An Ensemble Classification Algorithm Based on Information Entropy for Data Streams

Junhong Wang · Shuliang Xu · Bingqian Duan · Caifeng Liu · Jiye Liang ·

Received: 02 Aug 2017 / Accepted: date

Abstract  Data stream mining problem has caused widely concerns in the area of machine learning and data mining. In some recent studies, ensemble classification has been widely used in concept drift detection, however, most of them regard classification accuracy as a criterion for judging whether concept drift happening or not. Information entropy is an important and effective method for measuring uncertainty. Based on the information entropy theory, a new algorithm using information entropy to evaluate a classification result is developed. It uses ensemble classification techniques, and the weight of each classifier is decided through the entropy of the result produced by an ensemble classifiers system. When the concept in data streams changing, the classifiers’ weight below a threshold value will be abandoned to adapt to a new concept in one time. In the experimental analysis section, six databases and four proposed algorithms are executed. The results show that the proposed method can not only handle concept drift effectively, but also have a better classification accuracy and time performance than the contrastive algorithms.

J. Wang
1. School of Computer and Information Technology, Shanxi University, Taiyuan, China
2. Key Laboratory of Computational Intelligence and Chinese Information Processing, Ministry of Education, Taiyuan, China
Tel.: +86 0351-7010566
E-mail: wjhwjh@sxu.edu.cn

S. Xu
School of Computer and Information Technology, Shanxi University, Taiyuan, China
E-mail: xushulianghao@126.com

B. Duan
School of Computer and Information Technology, Shanxi University, Taiyuan, China
E-mail: 6206486@qq.com

C. Liu
School of Computer Science and Technology, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China
E-mail: liucaifeng12345@qq.com

J. Liang
Key Laboratory of Computational Intelligence and Chinese Information Processing, Ministry of Education, Taiyuan, China
E-mail: ljy@sxu.edu.cn
1 Introduction

With the development of information society, many fields have produced a large amount of data streams, such as: e-commerce, network monitoring, telecommunication, stock trading, etc. The data in these fields are very different from the conventional static data, due to achieving fast, unlimited number and concept drift in data streams, it makes the technologies of traditional data mining face a big challenge [1].

Since data streams mining problem was proposed, it has received many attentions [2-6]. Especially, it has been widely used when the ensemble classification method was put forward [7]. Li et al. proposed a data stream classification algorithm based on a random decision tree model [8]. The algorithm creates a number of random decision trees, then randomly chooses the split attribute, and determines the split value through the information gain; in addition, the algorithm can effectively distinguish noise data and concept drift. Elwell et al. proposed an incremental learning algorithm for periodic concept drift called Learn++.NSE [9]. Learn++.NSE preserves the historical classification models; when historical classifier to classify the data correctly, the classifier will be improved the weight; after classifying correctly many times, the weight of the historical classifier will reach the activation threshold which is predetermined by a user; the classifier is activated from sleeping state, and then joins to the system to participate in deciding the labels of the unlabeled data. Aiming at the problem of the unbalanced data stream classification, Rushing et al. proposed an incremental algorithm based on the overlaying mechanism called CBEA [10]. While the algorithm training a classifier, the training examples are also saved. When the number of classifiers reaches the threshold, the two most similar coverage sets are selected, and the oldest classifier will be deleted. The final classification results are determined by KNN algorithm. Gomes et al. proposed an adaptive algorithm based on the social network called SAE [11], the algorithm introduces the related concepts of social network and each sub classifier seeing as a node on the network, so a network consists of multiple classifiers. SAE algorithm sets a series of parameters to measure the performances of the classifiers, and updates the classifiers to adapt to the new concept. Brzezinski et al. proposed a data stream classification algorithm based on online data called OAUE [12], the algorithm uses mean square error to determine the weights of classification models; when a period of time for the detection coming, the replacement strategy is used to deal with concept drift. Farid et al. proposed an ensemble classification algorithm based on instances weighting mechanism [13], in order to detect outliers, the algorithm combines with the clustering algorithm, if a data point does not belong to any existed cluster in the system, the class of the data will be thought as a new concept, and then the algorithm counts the data information in leaf nodes to further confirm the result. Liang et al proposed an online sequential learning algorithm called OS-ELM [14-17], OS-ELM is a development of the extreme learning machine (ELM) algorithm [18-22]; when a new data block coming, OS-ELM uses new data to incrementally update the structures of the single hidden feedforward neural networks; by the mean of the online sequential learning mechanism, OS-ELM can effectively deal with concept drift in data stream environment.

For the above ensemble classification algorithms, most of them detect the concept drift based on the accuracy of the classification result. In traditional data stream classification algorithm, accuracy is an important indicator which is used to detect the concept drift, but by using accuracy, it can only characterize current performance, and can not reflect the amount of information contained in
An Ensemble Classification Algorithm Based on Information Entropy for Data Streams

Information entropy can be used to measure the degree of uncertainty of a system, and it has been proved to be an effective method to describe the information content in information systems [38-41]. Information entropy is a powerful tool to deal with uncertainty problem, and it has been applied in many fields. Aiming at the problem of the data stream classification, in this paper, we extended information entropy to measure the uncertainty of the concept in data stream, and then an ensemble classification algorithm based on information entropy (ECBE) is proposed. The new algorithm is based on ensemble classification technique; the weighted voting rule is adopted, and the weights of classifiers are determined according to the change of entropy values before and after classification; by using Hoeffding bound, ECBE algorithm can estimate whether concept drift appears or not. When concept drift appearing, the algorithm automatically adjusts the classifiers according to their weights. Comparing with the existed algorithms, the ECBE algorithm not only can effectively detect the concept drift, but also can get better classification results.

The rest of the paper is organized as follows: in section 2, we describe the related backgrounds; section 3 introduces the ensemble classification algorithm based on entropy for data stream; section 4 is the experimental process and data analysis; finally, section 5 gives the conclusions and summary.

2 Backgrounds

2.1 Data stream and concept drift

We assume \{\dots, d_{t-1}, d_t, d_{t+1}, \dots\} is a data stream generated by a system, where \(d_t\) is the instance generating at \(t\) moment; for every instance, we mark \(d_t = \{x_t, y_t\}\), where \(x_t\) is the features vector, and \(y_t\) is the label. In order to illustrate related notions, we have the following definitions.

**Definition 1.** A certain number of instances are organized as a data set according to the time sequence, so we call the data set as data block, which denoted by \(\{d_1, d_2, \cdots, d_n\}\), where \(n\) is the size of data block.

In the data stream classification, because in the face of massive data, the storage space of the data is far beyond the capacity of the computer memory, in order to make the algorithm handle massive data, the sliding window mechanism is widely used [23], in other words, each time, the window is only allowed one data block coming into the system, only if the data blocks in the current window are processed completely, the next data blocks can be acquired.

**Definition 2.** At \(t\) moment, the data in sliding window is used to train a classifier, and the target concept we get is \(M\); \(\Delta t\) time later, we use new data to train another classifier, and we get the target concept is \(N\); if \(M \neq N\), we can say concept drift has happened in data stream. According to the different of \(\Delta t\), concept drift can be divided into two types: when \(\Delta t\) is a short time, concept drift is called gradual concept drift; when \(\Delta t\) is a long time, concept drift is called abrupt concept drift [24,42].

After concept drift appearing, the distribution of data in sliding window has changed, and \(p_t(y \mid x) \neq p_{t+1}(y \mid x)\); at this time, the performance of the classifiers will decline, if the corresponding measures do not be taken, the error rate of the classification results will continuously rise. Therefore, in many applications, this property is used to detect concept drift [25].
2.2 Ensemble classification techniques for data streams

Ensemble classification is an important classification method for data streams, which uses a number of weak classifiers to combine into a strong classifier; so the method can effectively deal with the problem of concept drift. In the process of classification, the ensemble classification gives different classifiers with different weights, by adjusting the weights of the sub classifiers to update classifiers system to adapt to the new concept of a data stream, classification result of the data eventually is decided by voting mechanism [26]. In the data stream environment, the classification performance of ensemble classifiers is better than that of single classifier [27]. Because of the many advantages of ensemble classification, the method is widely used in data stream data mining [28-30].

3 Ensemble classification based on information entropy in data streams

3.1 Concept drift detection based on information entropy

Entropy is used to describe the disordered state of thermodynamic system. In physics, entropy is used to indicate the disorder of a system. Shannon extended the concept of the entropy to the information system, and utilized information entropy to represent the uncertainty of a source [31]. The entropy of a variable is greater, and it indicates that the uncertainty of this variable is larger, so it needs more information when changing the state of the variable from uncertain to certain.

In the classification of data streams, when data blocks are in the sliding window, at \( t \) moment, we assumes a data block in the sliding window is \( B_i \), and \( B_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \cdots, (x_{in}, y_{in})\} \), where \( x \) is the feature matrix of the \( i \)th data block, \( x = \{x_{i1}; x_{i2}; \cdots; x_{in}\} \), and \( y \) is a label set of the \( i \)th data block, \( \{y_{i1}, y_{i2}, \cdots, y_{in}\} \subseteq y \). So at this time, the entropy of the data in the sliding window is calculated as the equation (1):

\[
H = -\sum_{j=1}^{\mid y \mid} p_j \log p_j
\]  

(1)

Where \( \mid y \mid \) is the number of the labels in the \( y \), and \( p_k \) is the probability of the label \( y_k \) in the instances; \( p_k \) can be calculated as the equation (2):

\[
p_j = \frac{\sum_{m=1}^{n} \mid y_{im} = y_{ij} \mid}{n}
\]  

(2)

In the equation (2), if \( y_{im} = y_{ij} \), \( \mid y_{im} = y_{ij} \mid = 1 \); else if \( y_{im} \neq y_{ij} \), \( \mid y_{im} = y_{ij} \mid = 0 \).

By utilizing the classifiers to classify the data block, the classification result can be obtained with a weighted voting mechanism; at the same time, we can use the equations (1)-(2) to calculate the entropy values of the classification result and the real result of the data block which are marked as \( H_1 \) and \( H_2 \) respectively. The entropy’s deviation of the data block can be calculated:

\[
\Psi(H) = \mid H_1 - H_2 \mid
\]  

(3)

**Lemma** (Hoeffding bound) [32, 33]. By independently observing a random variable \( r \) for \( n \) times, the range of the random variable \( r \) is \( R \), and the observed average of \( r \) is \( \bar{r} \); when confidence level is \( 1 - \alpha \), the true value of \( r \) is at least \( \bar{r} - \varepsilon \), where \( \varepsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\alpha})}{2n}} \).
Theorem 1. When the concept of a data stream (the distribution of the data) is stable, $S_1$ and $S_2$ are the deviation of the two adjacent data blocks which are calculated according to the equation (3), so it can be known that:

$$|S_1 - S_2| \leq 2 \sqrt{\frac{R^2 \ln(\frac{1}{\alpha})}{2n}}$$

(4)

Proof: When the concept of a data stream is stable, the data distributions of the two adjacent data blocks are consistent, and the difference of the observed entropy values is very small. Let $S_0$ be the real entropy of the data block according to the equation (3), from the Hoeffding bound, it is known that:

$$|S_1 - S_0| \leq \varepsilon \quad \text{and} \quad |S_2 - S_0| \leq \varepsilon$$

So it have:

$$-\varepsilon \leq S_1 - S_0 \leq \varepsilon; \quad -\varepsilon \leq S_0 - S_2 \leq \varepsilon$$

If plus the two inequalities, the result is as follow:

$$-2\varepsilon \leq S_1 - S_2 \leq 2\varepsilon$$

So it can get $|S_1 - S_2| \leq 2\varepsilon$.

Therefore, it can use $\Delta H = |\Psi(H_1) - \Psi(H_2)|$ as a measure to detect the concept drift in data streams, where $\Psi(H_1)$ and $\Psi(H_2)$ are the deviation of the two adjacent data blocks. If $\Delta H > 2\varepsilon$, it is thought concept drift has appeared. But it is obvious that when a special situation occurring, only using $\Delta H$ cannot work well: if the two adjacent data blocks represent different concepts, so the classification accuracies of the two adjacent data blocks are very low and they are almost equal; while calculating $\Delta H$, it is still less than or equal to $2\varepsilon$, and the algorithm will think no change happening, but in fact, concept drift has appeared at this time. In order to solve the problem, we correct the decision criterion of concept drift:

$$\Delta H > 2\varepsilon \quad \text{or} \quad \Psi(H_1) > 2\varepsilon \quad \text{or} \quad \Psi(H_2) > 2\varepsilon$$

(5)

we can say concept drift has appeared. From the equation (5), it is known that when the two adjacent data blocks occurring two concept drifts, $\Delta H$ will be less than or equal to $2\varepsilon$, however, $\Psi(H_1)$ or $\Psi(H_2)$ will be still greater than $2\varepsilon$, so concept drift will be detected.

For each sub classifier $ensemble(j)$, ($j = 1, 2, 3, \ldots, k$); according to the classification results of the current data block, the entropy of the classification result and real result marked as $h_i$ and $H_i$ can be calculated by using the equation (1); so the change of the entropy before and after classification is $\Psi(h_i) = |h_i - H_i|$, and the weights of the classifiers are updated according to the equation (6):

$$\text{ensemble}(j).weight(t) = \delta(\Psi(h_i)) \cdot \text{ensemble}(j).weight(t - 1)$$

(6)

In the equation (6), $ensemble(j).weight(t)$ is the new weight, and $ensemble(j).weight(t - 1)$ is the weight before updating. The function $\delta(\Psi(h_i))$ is showed as the equation (7):

$$\delta(\Psi(h_i)) = \begin{cases} 
\frac{1}{\varepsilon^{1+\Psi(h_i)}} & \Psi(h_i) > 0 \\
\beta, \quad \beta \text{ is constant and } \beta > 1 & \Psi(h_i) = 0 
\end{cases}$$

(7)
Theorem 2. If the concept of the data stream is stable, \( w_1, w_2, w_3, \ldots, w_n \) are the weights of the ensemble classifiers which are updated on the basis of the equation (6), and the confidence level is \( 1 - \alpha \), for each sub classifier, the weight must satisfy the following condition:

\[
    w_i \geq \frac{1}{n} \sum_{q=1}^{n} w_q - \frac{S}{\sqrt{n}} t_{\alpha}(n - 1) - 3S
\]

\[
    S^2 = \frac{1}{n - 1} \sum_{m=1}^{n} \left( \frac{1}{n} \sum_{q=1}^{n} w_q - w_m \right)^2.
\]

**Proof:** If the concept of a data stream is stable, each sub classifier in the system is adaptive to the current concept; therefore, the difference of the classifier weights before and after classifying is very small. We assume that the weights of the ensemble classifiers according with a normal distribution, so the mean and variance of the weights are calculated as:

\[
    \mu = \frac{1}{n} \sum_{q=1}^{n} w_q \quad \text{and} \quad S^2 = \frac{1}{n - 1} \sum_{m=1}^{n} (\mu - w_m)^2
\]

So we have:

\[
    P \left\{ \frac{\mu - w}{S} \leq t_{\alpha}(n - 1) \right\} = 1 - \alpha
\]

From the inequality, it can be known

\[
    P \left\{ w \geq \mu - \frac{S}{\sqrt{n}} t_{\alpha}(n - 1) \right\} = 1 - \alpha
\]

At the level of confidence \( 1 - \alpha \), we have

\[
    w \geq \mu - \frac{S}{\sqrt{n}} t_{\alpha}(n - 1)
\]

We use \( S \) to approximately replace the standard deviation of the normal distribution \( \delta \), according to the 3\( \delta \) principle of the normal distribution, it can conclude:

\[
    w_i \geq w - 3S \geq \frac{1}{n} \sum_{q=1}^{n} w_q - \frac{S}{\sqrt{n}} t_{\alpha}(n - 1) - 3S.
\]

From the theorem 2, we can see that when concept drift appearing, the algorithm will obtain the statistic about the weights of all classifiers from this time to the last time when the last concept drift happening, and the lower limit of the updating weight for each sub classifier is calculated as follow:

\[
    \theta_{\text{weight}} = \frac{1}{n} \sum_{q=1}^{n} w_q - \frac{S}{\sqrt{n}} t_{\alpha}(n - 1) - 3S \quad (8)
\]

According to the classification result of sub classifiers for the new data, the weights are updated by using the equation (6). When the classifiers cannot adapt to the current concept, the weights of the classifiers will sharply decreases below \( \theta_{\text{weight}} \). Finally, the system will delete all classifiers which does not satisfy the equation (8), so in the next process of the classification, the classifiers with low performances do not continue to participate in the decision-making.
3.2 The execution of the ECBE algorithm

From the above knowledge, we can know that the implementation steps of ECBE are as follows:

**Algorithm 1 ECBE**

**Input:** Ensemble classifier: $ensemble=NULL$; data stream $S$; the number of sub classifier: $k$; the size of data block: $winsize$; the array preserving the weights of the classifiers in the period of two adjacent concept drifts: $num=NULL$;

**Output:** The trained classifiers: $ensemble$.

1: while $S!=NULL$ do 
2: Read $winsize$ instances to organize a data block; 
3: if $size(ensemble)< k$ then 
4: Use $B_i$ to train a new classifier $C_j$; 
5: $ensemble ← ensemble ∪ C_j$ and $C_j.weight = 1$; 
6: else 
7: Use the equation (3) to calculate the entropies $Ψ(H_1)$ and $Ψ(H_2)$; 
8: for each $ensemble(t) ∈ ensemble$ //updating the weights of the classifiers do 
9: Calculate the entropy $h_t$ of the result which is from the $ensemble(t)$ classifying the data block $B_i$, and update the weights of the classifiers according to the equation (6); 
10: $num ← num ∪ ensemble(t).weight$; 
11: end for 
12: if $∆H > 2ε$ or $Ψ(H_1) > 2ε$ or $Ψ(H_2) > 2ε$ // concept drift then 
13: Delete the classifier with the minimum weight; 
14: Calculate the weight $θ.weight$ according to the equation (8); 
15: $num=NULL$; 
16: end if 
17: if $size(ensemble)≥k$ then 
18: Use $B_i$ to train a new classifier $C_{new}$; 
19: if $size(ensemble)≥k$ then 
20: Delete the classifier with the minimum weight; 
21: end if 
22: $ensemble = ensemble ∪ C_{new}$; 
23: Use $B_i$ to train each classifier in the system; 
24: end if 
25: end while

In the ECBE algorithm, it uses entropy to detect concept drift; if the data block does not appear concept drift, the classifiers system will have a good performance, so the classification results and the actual labels are nearly consistent, and the difference of the two entropies is very small. When concept drift appearing, the error rate of the classification result will increase; the difference of the entropy will also increase, the increment will promote that concept drift in the data stream can be detected. After concept drift is detected, some classifiers in the system are unable to adapt to the current concept; if the defunct classifiers are not eliminated, it will not only decreases the accuracy of the classification results, but also gives a false alarm about concept drift. In order to solve this problem, ECBE algorithm preserves all the weights of the classifiers where the concept of
the data is stable in a period of two adjacent concept drift. When concept drift happens, by saving the statistical weights, the lower bound of the weight when the concept is stable can be computed. Meanwhile, because of the concept changing, the weights of the classifiers will be dramatically reduced, so all the classifiers whose weights are below the lower bound will be deleted; the measure ensures that the classifier system can adapt to new concept at a very fast speed.

4 Experimental analysis

In order to validate the performance of ECBE algorithm, in this paper, we chosen SEA [34], AddExp [35], AWE [27], DCO [36], WEAP_I [4], TOS-ELM [20] and OS-ELM [14] as the comparison algorithms; C4.5, CART and Naive Bayesian et al. are used as base classifiers; all the algorithms were tested on the three artificial datasets: waveform, Hyperplane and LED which were produced by MOA platform [37] and the three practical datasets: sensor_reading_24, shuttle and page-blocks. The parameters of the experimental environment are as follows: Windows7 operating system, Intel dual core CPU, 4 G memory. The algorithm is implemented by R2013a Matlab.

4.1 The descriptions of the datasets

The datasets using in the experiment are cited from UCI datasets. We only give a brief explain about the datasets, and the detail can be see from the website.

- **waveform** dataset: it is an artificial dataset and the data set has 50000 instances. Each instance has 22 attributes; the first 21 attributes are numeric and there are 3 different labels in the dataset.
- **LED** dataset: the dataset has 50000 instances, each of which contains 25 attributes, and the values of the first 24 attributes are 0 or 1. There are 10 different labels in the dataset; the data contains 5% of the noise.
- **Hyperplane** dataset: a sample \(X\) in a \(d\) dimensional hyperplane satisfies the following mathematical expression: \(\sum_{i=1}^{d} a_i x_i = a_0\), where \(a_0 = \frac{1}{2} \sum_{i=1}^{d} a_i\). The dataset contains 50000 instances and 11 dimensions; the values of the first 10 dimensions are in \([0, 1]\). If \(\sum_{i=1}^{d} a_i x_i \geq a_0\), the label of the instance is marked as positive, otherwise, the label of the instance is marked as negative. In addition, the dataset contains 10% noise.
- **sensor_reading_24** dataset: the dataset contains 5456 instances and 25 properties. The first 24 properties are real; there are 4 different labels in the dataset.
- **shuttle** dataset: the dataset contains 43500 instances and 11 attributes. The values of the first 10 attributes are continuous; there are 7 different labels in the dataset.
- **page-blocks** dataset: the dataset contains 5473 instances and 11 attributes. In the first 10 attributes, some attributes values are continuous, and some of them are discrete. There are 5 different labels in the dataset.

4.2 The experimental results

In order to verify the performance of the ECBE algorithm, SEA, AddExp, AWE, DCO, WEAP_I, TOS-ELM and ECBE are run on the three artificial datasets: waveform, Hyperplane and LED. For
all the algorithms, $k = 5$, $\text{winsize}=2000$; the parameters of AddExp are as follows: $\beta = 0.5$, $\gamma = 0.1$; the parameters of ECBE are as follows: $\alpha = 0.05$, $\beta = 0.5$, $\gamma = 0.1$; the test results are showed in Tables 1 and 2.

### Table 1 Average accuracy on the artificial datasets

|            | waveform | Hyperplane | LED  |
|------------|----------|------------|------|
| SEA        | 0.6777   | 0.6925     | 0.2878 |
| AddExp     | 0.3701   | 0.5404     | 0.1622 |
| AWE        | 0.6653   | 0.6912     | 0.2715 |
| DCO        | 0.6628   | 0.7086     | 0.3040 |
| WEAP,1     | 0.7090   | 0.6349     | 0.4165 |
| TOSELM     | 0.5411   | 0.6349     | 0.3143 |
| ECBE       | 0.7583   | 0.7109     | 0.8145 |

### Table 2 Time overhead on the artificial datasets (Unit: sec)

|            | waveform | Hyperplane | LED  |
|------------|----------|------------|------|
| SEA        | 2743.422 | 1429.004   | 6.2156 |
| AddExp     | 1124.080 | 1123.530   | 1249.691 |
| AWE        | 2272.473 | 1004.433   | 4.8481 |
| DCO        | 1084.533 | 407.561    | 2.5660 |
| WEAP,1     | 225.9743 | 99.3225    | 4.0223 |
| TOSELM     | 0.01633  | 0.0338     | 0.1407 |
| ECBE       | 164.946  | 85.381     | 26.551 |

From Tables 1 and 2, it is known that, on the three artificial datasets, for ECBE, the accuracy is the highest and the time overhead is far less than the other compared algorithms on most datasets except for TOSELM; the comprehensive performance of ECBE is the best of all.

From the data Table 1, comparing with the other algorithms, AddExp has the lowest accuracy, and it is time consuming. The reason for the result is that AddExp updates the classifiers based on a single data; on the three datasets, the data contains gradual concept drift, and the concept of the data is constantly and slowly changing, which makes the accuracy of the old classifier decreasing; when an instance is classified wrongly by a classifier, the algorithm would eliminate the classifier to deal with the concept drift, however, the classifiers in the AddExp algorithm can get a better performance only after trained many times; due to the training for the new classifiers is inadequate, the error rate of the classification result is very higher, which will lead to the more classifier being replaced, so all the classifiers filling in the system are not trained adequately and the AddExp algorithm has a higher error rate on the three datasets.

From Table 1, for the LED dataset, the accuracies of the algorithms are very low except the ECBE algorithm. On the LED dataset, there are 7 attributes relating to concept drift, and the others are redundant attributes. When concept drift appearing, the speed of the concept changing on the LED dataset is faster than that on the waveform and Hyperplane datasets; the three algorithms: SEA, AddExp and AWE delete the classifiers with low performance on the basis of a single classifier, so it will take a long training time to completely eliminate the weak classifiers; when the speed of
concept drift is lower than the elimination speed, the performance of the classifiers will be always at a low level. For the ECBE algorithm, when a concept drift is detected, the weights of the classifier will be greatly reduced, however, ECBE will delete all weak classifiers according to their weights at a time, so ECBE can adapt to the new concept at a fast speed. In this experiment, it shows the performance of ECBE is the best of all.

In order to verify the effectiveness of the algorithm to deal with the practical datasets, SEA, AddExp, AWE, DCO, WEAP_I, TOSELM and ECBE are run on the sensor_reading_24, shuttle and page-blocks datasets. For all the algorithms, the maximum number of classifiers $k=5$; in the sensor_reading_24 and page-blocks datasets, the size of data block $wsize=200$ and on the others datasets, $wsize=2000$; the parameters of AddExp are as follows: $\beta = 0.5$, $\gamma = 0.1$; the parameters of ECBE are as follows: $\alpha = 0.05$, $\beta = 2$, $\gamma = 0.1$; the obtained results are shown in Tables 3 and 4.

Table 3 Average accuracies on the practical datasets

|                | sensor_reading_24 | shuttle | page-blocks |
|----------------|-------------------|---------|-------------|
| SEA            | 0.8168            | 0.8717  | 0.8923      |
| AddExp         | 0.5715            | 0.8127  | 0.7877      |
| AWE            | **0.8673**        | 0.8691  | **0.9235**  |
| DCO            | 0.8458            | 0.8615  | 0.9231      |
| WEAP_I         | 0.9500            | 0.9054  | 0.9035      |
| TOSELM         | 0.5608            | 0.8333  | 0.8973      |
| ECBE           | 0.7939            | **0.9137** | 0.9215     |

Table 4 Average time overhead on the real datasets (Unit: sec)

|                | sensor_reading_24 | shuttle | page-blocks |
|----------------|-------------------|---------|-------------|
| SEA            | 269.944           | 1163.610| 98.826      |
| AddExp         | 132.719           | 1028.906| 127.921     |
| AWE            | 235.112           | 945.015 | 66.186      |
| DCO            | 78.760            | 365.326 | 20.060      |
| WEAP_I         | 9.7250            | 54.9151 | 3.9338      |
| TOSELM         | 2.1260            | 2.1030  | 1.2650      |
| **ECBE**       | **1.678**         | **1.156** | **1.013**  |

From Tables 3 and 4, it can be known that all the algorithms get good performances on the practical datasets. The accuracy of ECBE is most prominent only on the shuttle dataset, but every accuracy of the algorithm is more than 0.79 and the accuracies of ECBE on the other datasets are very close to the best results; in addition, for the time overhead, ECBE is far less than the comparison algorithms, so the advantage of ECBE is very obvious.

From the data in Tables 3 and 4, the performances of SEA, AddExp, AWE, DCO, WEAP_I and TOSELM testing on the three practical datasets are very close and most of them have achieved good results. On the sensor_reading_24, shuttle and page-blocks datasets, the data whose labels are the same is highly concentrated; in most cases, the data distribution of two adjacent data blocks is consistent, and the classification model trained by the data has a better adaptability for new data, so the most performances of the eight algorithms are very good on the practical datasets. After
analyzing Tables 1-4, SEA, AddExp, AWE and DCO are effective for the datasets where the data with the same labels is highly concentrated, but when dealing with the datasets containing gradual concept drift, the performance of those algorithms are not satisfactory; if we analyze the results of ECBE, it is very obvious that ECBE can deal with the two types of concept drift.

In the ECBE algorithm, we use the equation (8) to calculate the threshold which judges the change of the entropy of data stream, and the threshold has an important influence on the performance of the algorithm; in the algorithm, the $n$ in the equation (8) is $20 \times \text{winsize}$. To explore the impact of the winsize values for the algorithm, we select the waveform dataset as the experimental data, and the parameters of ECBE are as follows: $\alpha = 0.05$, $\beta = 2$, $\gamma = 0.1$; we run the ECBE algorithm, and the results are showed in Table 5 and Fig.1-6.

**Table 5** Average accuracy with different winsize values

| winsize | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 | 2200 | 2400 |
|---------|-----|------|------|------|------|------|------|------|------|
| Hyperplan | 0.7050 | 0.6968 | 0.7051 | 0.7081 | 0.7041 | 0.7132 | 0.7109 | 0.7065 | 0.7080 |
| LED | 0.5961 | 0.8284 | 0.7403 | 0.8245 | 0.8165 | 0.8151 | 0.8145 | 0.8166 | 0.8131 |
| shuttle | 0.8922 | 0.9143 | 0.9002 | 0.9013 | 0.9115 | 0.9042 | 0.9979 | 0.9145 | 0.9604 |
| waveform | 0.7602 | 0.7565 | 0.7559 | 0.7556 | 0.7561 | 0.7561 | 0.7583 | 0.7519 | 0.7575 |

| winsize | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 190 |
|---------|----|----|----|----|----|----|----|----|----|
| page | 0.9409 | 0.9220 | 0.9275 | 0.9303 | 0.9096 | 0.9150 | 0.9109 | 0.9287 | 0.9193 |
| sensor | 0.9474 | 0.8000 | 0.7349 | 0.7913 | 0.6758 | 0.7724 | 0.8083 | 0.7664 | 0.7623 |

**Fig. 1** Test result with different winsize values on Hyperplane

In the Fig.1-6 and Table 5, under the situation of different winsize values, it can be seen, from the test result of the individual data blocks and the entire test results, there is no obvious linear rule between the winsize values and the performance of ECBE. For the ECBE algorithm, a small winsize value can lead to a large threshold, which can make ECBE is not too sensitive to the change of data distribution, but a small data block will cause the classifiers trained by the data are under
fitting, so the performance of the classifiers system will not be remarkably improved. If the winsize value is very large, it will lead to a small threshold, which makes the algorithm is too sensitive to the change of data distribution, so the classifiers will give a false alarm of concept drift and affect the performance of ECBE. Therefore, on a practical application, the value of winsize needs to consider the data distribution and the speed of concept changing; after trying many times, we select the optimal winsize value from a series of candidate values.

In order to study the anti noise performance of ECBE, we select the waveform dataset as the experimental data, and add 5%, 10%, 15% and 20% noise. The parameters of ECBE are set as follows: winsize=2000, $\alpha = 0.05$, $\beta = 2$, $\gamma = 0.1$. We run the ECBE algorithm on the datasets, and the test results are showed in the Fig.7 and Table 6.

From the Fig.7, it can be known that the accuracy of ECBE also gradually reduces with the increase of noise. From the table 6, we can see, comparing with the data containing 5% noise,
An Ensemble Classification Algorithm Based on Information Entropy for Data Streams

Fig. 4 Test result with different winsize values on waveform

Fig. 5 Test result with different winsize values on page

Table 6 Average accuracy of algorithm with different noise

| Noise | 5%  | 7%  | 10% | 12%  | 15%  | 17%  | 20%  | 25%  |
|-------|-----|-----|-----|------|------|------|------|------|
|       | 0.7348 | 0.7263 | 0.7167 | 0.7009 | 0.6929 | 0.6863 | 0.6842 | 0.6678 |

when the noise increasing 2%, 5%, 7%, 10%, 12%, 15% and 20%, the accuracy of ECBE only decreases 0.0085, 0.0181, 0.0339, 0.0419, 0.0485, 0.0506 and 0.067 respectively and the magnitude of the change is very small, thus it is known that ECBE has a good noise immunity. The good noise immunity of ECBE comes from the elimination strategy; when the noise in the data stream is increasing, the difference of the entropies of the classification results and the actual results increases, and the weights of classifiers will decay at a fast speed. If the weights are below a specified threshold,
Fig. 6 Test result with different winsize values in Hyperplane on sensor_reading

Fig. 7 Test result with different noise level

the classifiers trained by the noise data will be deleted, so the weak classifiers will not affect the classification of new data.

In order to further test the performance of ECBE, we compare ECBE with a representative neural network algorithm called OS-ELM on the 6 datasets and test OS-ELM with different activation functions. In the experiment, we choose radbas, sigmoid, sine and hardlim functions as the activation functions. The parameters of ECBE are as follows: \( \alpha = 0.05, \beta = 2, \gamma = 0.1 \); the test results are showed in Tables 7 and 8.

From Table 7, it can be seen, for the accuracy, ECBE is better than OS-ELM on 4 datasets; on the waveform and Hyperplane datasets, when the activation function of OS-ELM is sigmoid, OS-ELM is better than ECBE, but in fact, the advantage is very weak, after calculating, we can know the accuracies of OS-ELM are only more 0.0124 and 0.0194 than that of ECBE. In the aspect of time overhead, ECBE is better than OS-ELM on 4 datasets and OS-ELM is only better than ECBE.
Table 7  Average accuracy testing on datasets

| Dataset            | ECBE | OS-ELM | winsize |
|--------------------|------|--------|---------|
| waveform           | 0.7505 | 0.7629 | 0.3270 | 0.3595 | 0.7116 | 2500 |
| Hyperplane         | 0.7080 | 0.7274 | 0.6936 | 0.6763 | 2500   |
| LED                | 0.8119 | 0.2100 | 0.7480 | 0.4737 | 0.4673 | 2500 |
| sensor_reading_24  | 0.7938 | 0.5645 | 0.4366 | 0.4740 | 0.6562 | 300  |
| shuttle            | 0.9071 | 0.7825 | 0.8364 | 0.4618 | 0.8584 | 1500 |
| page-blocks        | 0.9321 | 0.7888 | 0.9127 | 0.4034 | 0.8977 | 300  |

Table 8  Time overhead on datasets (Unit: sec)

| Dataset            | ECBE | OS-ELM | winsize |
|--------------------|------|--------|---------|
| waveform           | 144.6957 | 74.5529 | 72.5873 | 71.9789 | 72.6185 |
| Hyperplane         | 68.7527 | 72.9289 | 74.4905 | 773.6481 | 73.6169 |
| LED                | 24.1768 | 73.3113 | 72.6981 | 73.9757 | 72.5395 |
| sensor_reading_24  | 0.8221 | 0.3586 | 0.2652 | 0.3486 |
| shuttle            | 7.7122 | 99.4194 | 24.1985 | 24.1958 | 24.1802 |
| page-blocks        | 0.8741 | 0.6552 | 0.2964 | 0.2187 | 0.3987 |

Table 9  The number of concept drift detected by ECBE

| Dataset | Hyperplane | SEA | waveform | RBF |
|---------|------------|-----|----------|-----|
| number  | 6          | 4   | 6        | 7   |
| accuracy| 0.6420     | 0.7460 | 0.746 | 0.2002 |
| winsize | 100        | 100  | 100      | 1650 |

on 2 datasets. If we synthetically consider the time overhead and accuracy of the two algorithms, on the sensor_reading_24, although ECBE costs more time, the accuracy of ECBE is far higher than that of OS-ELM, so ECBE is still better than OS-ELM. From the analyses, it is obvious that ECBE is the best of all.

In order to verify the ability of ECBE detecting the concept drift, we select Hyperplane, SEA, waveform and RBF as experimental datasets which are produced by MOA. In consideration of concept drift in the datasets unknown to us, so we reprocess Hyperplane, SEA and waveform, and make the three dataset contain 5 concept drift in which the label changes form 1 to 2 or reverse direction changing. The size of the four datasets are 3000, 3000, 3000 and 50000 respectively. The results are shown in Table 9 and Fig.8.

From Fig.8, we can see when concept drift appearing, the accuracy of ECBE will degrade rapidly, and then ECBE will delete classifiers with weak performance, so the accuracy will be restored to the previous high level. In the experiment, ECBE detects concept drift for 15 times; from Table 9, we can know, on the Hyperplane and waveform datasets, the algorithm has detected all concept drifts, but producing a error alert; on the SEA dataset, no error alert has produced, but a real changing is omitted; RBF is a gradual concept drift dataset which is generated by computer and the number of concept drift in the datset is unknown; we use RBF to test the effectiveness of ECBE handling gradual concept drift; from the result on RBF, it is obvious that ECBE can detect gradual concept drift. So in summary, we can include that although there are some deviations for concept...
drift detection in the algorithm, the experimental data indicates the mechanism of ECBE detecting concept drift is effective and ECBE can cope with different types of concept drift.

5 Conclusion

In this paper, to solve the classification problem of the data streams, an ensemble classification algorithm based on information entropy was proposed. The algorithm is on the basis of the ensemble classification method, and utilizes the change of entropies before and after classification to detect concept drift. The experimental results show that ECBE is effective. But it is obvious that the ECBE algorithm is only suitable for the single labeled data, so how to apply the algorithm to the classification of multi-label data is the focus of our future research.

Acknowledgments

This work was supported by National Natural Science Fund of China (Nos.U1435212, 61432011, 61202018, 61303008), National Key Basic Research and Development Program of China (973) (No.2013CB329404). The authors are grateful to the editor and the anonymous reviewers for constructive comments that helped to improve the quality and presentation of this paper.

References

[1] J. Gama, R. Sebasti, P.P. Rodrigues, Issues in evaluation of stream learning algorithms, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, (ACM, Paris, France, 2009), pp. 329-338.
[2] L.L. Minku, X. Yao, DDD: A New Ensemble Approach for Dealing with Concept Drift, IEEE Transactions on Knowledge & Data Engineering, 24 (2011) 619-633.
[3] A. Bilel, G. Holmes, B. Plaßhinger, R. Kirkby, Gavald, R., New ensemble methods for evolving data streams, ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (ACM 2009), pp. 139-148.
[4] Z Ouyang, M Zhou, et al. Mining concept-drifting and noisy data streams using ensemble classifiers, Artificial Intelligence and Computational Intelligence, 2009. AICI’09. International Conference on. Vol. 4. IEEE, 2009.
[5] X. Wu, P. Li, X. Hu, Learning from concept drifting data streams with unlabeled data, Neurocomputing, 92 (2012) 145-155.

[6] D. Liu, Y. Wu, H. Jiang, FP-ELM: An online sequential learning algorithm for dealing with concept drift, Neurocomputing, 207 (2016) 322-334.

[7] P. Jedrzejowicz, I. Czarnowski, R.J. Howlett, L.C. Jain, I. Czarnowski, P. J drzejowicz, Ensemble Classifier for Mining Data Streams, Procedia Computer Science, 35 (2013) 397-406.

[8] P. Li, X. Wu, X. Hu, H. Wang, Learning concept-drifting data streams with random ensemble decision trees, Neurocomputing, 166 (2015) 68-83.

[9] R. Elwew, R. Polikar, Incremental Learning of Concept Drift in Nonstationary Environments, IEEE Transactions on Neural Networks, 22 (2011) 1517-1531.

[10] J. Rushing, S. Graves, E. Criswell, A. Lin, A Coverage Based Ensemble Algorithm (CBEA) for Streaming Data, IEEE International Conference on TOOLS with Artificial Intelligence (2004), pp. 106-112.

[11] H. Murilo Gomes, F. Enembreck, SAE; Social Adaptive Ensemble classifier for data streams, Computational Intelligence and Data Mining (2013), pp. 199-206.

[12] D. Brzezinski, J. Stefanowski, Combining block-based and online methods in learning ensembles from concept drifting data streams, Information Sciences An International Journal, 265 (2014) 50-67.

[13] D.M. Farid, L. Zhang, A. Hossain, C.M. Rahman, R. Strachan, G. Sexton, K. Dahal, An adaptive ensemble classifier for mining concept drifting data streams, Expert Systems with Applications, 40 (2013) 5895-5906.

[14] N.Y. Liang, G.B. Huang, P. Saratchandran, N. Sundararajan, A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks, IEEE Transactions on Neural Networks, 17 (2006) 1411-1423.

[15] J.S. Lim, S. Lee, H.S. Pang, Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations, Neural Computing & Applications, 22 (2013) 569-576.

[16] V. Kamar, P. Gaur, A.P. Mittal, Trajectory control of DC servo using OS-ELM based controller, Power India Conference, 2012 IEEE Fifth (2012), pp. 1-5.

[17] Z. Yang, Q. Wu, C. Leung, C. Miao, OS-ELM Based Emotion Recognition for Empathetic Elderly Companion (Springer International Publishing, 2015).

[18] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 42 (2012) 513-529.

[19] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing, 70 (2006) 489-501.

[20] Gu, Yang, et al., TOSELM: timeliness online sequential extreme learning machine. Neurocomputing 128 (2014): 119-127.

[21] Z. Ma, G. Luo, D. Huang, Short term traffic flow prediction based on on-line sequential extreme learning machine, Eighth International Conference on Advanced Computational Intelligence (2016).

[22] P. Escandell-Montero, D. Lorente, J.M. Martinez-Martinez, E. Soria-Olivas, J. Vila-Frances, J.D. Martinez-Guerrero, Online fitted policy iteration based on extreme learning machines, Knowledge-Based Systems, 100 (2016) 200-211.

[23] H. Ryang, U. Yun, High utility pattern mining over data streams with sliding window technique, Expert Systems with Applications, 57 (2016) 214-231.

[24] H. Becker, M. Arias, Real-time ranking with concept drift using expert advice, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2007), pp. 86-94.

[25] J. Gama, P. Medas, P.P. Rodrigues, Learning decision trees from dynamic data streams, ACM Symposium on Applied Computing (2005), pp. 685-688.

[26] D. Brzezinski, J. Stefanowski, Reacting to different types of concept drift: the Accuracy Updated Ensemble algorithm, IEEE Transactions on Neural Networks & Learning Systems, 25 (2014) 81 - 94.

[27] H. Wang, W. Fan, P.S. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers. Kdd, (2003) 226-235.

[28] Z. Ouyang, M. Zhou, T. Wang, Q. Wu, Mining Concept-Drifting and Noisy Data Streams Using Ensemble Classifiers, International Conference on Artificial Intelligence and Computational Intelligence (2009), pp. 369-364.

[29] Q. Wei, Z. Yang, Z. Jumping, W. Yong, Mining multi-label concept-drifting data streams using ensemble classifiers, International Conference on Fuzzy Systems & Knowledge Discovery (2009), pp. 789.

[30] S. Ramanurthy, R. Bhattacharya, Tracking Recurrent Concept Drift in Streaming Data Using Ensemble Classifiers, International Conference on Machine Learning and Applications (2007), pp. 404-409.

[31] C.E. Shannon, A mathematical theory of communication (McGraw-Hill, 1974).

[32] L. Rutkowski, L. Pietruczuk, P. Duda, M. Jaworski, Decision Trees for Mining Data Streams Based on the McDiarmid’s Bound, Knowledge & Data Engineering IEEE Transactions on, 25 (2013) 1272-1279.

[33] P. Domingos, G. Hulten, Mining high-speed data streams, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2000), pp. 71-80.

[34] W.N. Street, A streaming ensemble algorithm (SEA) for large-scale classification, ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2001), pp. 1152-1165.

[35] J.Z. Kolter, M.A. Maloof, Using additive expert ensembles to cope with concept drift, International Conference, Bonn, Germany, August (2005), pp. 449–456.

[36] Y. Zhang, X. Jin, An automatic construction and organization strategy for ensemble learning on data streams, SIGMOD Rec., 35 (2006) 28-33.

[37] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, MOA: Massive Online Analysis, Journal of Machine Learning Research, 11 (2010) 1601-1604.

[38] Xie Z, Tie Y, Guan L. A new audiovisual emotion recognition system using entropy-estimation-based multimodal information fusion, IEEE International Symposium on Circuits and Systems. IEEE, 2015:726-729.

[39] Zhang X, Mei C, Chen D, et al., Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy, Pattern Recognition, 2016, 56(1):1-15.
[40] Chen Y, Zhang Z, Zheng J, et al., Gene selection for tumor classification using neighborhood rough sets and entropy measures, Journal of Biomedical Informatics, 2017, 67:59-68.
[41] Y Li, D Li, S Wang, Y Zhai, Incremental entropy-based clustering on categorical data streams with concept drift, Knowledge-Based Systems, 2014, 59:33-47.
[42] Y Yao, L Feng, The Research on Massive and Dynamic Data Stream Classification Method, Dalian University of Technology, 2013. (in Chinese)