Neural Processing Letters (2019) 50:2101–2117
https://doi.org/10.1007/s11063-019-09995-7

An Ensemble Classification Algorithm Based on Information Entropy for Data Streams

Junhong Wang[1] · Shuliang Xu[1] · Bingqian Duan[1] · Caifeng Liu[2] · Jiye Liang[1]

Published online: 6 February 2019

Abstract
Data stream mining has attracted much attention from scholars. In recent researches, ensemble classification has been wide aplied in concept drift detection; however, most of them regard classification accuracy as a criterion for judging whether concept drift happens 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 utilizes the methods of ensemble learning and the weight of each classifier is decided by the entropy of the result produced by an ensemble classifiers system. When the concept in data stream changes, the classifiers whose weight are below a predefined threshold will be abandoned to adapt to a new concept. In the experimental analysis, the proposed algorithm and six comparision algorithms are executed on six experimental data sets. The results show that the proposed method can not only handle concept drift effectively, but also have a better performance than the comparision algorithms.

Keywords Data streams · Data mining · Concept drift · Information entropy · Ensemble classification

1 Introduction

With the development of information society, many fields have generated a large amount of data streams, such as e-commerce, network monitoring, telecommunication, stock trading, etc. The data in these fields are different from the conventional static data; due to achieving fast, unlimited number and concept drift in data streams, it makes the methods of traditional data mining face a large challenge [14].

✉ Junhong Wang
wjhwjh@sxu.edu.cn

Shuliang Xu
slx_cs@mail.dlut.edu.cn

1 Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China

2 Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China

Since data streams mining problem was proposed, it has received much attention [1,4, 25,27,37]. Especially, it has been wide used when the method of ensemble classification was put forward [7]. Li et al. [22] proposed a data stream classification algorithm based on a random decision tree model. The algorithm creates a number of random decision trees; it randomly chooses the split attribute and the split value is determined by the information gain; in addition, the algorithm can effectively distinguish noise data and concept drift. Elwell et al. [10] proposed an incremental learning algorithm for recurrent concept drift called Learn++.NSE. Learn++.NSE preserves the historical classification models; when the historical classifier classifies the data correctly, the weight of the classifier will be improved; after classifying data correctly many times, the weight of the historical classifier will reach the activation threshold which is predetermined by a user; then the classifier is activated from sleeping state and joins to the system to participate in deciding the labels of the unlabeled data. Aiming at the classification problem of the unbalanced data stream, Rushing et al. proposed an incremental algorithm based on the overlaying mechanism called CBEA [30]. When the algorithm trains a classifier, the training samples are also saved. If the number of classifiers reaches the threshold, the two coverage sets whch are the most similar are selected and the oldest classifier will be deleted. The final classification results are determined by KNN algorithm. Gomes et al. [15] proposed an adaptive algorithm based on social network called SAE, the algorithm introduces the some concepts of social network and each sub classifier is seen as a node of a network, therefore the network is consisted of multiple classifiers. SAE algorithm sets a series of evaluation criteria to measure the performances of the classifiers and updates the classifiers to adapt to new concept. Brzezinski et al. [6] proposed a data stream classification algorithm based on the online learning mechanism called OAUE, the algorithm uses the mean square error to determine the weights of classification models; when a period of time for the detection comes, the replacement strategy is used to deal with concept drift. Farid et al. [12] proposed an ensemble classification algorithm based on the samples weighting mechanism; 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 this data will be seen a new concept and then the algorithm counts the information of data in leaf nodes to further confirm the result. Liang et al. proposed an online sequential learning algorithm called OS-ELM [21,23,24,38–40], OS-ELM is a development of extreme learning machine (ELM) algorithm [11,17–19,26]; when a new data block coming, OS-ELM uses the new data block to incrementally update the parameters of the single hidden feedforward neural network; by the method of the online sequential learning mechanism, OS-ELM can effectively deal with gruadual concept drift in data stream environment.

For the above, many classification algorithms for data stream detect the concept drift based on the accuracy of the classification result. Information entropy is a powerful tool to deal with the uncertainty problem and it has been applied in many fields. Aiming at the problem of the data stream classification, in this paper, we extend information entropy to measure the uncertainty of the concept in data stream and an ensemble classification algorithm based on information entropy (ECBE) is proposed. The new algorithm is based on the method of ensemble classification and the weighted voting rule is adopted; the weights of the classifiers are determined according to the change of the entropy values before and after classification; by Hoeffding bound, ECBE algorithm can estimate whether concept drift happens or not. When concept drift happens, the algorithm automatically adjusts the classifiers according to their weights. Comparing with the existed algorithms, ECBE algorithm not only can effectively detect the concept drift, but also can otain better classification results.

The rests of the paper are organized as follows: in Sect. 2, we describe the related backgrounds; Sect. 3 introduces the ensemble classification algorithm based on information entropy for data stream; Sect. 4 is the experimental result; finally, Sect. 5 makes a conclusion and describes the direction of our future research.

2 Backgrounds

2.1 Data Stream and Concept Drift

We assumes $\{\ldots, d_{t-1}, d_t, d_{t+1}, \ldots\}$ is a data stream generated by a system, where d_t is the instance generating at t moment; for each instance, we denotes $d_t = \{x_t, y_t\}$ 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 Several instances are organized as a data set according to the time sequence, therefore we call it as data block which is denoted by $\{d_1, d_2, \ldots, d_n\}$ where n is the size of the data block.

In the classification of data stream, because of massive data, the demand of the storage space is far beyond the memory of a computer. In order to make the algorithm handle massive data stream, the sliding window mechanism is wide used [32]; the window only allows one data block coming into the system; only when the data block in the current window is processed completely, can the next data block be acquired.

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

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 decrease; 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 [13].

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 group into a strong classifier; therefore 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, it updates the classifiers system to adapt to the new concept of a data stream and the classification result of the data eventually is decided by the voting mechanism [5]. In the data stream environment, the classification performance of the ensemble classifiers is better than the single classifier system [35]. Because of the many advantages of the ensemble classification, the method is wide used in data stream data mining [28,29,36].

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 the thermodynamic system. In physics, the value of entropy is used to indicate the disorder of a system. Shannon [33] extended the concept of the entropy to the information system and utilized information entropy to represent the uncertainty of a source. The entropy of a variable is greater and it indicates that the uncertainty of the variable is larger, therefore it needs more information when the state of the variable is changed from uncertain to certain.

In data stream, the weights of the data are considered. Generally speaking, the weight of a sample will decay with time going on. For a sample $\boldsymbol{x}_t$, the weight is defined as

$$f(\boldsymbol{x}_t) = e^{-\lambda(T-t)} \tag{1}$$

where T is the current time, λ is a predefined parameter and t is the time of $\boldsymbol{x}_t$ generated. Therefore the latest data point will have a greater weight. It is obvious that the time that a weight of a data point decays to β_0 will not more than $-\frac{\ln \beta_0}{\lambda}$.

In the classification of data streams, when the data blocks are in the sliding window, at t moment, we assumes the data block in the sliding window is $\boldsymbol{B}_i$. After the weight of each sample is considered, $\boldsymbol{B}_i = \{(f(\boldsymbol{x}_{i1}) \cdot \boldsymbol{x}_{i1}, y_{i1}), (f(\boldsymbol{x}_{i2}) \cdot \boldsymbol{x}_{i2}, y_{i2}), \ldots, (f(\boldsymbol{x}_{in}) \cdot \boldsymbol{x}_{in}, y_{in})\}$ where $\boldsymbol{x}$ is the feature matrix of the ith data block, $\boldsymbol{x} = \{f(\boldsymbol{x}_{i1}) \cdot \boldsymbol{x}_{i1}, f(\boldsymbol{x}_{i2}) \cdot \boldsymbol{x}_{i2}, \ldots, f(\boldsymbol{x}_{in}) \cdot \boldsymbol{x}_{in}\}$ and $\boldsymbol{y}$ is the label matrix of the ith data block, $\{y_{i1}, y_{i2}, \ldots, y_{in}\} \subseteq \boldsymbol{y}$. Therefore at this time, the entropy of the data in the sliding window is calculated as Eq. (2):

$$H = -\sum_{k=1}^{|y|} p_k \log p_k \tag{2}$$

where $|\boldsymbol{y}|$ is the number of the labels of the $\boldsymbol{y}$ and p_k is the probability of the label y_k in the data block; p_k can be calculated as In the Eq. (3):

$$p_k = \frac{\sum_{m=1}^{n} |y_{im} = y_{ik}|}{n} \tag{3}$$

In the Eq. (3), if $y_{im} = y_k$, $|y_{im} = y_{ik}| = 1$; else if $y_{im} \neq y_k$, $|y_{im} = y_{ik}| = 0$.

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 Eqs. (2)–(3) to calculate the entropy of the classification result and real result of the data block which are denoted as H_1 and H_2 respectively. The deviation of the entropy of the data block can be calculated:

$$\Psi(H) = |H_1 - H_2| \tag{4}$$

Lemma 1 (Hoeffding bound) [8,31]. *By independently observing the 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 the 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 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 Eq.* (4)*, therefore it can be known that:*

$$|S_1 - S_2| \le 2\sqrt{\frac{R^2 \ln(\frac{1}{\alpha})}{2n}} \tag{5}$$

Proof When the concept of the data stream is stable, the data distributions of the two adjacent data blocks are consistent and the difference of the observed entropy is small. Let S_0 be the real entropy of the data block, from *Hoeffding bound*, it is known that:

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

Hence we have:

$$-\varepsilon \le S_1 - S_0 \le \varepsilon; \ \ -\varepsilon \le S_0 - S_2 \le \varepsilon$$

If we add the above two inequalities, the result is as follow:

$$-2\varepsilon \le S_1 - S_2 \le 2\varepsilon$$

We can get $|S_1 - S_2| \le 2\varepsilon$. □

Therefore, it can use $\Delta H = |\Psi(H_1) - \Psi(H_2)|$ as a measure to detect 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 happened. However, it is obvious that when a special situation appears in sliding window, the detection method cannot work well only by using ΔH: if the two adjacent data blocks represent different concepts, hence the classification accuracies of the two adjacent data blocks are in a low level and the difference of them is small; after calculating ΔH, it is still less than or equal to 2ε and the algorithm will think no change happening, but in fact, concept drift has happened at this time. In order to solve the problem, we correct the decision criterion of concept drift:

$$\Delta H > 2\varepsilon \text{ or } \Psi(H_1) > 2\varepsilon \text{ or } \Psi(H_2) > 2\varepsilon \tag{6}$$

we can say concept drift has happened. From Eq. (6), it is known that when the two adjacent data blocks appear two concept drifts, ΔH will be less than or equal to 2ε; however, $\Psi(H_1)$ or $\Psi(H_2)$ will be still greater than 2ε, therefore concept drift can 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 denoted as h_i and H_i correspondingly can be calculated by using Eq. (2); therefore the change of the entropy before and after classification is $\Psi(h_i) = |h_i - H_i|$ and the weight of the classifier is updated according to Eq. (7):

$$ensemble(j).weight(t) = \delta(\Psi(h_i)) \cdot ensemble(j).weight(t-1) \tag{7}$$

In Eq. (7), $ensemble(j).weight(t)$ is the new weight and $ensemble(j).weight(t-1)$ is the weight before updating. The function $\delta(\Psi(h_i))$ is showed as Eq. (8):

$$\delta(\Psi(h_i)) = \begin{cases} \dfrac{\beta}{e^{1+\Psi(h_i)}} & \Psi(h_i) > 0 \\ \beta, \quad \beta \text{ is constant and } \beta \ge 1 & \Psi(h_i) = 0 \end{cases} \tag{8}$$

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 Eq. (7) and the confidence level is $1 - \alpha$; for each sub classifier, the weight satisfies the following condition:*

$$w_i \geq \frac{1}{n}\sum_{k=1}^{n} w_k - \frac{S}{\sqrt{n}} t_\alpha(n-1) - 3S$$

$$S^2 = \frac{1}{n-1}\sum_{m=1}^{n}\left(\frac{1}{n}\sum_{k=1}^{m} w_k - w_m\right)^2.$$

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 weight of the classifier before after classifying is small. We assume that the weight of the ensemble classifiers according with a normal distribution and the mean and variance of the weights can be calculated as:

$$\mu = \frac{1}{n}\sum_{k=1}^{n} w_k \text{ and } S^2 = \frac{1}{n-1}\sum_{m=1}^{n}(\mu - w_m)^2$$

Hence we have:

$$P\left\{\frac{\mu - w}{\frac{S}{\sqrt{n}}} \leqslant t_\alpha(n-1)\right\} = 1 - \alpha$$

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 δ; according to the 3δ principle of the normal distribution, it can conclude:

$$w_i \geq w - 3S \geq \frac{1}{n}\sum_{k=1}^{n} w_k - \frac{S}{\sqrt{n}} t_\alpha(n-1) - 3S.$$

□

From Theorem 2, we can see that when concept drift happens, the algorithm obtains the statistic about the weights of all classifiers from this time to the last time when the last concept drift happens; the lower limit of the updating weight for each sub classifier is calculated as follow:

$$\theta\_weight = \frac{1}{n}\sum_{k=1}^{n} w_k - \frac{S}{\sqrt{n}} t_\alpha(n-1) - 3S \tag{9}$$

According to the classification result of sub classifiers for the new data, the weights are updated using Eq. (7). When the classifiers cannot adapt to the current concept, the weights of the classifiers will sharply decreases below θ_weight. Finally, the system will delete all classifiers which do not satisfy Eq. (9), therefore in the next process of the classification, the classifiers with low performances do not continue to participate in the decision-making.

3.2 The Details of the ECBE Algorithm

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

Algorithm 1 The feature selection method for categorical attributes

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; β and λ;
Output: The trained classifiers: ensemble.

while $S \neq NULL$ **do**
  Read *winsize* instances to organize a data block;
  if $size(ensemble) < K$ **then**
    Use $\boldsymbol{B}_i$ to train a new classifier C_j;
    $ensemble \leftarrow ensemble \cup C_j$ and $C_j.weight \leftarrow 1$;
  else
    Use Eq.(4) to calculate the entropies $\Psi(H_1)$ and $\Psi(H_2)$;
    for ***each*** $ensemble(t) \in ensemble$ **do**
      Calculate h_t of the result which is from the *ensemble(t)* classifying $\boldsymbol{B}_i$;
      Update the weights of the classifiers according to Eq.(7);
      $num \leftarrow num \cup ensemble(t).weight$;
    end for
    if $\Delta H > 2\varepsilon$ or $\Psi(H_1) > 2\varepsilon$ or $\Psi(H_2) > 2\varepsilon$ **then**
      Delete the classifier with the minimum weight;
      Calculate the weight θ_weight according to Eq.(9);
      $num \leftarrow NULL$;
    end if
    Delete the classifier with weights less than θ_weight;
    Use $\boldsymbol{B}_i$ to train a new classifier C_{new};
    if $size(ensemble) >= K$ **then**
      Delete the classifier with the minimum weight;
    end if
    $ensemble \leftarrow ensemble \cup C_{new}$;
    Use $\boldsymbol{B}_i$ to train each classifier in the system;
  end if
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, therefore the classification results and the actual labels are nearly consisten and the difference of the two entropies is small. When concept drift happens, the error rate of the classification result will increase; the difference of the entropy will also increase and 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 decrease the accuracy of the classification results, but also produce 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 the concept drift happens, by saving the statistical information of the weights, the lower bound of the weights when the concept is stable can be computed. Meanwhile, because of the concept changing, the weights of the classifiers will be dramatically reduced, therefore all the classifiers whose weights are below the lower bound will be deleted; the measure ensures that the classifiers 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 [20], AWE [35], DCO [9], OS-ELM [23], TOSELM [16] and EOSELM [41] as the comparison algorithms; all the algorithms were tested on the three artificial data sets: *waveform*, *Hyperplane* and *LED* which were produced by MOA platform [3] and the three practical data sets: *sensor_reading_24*, *shuttle* and *page-blocks*. The parameters of the experimental environment are set as follows: Windows 7 operating system, Intel dual core CPU, 4 G memory. The algorithm is implemented by R2013a Matlab. For all the algorithms, $k = 5$; the parameters of AddExp are as follows: $\beta = 0.5$, $\gamma = 0.1$; the parameters of ECBE are as follows: $\lambda = 0.00002$, $\alpha = 0.05$, $\beta = 1$ and the base classifiers are CART; the parameters of TOSELM are as $w = 0.01$, $\epsilon = 0.1$, the activation function is *hardlim* and the number of nodes in hidden layer is set as *col* (*col* is the number of dimensions of data set); the parameters of EOSELM are set as $\lambda = 0.7$, the activation function is *hardlim* and the number of nodes in hidden layer is *col*.

4.1 The Descriptions of the Datasets

waveform dataset: it is an artificial data set and the data set has 50000 samples. Each instance has 22 attributes; the first 21 attributes are numeric and there are 3 different labels in the data set.

LED dataset: the data set has 50000 samples, 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 data set; the data contains 5% noise.

Hyperplane data set: 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 data set contains 50000 samples 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 sample is as positive; otherwise, the label of the instance is negative. In addition, the data set contains 10% noise.

sensor_reading_24 data set: the data set contains 5456 samples and 25 properties. The first 24 properties are numberic; there are 4 different labels in the data set.

shuttle data set: the data set contains 43500 samples and 11 attributes. The values of the first 10 attributes are numberic; there are 7 different labels in the data set.

page-blocks data set: the data set contains 5473 samples and 11 attributes. In the first 10 attributes, some attributes values are numberic and some of them are categorical. There are 5 different labels in the data set.

4.2 The Experimental Results

In order to verify the performance of ECBE algorithm, SEA, AddExp, AWE, DCO, TOSELM, EOSELM and ECBE are run on the three artificial datasets: *waveform*, *Hyperplane* and *LED*. *winsize* = 2000. The test results are showed in Tables 1 and 2. The bold in all tables represent the best values in a column.

From Tables 1 and 2, it is known that, on the three artificial data sets, the accuracies of ECBE are the highest and the time overheads of ECBE are far less than the other compared algorithms on the most data sets; therefore it can conclude that the comprehensive performance of ECBE is the best of all.

Table 1 Average accuracy on the artificial data sets

	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
TOSELM	0.6065	0.6356	0.4501
EOSELM	0.6693	0.5702	0.4262
ECBE	**0.7480**	**0.7052**	**0.6649**

Table 2 Time overhead on the artificial data sets (Unit: s)

	waveform	Hyperplane	LEDc
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	2.566
TOSELM	0.129	0.111	0.135
EOSELM	**0.089**	**0.088**	**0.006**
ECBE	39.849	21.343	32.213

From the results of Table 1, comparing with the other algorithms, AddExp has the lowest accuracy and it is a time-consuming algorithm. The reason for the result is that AddExp updates the classifiers based on a single data; for the three data sets, all of them contain gradual concept drift and the concept in data stream is slowly changing which makes the accuracy of the old classifier be in a state of decreasing; when a sample is classified wrongly by a classifier, the algorithm eliminates the classifier to adapt to concept drift; however, the classifiers of AddExp algorithm can get a better performance only after it is trained by many data blocks; in other words, the new classifier has a weak approximation ability and the error rate of the classification result is high which leads to more classifier being replaced, hence all the classifiers filling in the system are not trained adequately and AddExp has a higher error rate than the other algorithms on the three data sets.

From Table 1, for the *LED* data set, the accuracies of the algorithms are in a low level except ECBE algorithm. On the *LED* data set, there are 7 attributes relating to concept drift and the others are redundant attributes. When concept drift happens, the changing speed of concept on the *LED* data set is faster than that on waveform and Hyperplane data sets; the four algorithms: SEA, AddExp, AWE and EOSELM delete the classifiers with a low performance on the basis of a single classifier, therefore it takes a long time to completely eliminate weak classifiers; when the speed of concept drift is faster than the elimination speed of classifiers, the performance of the classifier keeps at a low level. Although there is only one classifier for TOSELM, however TOSELM has no the mechanism of concept drift detection, therefore it cannot deal with data stream with concept drift. For ECBE algorithm, when a concept drift is detected, the weights of the classifier decay and ECBE deletes all weak classifiers according to their weights at this time, therefore ECBE can adapt to new concept at a fast speed. From the results, it can conclude that the performance of ECBE is better than the comparison algorithms on the most data sets.

In order to verify the effectiveness of ECBE to deal with the practical data sets, SEA, AddExp, AWE, DCO, TOSELM, EOSELM and ECBE are run on the *sensor_reading_24*,

Table 3 Average accuracies on the practical data sets

	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
TOSELM	0.5565	0.7790	0.8812
EOSELM	0.5285	0.8050	0.8634
ECBE	0.8463	**0.9129**	0.9183

Table 4 Average time overhead on the real data sets (Unit: s)

	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
TOSELM	0.100	0.073	0.088
EOSELM	0.069	0.083	0.078
ECBE	**0.603**	**0.782**	**0.358**

shuttle and *page-blocks* data sets, For all the algorithms, the maximum number of classifiers $k = 5$; in the *sensor_reading_24* and *page-blocks* data sets, the size of data block $winsize = 200$ and on the other data sets, $winsize = 2000$; the obtained results are shown in Tables 3 and 4.

From Tables 3 and 4, it is known that the performance of ECBE is not the best of all. In Table 3, ECBE obtains the best result only on *shuttle* data set. However, ECBE obtains excellent results on the most data set. The the average accuracies of SEA, AddExp, AWE, DCO, TOSELM, EOSELM and ECBE are 0.8603, 0.7240, 0.8866, 0.8768, 0.7389, 0.7323 and 0.8925 correspondingly. It is obvious that the average accuracy of ECBE is the best of all.

From the results of Tables 3 and 4, the performances of SEA, AddExp, AWE and DCO testing on the three practical data sets are closely to each other and most of them have achieved good results. On the *sensor_reading_24*, *shuttle* and *page-blocks* data sets, the data with the same labels are highly concentrated. For the three data sets, the data distribution of two adjacent data blocks is similar in most cases, therefore the classification model trained by current data block has an excellent approximation ability for next data block which produces the result that the performances of the four algorithms are good on the practical data sets. After analyzing Tables 1, 2, 3 and 4, it can know that SEA, AddExp, AWE and DCO are effective for the data sets; when dealing with data sets with gradual concept drift such as *waveform* and *Hyperplane* data set, the performance of ECBE is also excellent. For TOSELM and EOSELM, they have lower time costs, but the accuracies are also in a low level. By synthesizing the results of ECBE, it is obvious that ECBE can deal with the two types of concept drift.

In ECBE algorithm, Eq. (9) is used to calculate the threshold which judges concept drift whether happens or not. The threshold has an important influence on the performance of ECBE; in the algorithm, the n in Eq. (9) is set as $20 \times winsize$. To explore the impact of the *winsize* values for the algorithm, we select the *waveform* data set as the experimental data set; we run ECBE algorithm on the data sets and the results are showed in Table 5 and Fig. 1.

Table 5 Average accuracy with different *winsize* values

winsize	800	1000	1200	1400	1600	1800	2000	2200	2400
Average accuracy	0.7633	0.7586	0.7549	0.7539	0.7501	0.7526	0.7480	0.7493	0.7480

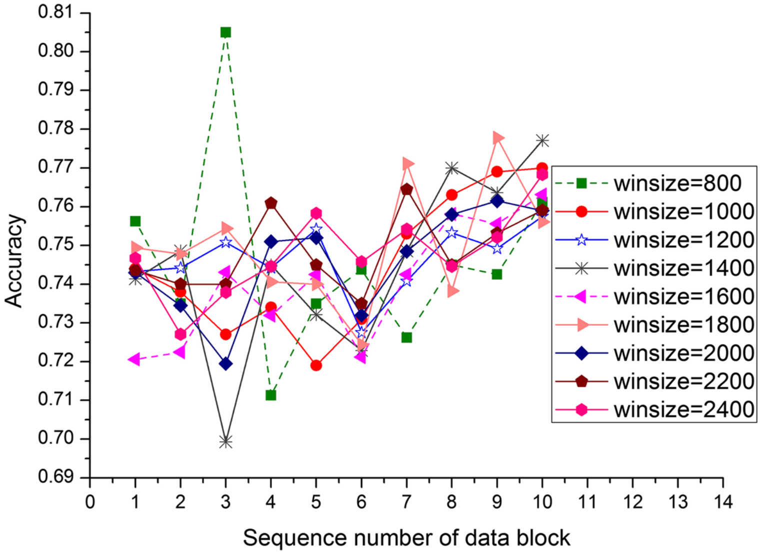

Fig. 1 Test result with different *winsize* values

Table 6 Average accuracy of algorithm with different noise

Noise	5%	7%	10%	12%	15%	17%	20%	25%
Average accuracy	0.7335	0.7247	0.7141	0.7020	0.6879	0.6844	0.6676	0.6535

From Fig. 1 and Table 5, it can be seen that there is no obvious linear rule between the *winsize* values and the performance of ECBE. For ECBE algorithm, a small *winsize* value can lead to a large threshold which can make that ECBE is not too sensitive to the change of data distribution, however a small data block will cause that the classifiers are under fitting, therefore the performance of the classifiers will not be remarkably improved. If the *winsize* value is large, it will lead to a small threshold which makes the algorithm is too sensitive to the change of data distribution; the classifiers will give a false alarm of concept drift and affect the performance of ECBE. Therefore the value of *winsize* needs to consider the distribution of data set and the changing speed of concept in practical applications; after trying different values, we can 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* data set as the experimental data and add 5%, 7%, 10%, 12%, 15%, 17%, 20% and 25% noise into the data set. The parameters of ECBE is set as follow: *winsize*=2000. We run ECBE algorithm and the test results are showed in Fig. 2 and Table 6.

From the Fig. 2, it can be known that the accuracy of ECBE gradually reduces with the increase of noise. From Table 6, we can see, comparing with the data set containing 5% noise,

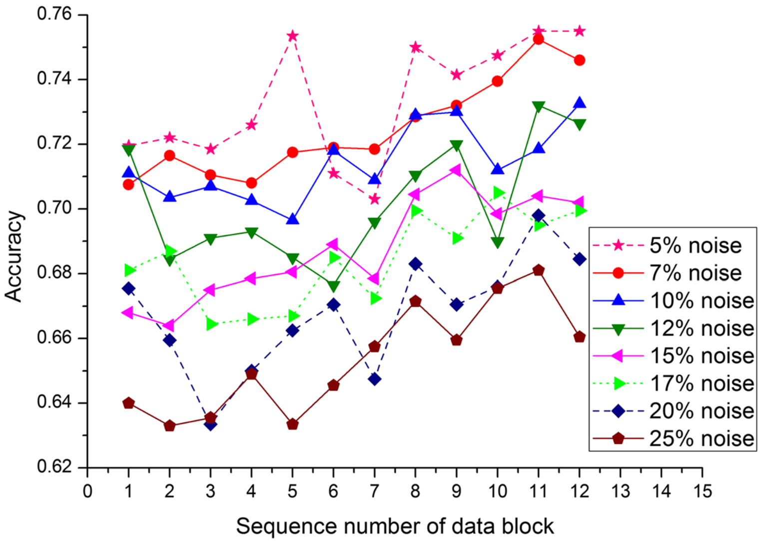

Fig. 2 Test result with different noise level

Table 7 Average accuracy testing on datas ets

	ECBE	OS-ELM				winsize
		radbas	sigmoid	sine	hardlim	
waveform	0.7511	0.3270	**0.7629**	0.3595	0.7116	2500
Hyperplane	0.7070	0.7020	**0.7274**	0.6936	0.6763	2500
LED	0.6545	0.2100	**0.7480**	0.4737	0.4673	2500
sensor_reading_24	**0.8142**	0.5645	0.4366	0.4740	0.6562	300
shuttle	**0.9037**	0.7825	0.8364	0.4618	0.8584	1500
page-blocks	**0.9193**	0.7888	0.9127	0.4034	0.8977	300

when the noise rate is increased 2%, 5%, 7%, 10%, 12% ,15% and 20%, the accuracies of ECBE only decrease 0.0088, 0.0194, 0.0339, 0.0315, 0.0456, 0.0491 and 0.0659 respectively and the changed magnitude of each accuracy is small, therefore it is known that ECBE has a good noise immunity. The good noise immunity of ECBE results from the elimination strategy; when the noise in the data stream is increasing, the difference between the entropies of the classification results and the actual results increases and the weights of classifiers decay at a fast speed. If the weights are below a predefined threshold, the classifiers which is trained by noise data will be deleted, therefore weak classifiers will not affect the classification of the new data block.

In order to further test the performance of ECBE, we compare ECBE with a representative neural network algorithm called OS-ELM on the 6 data sets and OS-ELM is tested with different activation functions. In the experiment, we choose *radbas, sigmoid, sine* and *hardlim* functions as the activation function. The test results are showed in Tables 7 and 8.

From Table 7, it can be seen that the accuracy of ECBE is better than OS-ELM on 3 data sets. When the activation function of OS-ELM is *sigmoid*, OS-ELM is better than ECBE on *waveform*, *Hyperplane* and *LED* data sets, but in fact, the advantage of OS-ELM is weak;

Table 8 Time overhead on data sets (Unit: s)

	ECBE	OS-ELM			
		radbas	sigmoid	sine	hardlim
waveform	**37.5227**	74.5529	72.5873	71.9789	72.6185
Hyperplane	**18.7280**	72.9289	74.4905	773.6481	73.6169
LED	**27.0643**	73.3113	72.6981	73.9757	72.5395
sensor_reading_24	0.6301	0.3586	**0.2652**	0.3588	0.3486
shuttle	**1.2246**	99.4194	24.1985	24.1958	24.1802
page-blocks	**0.4757**	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
winsize	100	100	100	1650

after calculating, we can know that the accuracies of OS-ELM are only more 0.0118, 0.0204 and 0.0905 than that of ECBE. In the aspect of time overhead, ECBE is better than OS-ELM on 5 data sets and OS-ELM is only better than ECBE on 1 data set. 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, therefore 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 concept drift, we select *Hyperplane*, *SEA*, *waveform* and *RBF* as experimental data sets which are generated by MOA. In consideration of concept drift in the data sets are unknown to us, therefore we reprocess *Hyperplane*, *SEA* and *waveform* data set and make the three data sets contain 5 concept drift in which the labels change from 1 to 2 or reverse direction changing. The size of the four data sets are 3000, 3000, 3000 and 50,000 respectively. The results are shown in Table 9 and Fig. 3.

From Fig. 3, we can see when concept drift happens, the accuracy of ECBE will degrade rapidly and ECBE will delete classifiers with weak performance, therefore 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 *Hyperplane* and *waveform* data sets, the algorithm has detected all concept drifts, but produced a error alert; on *SEA* data set, no error alert has produced, but a real changing is omitted; *RBF* is a gradual concept drift data set which is generated by MOA and the number of concept drift in the data set 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. Therefore in summary, it can include that although there are some deviations for concept drift detection, the experimental data indicates that the mechanism of ECBE detecting concept drift is effective and ECBE can cope with the two types of concept drift.

In order to test the influence of the parameters λ and β on the performance of ECBE, *waveform*, *Hyperplane* and *LED* are selected as the experimental data set and ECBE is executed with different λ and β values; *winsize*=2000. If λ is changed, β is set as $\beta = 1$. If β is changed, λ is set as $\lambda = 0.00002$. The results are showed as Figs. 4 and 5.

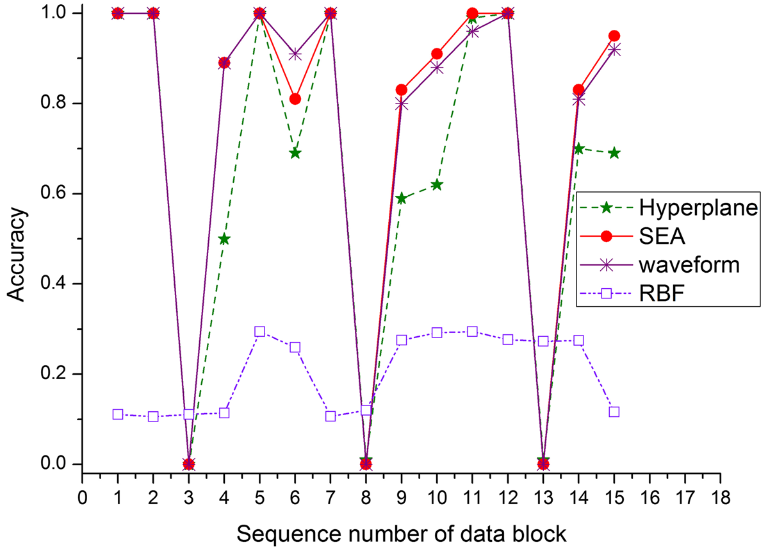

Fig. 3 Test result with different datasets of ECBE

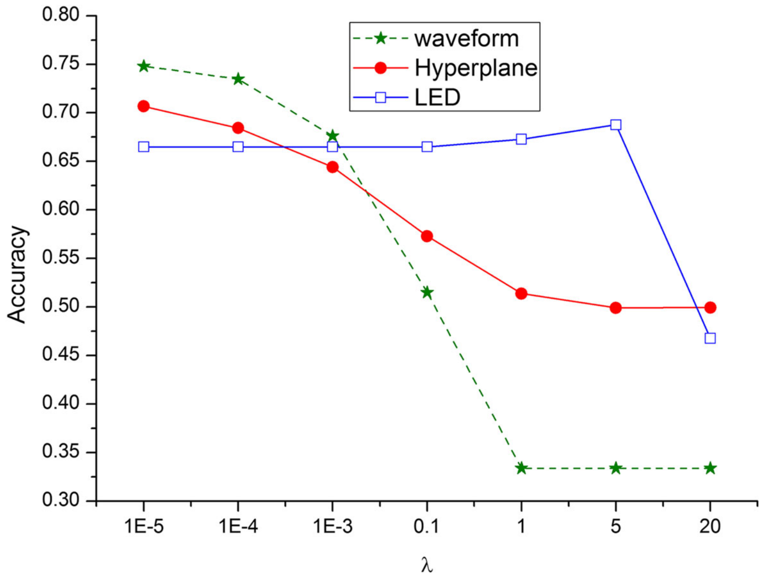

Fig. 4 The accuracies of ECBE with different λ values

From Fig. 4, it can be seen that the accuracy of ECBE changes with different λ values. On *waveform* and *Hyperplane* data sets, the accuracy decreases with λ increasing; when λ is greater than 1, the fluctuation of the accuracy is small. On *LED* data set, the accuracy of ECBE is slowly increasing with λ increasing; however, after λ is greater than 5, the accuracy sharply decrease. It can conclude that the optimal λ of ECBE is different on different data set. From Fig. 4, it is known that the trend of the accuracy of ECBE is first increasing and

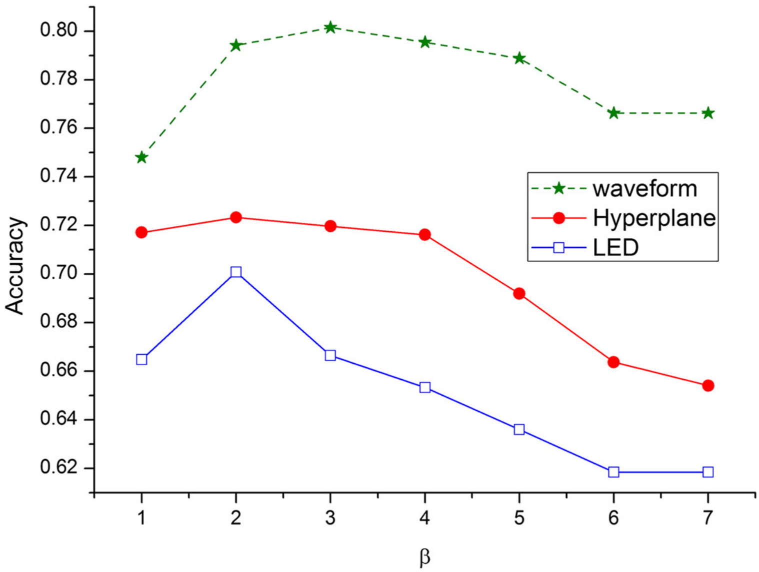

Fig. 5 The accuracies of ECBE with different β values

then reducing. On *Hyperplane* and *LED* data sets, the accuracy of ECBE is increasing when β is not greater than 2; affter β exceeds 2, the accuracy decreases with β increasing. On *waveform* data set, the optimal λ is 3. After β is greater than 6, the performance of ECBE is stable. It is ovious that the optimal β value is also different on different data set. How to determin the parameters λ and β needs the prior knowledge.

5 Conclusion

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

Acknowledgements This research was supported by the National Natural Science Foundation of China (Nos. 61772323, 61202018, 61432011, and U1435212), the National Key Basic Research and Development Program of China (973) (No. 2013CB329404), and the Natural Science Foundation of Shanxi Province, China (Nos. 201701D121051 and 201701D221098). 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. Abdulsalam H, Skillicorn DB, Martin P (2007) Streaming random forests. In: 11th International database engineering & applications symposium, pp 225–232

2. Becker H, Arias M (2007) Real-time ranking with concept drift using expert advice. In: ACM SIGKDD international conference on knowledge discovery & data mining, pp 86–94
3. Bifet A, Holmes G, Kirkby R, Pfahringer B (2010) Massive online analysis. J Mach Learn Res 11(2):1601–1604
4. Bifet A, Holmes G, Pfahringer B, Kirkby R (2009) New ensemble methods for evolving data streams. In: ACM SIGKDD international conference on knowledge discovery & data mining. ACM 2009, pp 139–148
5. Brzezinski D, Stefanowski J (2013) Reacting to different types of concept drift: the accuracy updated ensemble algorithm. IEEE Trans Neural Netw Learn Syst 25(1):81–94
6. Brzezinski D, Stefanowski J (2014) Combining block-based and online methods in learning ensembles from concept drifting data streams. Inf Sci 265(5):50–67
7. Czarnowski I, Jedrzejowicz P (2014) Ensemble classifier for mining data streams. Procedia Comput Sci 35(9):397–406
8. Domingos P, Hulten G (2000) Mining high-speed data streams. In: Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, pp 71–80
9. Domingos P, Hulten G (2001) A general method for scaling up machine learning algorithms and its application to clustering. In: Proceedings of the 18th international conference on machine learning, pp 106–113
10. Elwell R, Polikar R (2011) Incremental learning of concept drift in nonstationary environments. IEEE Trans Neural Netw 22(10):1517–1531
11. Escandell-Montero P, Lorente D, Martnez-Martnez JM, Soria-Olivas E, Martn-Guerrero JD (2016) Online fitted policy iteration based on extreme learning machines. Knowl-Based Syst. 100:200–211
12. Farid D, Li Z, Hossain A, Rahman C, Strachan R, Sexton G, Dahal K (2013) An adaptive ensemble classifier for mining concept drifting data streams. Expert Syst Appl 40(15):5895–5906
13. Gama J, Medas P, Rodrigues P (2005) Learning decision trees from dynamic data streams. In: Acm symposium on applied computing, pp 573–577
14. Gama J, Sebastiao R, Rodrigues P (2009) Issues in evaluation of stream learning algorithms. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 329–338
15. Gomes HM, Enembreck F (2013) Sae: social adaptive ensemble classifier for data streams. In: Computational intelligence & data mining, pp 199–206
16. Gu Y, Liu J, Chen Y, Jiang X, Yu H (2014) Toselm: timeliness online sequential extreme learning machine. Neurocomputing 128(27):119–127
17. Huang G, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B 42(2):513–529
18. Huang G, Zhu Q, Siew C (2005) Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE international joint conference on neural networks
19. Huang G, Zhu Q, Siew C (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
20. Kolter JZ, Maloof M.A (2005) Using additive expert ensembles to cope with concept drift. In: International conference on machine learning, pp 449–456
21. Kumar V, Gaur P, Mittal AP (2013) Trajectory control of dc servo using os-elm based controller. In: Power India conference, pp 1–5
22. Li P, Wu X, Hu X, Hao W (2015) Learning concept-drifting data streams with random ensemble decision trees. Neurocomputing 166(C):68–83
23. Liang NY, Huang GB, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17(6):1411–23
24. Lim J, Lee S, Pang H (2013) Low complexity adaptive forgetting factor for online sequential extreme learning machine (os-elm) for application to nonstationary system estimations. Neural Comput Appl 22(3–4):569–576
25. Liu D, Wu Y, Jiang H (2016) Fp-elm: an online sequential learning algorithm for dealing with concept drift. Neurocomputing 207:322–334
26. Ma Z, Luo G, Huang D (2016) Short term traffic flow prediction based on on-line sequential extreme learning machine. In: Eighth international conference on advanced computational intelligence, pp 143–149
27. Minku L, Yao X (2012) Ddd: a new ensemble approach for dealing with concept drift. IEEE Trans Knowl Data Eng 24(4):619–633
28. Ouyang Z, Min Z, Tao W, Wu Q (2009) Mining concept-drifting and noisy data streams using ensemble classifiers. In: International conference on artificial intelligence & computational intelligence, pp 360–364
29. Ramamurthy S, Bhatnagar R (2007) Tracking recurrent concept drift in streaming data using ensemble classifiers. In: International conference on machine learning & applications, pp 404–409

30. Rushing J, Graves S, Criswell E.e.a (2004) A coverage based ensemble algorithm (cbea) for streaming data. In: IEEE international conference on tools with artificial intelligence, pp 106–112
31. Rutkowski L, Jaworski M, Pietruczuk L, Duda P (2013) Decision trees for mining data streams based on the Gaussian approximation. IEEE Trans Knowl Data Eng 25(6):1272–1279
32. Ryang H, Yun U (2016) High utility pattern mining over data streams with sliding window technique. Expert Syst Appl 57(C):214–231
33. Shannon CE (1938) A mathematical theory of communication. Bell Syst Tech J 196(4):519–520
34. Street W (2001) A streaming ensemble algorithm (sea) for large-scale classification. In: ACM SIGKDD international conference on knowledge discovery & data mining, pp 377–382
35. Wang H, Yu P, Han J (2003) Mining concept-drifting data streams. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining, pp 226–235
36. Wei Q, Yang Z, Zhu J, Qiang Q (2009) Mining multi-label concept-drifting data streams using dynamic classifier ensemble. In: International conference on Fuzzy systems and knowledge discovery, pp 275–279
37. Wu X, Li P, Hu X (2012) Learning from concept drifting data streams with unlabeled data. Neurocomputing 92(3):145–155
38. Xu S, Wang J (2016) A fast incremental extreme learning machine algorithm for data streams classification. Expert Syst Appl 65:332–344
39. Xu S, Wang J (2017) Dynamic extreme learning machine for data stream classification. Neurocomputing 238:433–449
40. Yang Z, Wu Q, Leung C, Miao C (2015) OS-ELM based emotion recognition for empathetic elderly companion. Proceedings of ELM-2014, vol 2. Springer, Cham
41. Zhai J, Wang J, Wang X (2014) Ensemble online sequential extreme learning machine for large data set classification. In: IEEE international conference on systems, man and cybernetics, pp 2250–2255