0.60	0.61	0.55

The shape of curve that demonstrates the relationship between the size of the block and *G*-mean is shown in Figure 2. We observe that a too small block size will not provide enough amounts of instances for building a new classifier to be accurate, whereas a too large block size may contain more than one concept drift, delaying in reacting to new concepts accordingly. ECISD is not quite sensitive to the block size. As there is no strong dependency upon the block size in terms of *G*-mean, therefore in the following experiments we chose 1000 as the value.

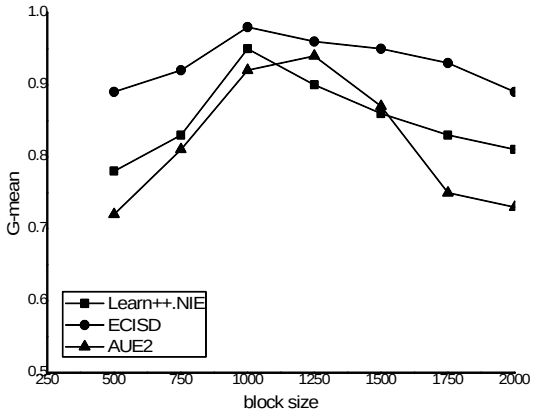

Figure 2. Average *G*-mean using different block sizes on SEA dataset

3.4.4. Comparative Performance Study

The proposed algorithm was evaluated against the following methods: Learn^{++}.NIE, Accuracy Updated Ensemble (AUE2) and Very Fast Decision Tree (VFDT) [8]. VFDT was chosen as representatives of single classifier for data streams.

It is found from Figure 3 that the trend of all algorithms is basically the same. Among them, Learn $^{++}$.NIE is the best, followed by ECISD, VFDT is the worst. The *G*-mean value of VFDT suffered a dramatic fluctuation at 400, 000th and the 800, 000th instance. The reason is the fact that ensemble classifiers adopt a combination of models to obtain better predictive performance than the single model VFDT. At the same time, the algorithm is based on the performance of the base classifier and the misclassification cost. Before the classification, the sampling mechanism is applied to handle the issue of class imbalance to improve the performance of the classifier. While VFDT does not have any concept drift mechanism, so it is not suitable for concept drift environments.

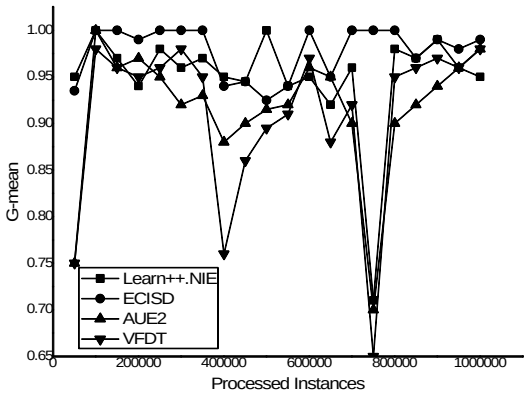

Figure 3. *G*-mean on STAGGER dataset

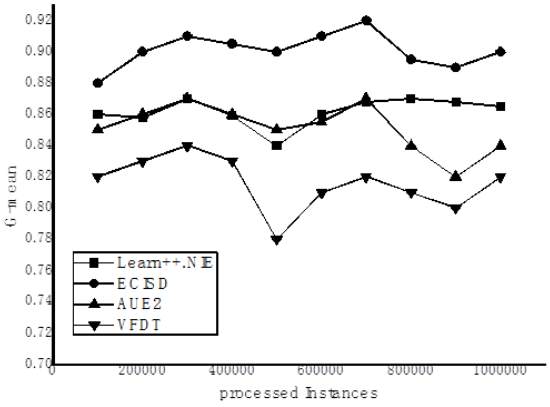

Figure 4. *G*-mean on SEA dataset

Figure 4 shows the performance of the algorithm on the SEA. When a concept drift occurred, the *G*-mean of all the algorithms will undergo instantaneous fluctuations except for our algorithm, which maintains a high, stable value and suffered the smallest drops. It is due to the fact that ECISD can quickly capture the changing concept, and then establish a new classifier, which can deal with this change accordingly.

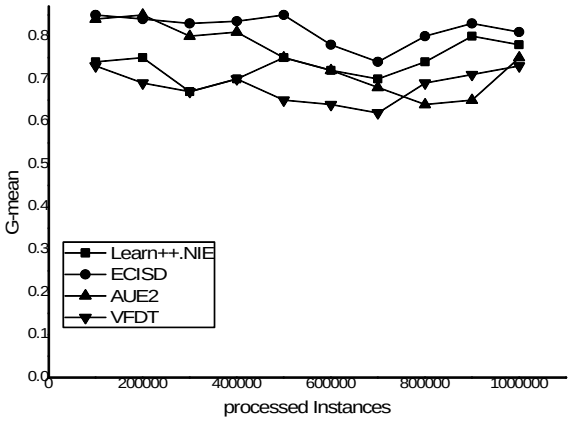

Figure 5. *G*-mean on Rotating Spiral dataset

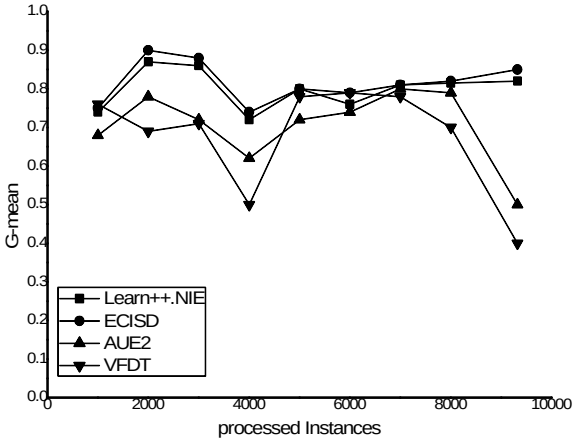

Figure 6. *G*-mean on Spam dataset

Figure 5 shows the performance of the algorithm on the Rotation spiral, which contains gradual concept drift. VFDT is the worst, and the performance of the other three ensemble algorithms is better than that of VFDT. The reason is the algorithm adopts the adjustment strategy and method of data imbalance problem in real-time stream data environment. VFDT does not perform very well in the presence of changes in data streams, possibly due to the fact that it always learns a concept from scratch upon trigger detection no matter whether there is a dift.

Then, it is supposed to verify the validity of the proposed algorithms on the real data stream dataset. The change of concept in real stream data is unpredictable and uncertain, so it can verify the generalization ability of the algorithm. Figure 5 reflects the performance of the real dataset Spam, the *G*-mean curves of all algorithms show varying degrees of fluctuation, which indicating that there is a concept drift in the dataset. Our algorithm and Learn++.NIE is relatively stable and is least affected by the concept drift. It is also indicates that the algorithm has good adaptability to the real data stream environment. While the performance of the VFDT algorithm and the AUE2 algorithm is poor.

Table 3. Average *G*-mean of algorithms

	VFDT	AUE2	Learn++.NIE	ECISD
STAGGER	0.93 (4)	0.95 (3)	0.97 (2)	**0.98 (1)**
SEA	0.79 (4)	0.83 (2)	**0.84 (1)**	0.82 (3)
Rotating Spiral	0.73 (4)	**0.79 (1)**	0.76 (3)	0.78 (2)
Covertype	0.69 (3)	0.73 (2)	0.67 (4)	**0.75 (1)**
Spam	0.52 (4)	0.56(3)	0.60 (2)	**0.63 (1)**
average rank	3.80	2.20	2.40	1.60

Finally, we analyze the average *G*-mean of the algorithms on the five datasets. As can be seen from Table 3, ECISD has the best performance on STAGGER, Covertype and Spam, poor performance on SEA, and the highest ranking of the ECISD algorithm. This is because ECISD can handle both class imbalances and conceptual drift issues. The VFDT algorithm lacks of dealing with class imbalances, and it does not handle the mechanism of concept drift, so its average performance is the worst.

To sum up, ECISD achieves better performance than the other methods from the following viewpoints: (1) The proposed method is capable of constructing a satisfactory model for handling different type of concept drifts and has been specifically built to deal with recurring concepts; (2) it better solves the class imbalance problem; and (3) it can adapt to a dynamic environment effectively and efficiently.

4. Conclusions

This study concerns about learning from data streams with class imbalance and concept drift. An efficient ensemble scheme is devised to address the joint issue. The purpose is to seek suitability for real-time data streams environment data imbalance regulation strategy by using the data sampling method to adjust the strategy of positive and negative instances in the whole model or the algorithm of contributions in the process of data processing, which ensures the algorithm can effectively adapt

to the change of data stream. We also designed solutions suitable for various types of concept drift, which was both based on the reduction of the imbalance distribution of data stream and on the consideration of the characteristics of different types of drift. The experimental results showed that the proposed algorithm could produce better performance on the concept drift in imbalanced data streams.

Since the proposed algorithm is a block-based ensemble, and the use of small chunk size may help to deal with the unexpected concept drift, it may lead to over-fitting due to the lack of training instances. On the contrary, the big data block may produce more accurate classifiers, which will not only consume more time and internal memory, but contain more concepts in the same data block. Therefore, how to design an integrated algorithm which is self-adaptive to block size is worth of further research.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China (No. 61672086; No. 61702030; No. 61771058), the Science and Technology Project of Henan Province (No. 172102210454) and Excellent Young Teachers Program of XYNU (No. 2016GGJS-08).

References

1. A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601-1604, May 2010.
2. D. Brzeziński, and J. Stefanowski, "Accuracy Updated Ensemble for Data Streams with Concept Drift," In *Proceedings of the 6th International Conference on Hybrid Artificial Intelligent Systems*, Berlin, Springer-Verlag, pp. 155-163, 2011.
3. D. Brzezinski, and J. Stefanowski, "Reacting to Different Types of Concept Drift: The accuracy updated ensemble algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 81-94, January 2014.
4. N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: Synthetic Minority Oversampling Technique,". *J. Artif. Intell. Res.*, vol. 16, pp. 321-357, 2002.
5. S. Chen, and H. He, "Towards Incremental Learning of Nonstationary Imbalanced Data Stream: a Multiple Selectively Recursive Approach," *Evol. Syst.*, vol. 2, no. 1, pp. 35-50, March 2011.
6. L. Cohen, G. Avrahami-Bakish, M. Last, A. Kandel, and O. Kipersztok, "Real-Time Data Mining of Non-stationary Data Streams from Sensor Networks," *Information Fusion*, vol. 9, no. 3, pp. 344-353, 2008.
7. G. Ditzler, and R. Polikar, "Incremental Learning of Concept Drift from Streaming Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol 25, no. 10, pp. 2283-2301, October 2013.
8. P. Domingos, and G. Hulten, "Mining High-Speed Data Streams," in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 71-80, August 2000.
9. R. Elwell, and R. Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517-1531, October 2011.
10. J. Gama, "Knowledge Discovery from Data Streams," New York: CRC Press, 2010.
11. J. Gama, R. Sebastiao, and P. P. Rodrigues, "On Evaluating Stream Learning Algorithms," *Machine Learning*, vol. 90, no. 3, pp. 317-346, March 2013.
12. J. Gama, I. Žliobaitė, A. Bifet, and M. Pechenizkiy, "A Survey On Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 231-238, April 2014.
13. 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. Minneapolis, Society for Industrial and Applied Mathematics*, pp. 3-14, 2007.
14. H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A Survey on Ensemble Learning for Data Stream Classification," *ACM Computing Surveys* (CSUR), vol. 50, no. 2, Article 23. pp. 1-36, June 2017.
15. H. He, and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, September 2009.
16. I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking Recurring Contexts Using Ensemble Classifiers: An application to email filtering," *Knowledge and Information Systems*, vol. 22, issue 3,pp. 371-391, March 2010.
17. C. X. Ling, V. S. Sheng, and Q. Yang, "Test Strategies for Cost-sensitive Decision Trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1055-1067, August 2006.
18. B. Mirza, Z. Lin, and K. A. Toh, "Weighted Online Sequential Extreme Learning Machine for Class Imbalance Learning," *Neural Processing Letters*, vol. 38, no. 3, pp. 465-486, December 2013.
19. G. D. F. Morales, A. Bifet, L. Khan, J. Gama, and W. Fan, "IoT Big Data Stream Mining," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 2119-2120, August 2016.
20. W. N. Street, and Y. S. Kim, "A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, pp. 377-382, August 2011.

21. Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-Sensitive Boosting for Classification of Imbalanced Data," *Pattern Recognition*, vol. 40, pp. 3358-3378, 2007.
22. A. Tsymbal, "The Problem of Concept Drift: Definitions and Related Work. Technical Report," Department of Computer Science, Trinity College, Dublin, Ireland, 2004.
23. G. I. Webb, R. Hyde, H. Cao, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing Concept Drift," *Data Mining and Knowledge Discovery*, vol. 30, Issue 4, pp. 964-994, July 2016.
24. G. Widmer, and M. Kubat, "Learning in the Presence of Concept Drift and Hidden Contexts," *Machine Learning*, vol. 23, no. 1, pp. 69-101, April 1996.
25. H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining Concept-Drifting Data Streams Using Ensembles Classifiers," in *Proceedings of Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 226-235, August 2003.
26. S. Wang, and X. Yao, "Diversity Analysis on Imbalanced Data Sets by Using Ensemble Models," in *Proceedings of the IEEE Symposium Series on Computational Intelligence and Data Mining*, Washington, DC: IEEE Computer Society, pp. 324-331, 2009.

Yange Sun was born in Henan Province, China, in 1982.She received the B.S. degree from the Xinyang Normal University (XYNU), Xinyang, in 2004 and the M.S. degree from the Central China Normal University (CCNU), Wuhan, in 2007, both in computer science. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Beijing Jiaotong University. Her research interests include data mining and machine learning.

Zhihai Wang was born in Henan, China, in 1963. He received the Doctor's degree from Hefei University of Technology in 1998. He is now a Professor in School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. He has published dozens of papers in international conferences and journals. His research interest includes data mining and artificial intelligence.

Hongtao Li was born in Shanxi, China, in 1993. She is currently pursuing the M. S. degree with the Computer and Information Technology, Beijing Jiaotong University. Her research interests include data mining and machine learning.

Yao Li was born in Anhui, China, in 1993. He is currently pursuing the M.S. degree with the Computer and Information Technology, Beijing Jiaotong University. His research interests include data mining and machine learning.