Appl Intell (2012) 37:390–404
DOI 10.1007/s10489-011-0334-8

Data stream classification with artificial endocrine system

Li Zhao · Lei Wang · Qingzheng Xu

Published online: 19 January 2012

Abstract Due to concept drifts, maintaining an up-to-date model is a challenging task for most of the current classification approaches used in data stream mining. Both the incremental classifiers and the ensemble classifiers spend most of their time in updating their temporary models and at the same time, a big sample buffer for training a classifier is necessary for most of them. These two drawbacks constrain further application in classifying a data stream. In this paper, we present a hormone based nearest neighbor classification algorithm for data stream classification, in which the classifier is updated every time a new record arrives. The records could be seen as locations in the feature space, and each location can accommodate only one endocrine cell. The classifier consists of endocrine cells on the boundaries of different classes. Every time a new record arrives, the cell that resides in the most unfit location will move to the new arrived record. In this way, the changing boundaries between different classes are recorded by the locations where endocrine cells reside in. The main advantages of the proposed method are the saving of the sample buffer and the improving of the classification accuracy. It is very important for conditions where the hardware resources are very expensive or the main memory is limited. Experiments on synthetic and real life data sets show that the proposed algorithm is able to classify data streams with less memory space and classification error.

Keywords Data stream · Nearest neighbor classification · Artificial endocrine system (AES) · Hormone

1 Introduction

For classification algorithms, the two major problems on classifying a data stream are the infinite length and the concept drift [1–3]. The first one makes the traditional multi-pass classification algorithms incapable of classifying a data stream for their requirement of infinite storage and large amount of training time [4–6]. The second one makes the most static stream classification algorithms incapable of classifying a data stream with concept drifts for the underlying changes occurred in the stream. For a time-changing data stream, an incremental updating manner of the classifier is very important. A temporal model is used to capture the evolutions of the stream.

In general, the classification process is always accompanied by the course of model construction and test [7–9]. The classification model keeps changing with the progression of the stream. If a static classifier is used to classify an evolving data stream, the accuracy of it will drop greatly. For a sudden burst of concept drift in a time-changing stream, an up-to-date model always provides better accuracy. But for relatively stable time-changing streams, models built with long-term samples will be great. Both of the short-term and long-term behaviors of the stream are important for a classifier. It is decided by the stream itself and can not be known a priori. In some algorithms, rule-based stream classification methods have been proposed in which a temporary window of samples is needed and the updating process of the rule

L. Zhao · L. Wang · Q. Xu
School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

L. Wang
e-mail: wanglei_xaut@163.com

L. Zhao (✉)
Department of Information Engineering, ShiJiaZhuang Vocational Technology Institute, ShiJiaZhuang 050081, China
e-mail: zhaoli_xaut@163.com

set is very complex [10]. Decision trees have also been proposed to classify a data stream, but the structure of it is very unstable. A slight change of the data distribution will trigger substantial changes of the tree [11, 12]. Nearest neighbor classifier is more suitable for an evolving environment, but it is too expensive to keep the whole data samples or to condense the samples according to the changing streams [13].

The artificial endocrine system (AES) is a model that simulates the manner of information processing in biological endocrine systems [14–17]. It allows cells in a large system to communicate and interact with each other forming an entire system. The features of self-organizing and self-repairing are shown without unique identifiers and complex control strategies in this model. The autonomous decentralized system (ADS) is perhaps the first hormone-inspired system for building an online maintenance, robustness and flexible mechanism. And it was widely used in various systems to control trains, water supplies, and multistage dams. The theoretical foundation of it was firstly proposed by Kravitz [18]. Then Avila-Garcia and Canamero proposed a regulation model of hormones in which the internal and external stimuli are considered comprehensively to choose an appropriate behavior [19]. A digital hormone model (DHM) was also proposed by Shen in which the Turing's reaction-diffusion model, stochastic reasoning and action, self-organization, and distributed control are integrated into one [20]. Now, the artificial endocrine systems have been widely used in heterogeneous processing system, decoupling control, and multiprocessor system control.

In this paper, we proposed a nearest neighbor rule based method for data stream classification, in which an artificial endocrine system (AES) was used for updating and condensing the samples in the classifier. The classifier can efficiently work in both static and evolving environments. The classification model could be updated with not a whole data block, but a new arrived record. And the big data buffer of samples for training the classifier is no longer needed. It is very important for the conditions where the hardware resources are very expensive or the main memory is limited. At the same time, the classifier can be updated immediately when a new record arrived. This will improve the classification accuracy and decrease the burst of classifying error.

The rest of the paper is organized as follows. We first review the related work on the data stream, nearest neighbor classification and artificial endocrine system. Secondly, we represent the artificial endocrine model based approach for building nearest neighbor classifier and give some theorem analysis. Then we describe several experiments in which parameters of the algorithm are examined. With these experiments, we demonstrate how to use the algorithm to build an effective and efficient classifier. Finally, we conclude this paper by highlighting the key contributions of this work.

2 Related work

2.1 K nearest-neighbor classification

The K nearest neighbor (NN) classification approach is a useful and simple method, the object of which is to find the K nearest neighbors of a given query. Then, predict the class label of the query with the k nearest neighbors [21, 22]. Nearest neighbor (NN) classification is a nonparametric method for pattern classification, which makes no assumption on probability distribution and avoids the difficulties involved with the determination of multidimensional conditional probability densities [23]. So many works has been done within this area [24–26].

Assume there are M pattern classes, numbered $1, 2, \ldots, M$. Let each pattern be defined in an N-dimensional feature space. Given K training patterns, each of them is a pair (x_i, c_i), $1 \le i \le K$ and $c_i \in \{1, 2, \ldots, M\}$ denotes the correct pattern class and $x_i = (x_{1i}, x_{2i}, \ldots, x_{Ni})$ is the set of feature values for a pattern. Let $\mathrm{TNN} = \{(x_1, c_1), (x_2, c_2), \ldots, (x_K, c_K)\}$ be the nearest neighbor training set. Given an unknown pattern x, the decision rule is to decide x is in class c_j if

$$d(x, x_j) \le d(x, x_i), \quad (1 \le i \le K, 1 \le j \le K) \tag{1}$$

where d is some N-dimensional distance metric. In this paper, let d be the Euclidean distance between points p and q. Thus, we can get,

$$\begin{aligned} d(p, q) &= d(q, p) \\ &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \end{aligned} \tag{2}$$

Actually, the preceding rule is called the 1-NN rule for only one nearest neighbor was used. A generalization of this is the k-NN rule, which takes the k nearest patterns and decides upon the pattern class that appears most frequently.

2.2 Artificial endocrine system

Based on hormone inspired methodology, Ihara [27] proposed an autonomous decentralized system (ADS), in which a content code communication protocol was used to communicate by a semantic based system. Then, Shen [28] described a digital hormone model (DHM) in which the advantages of stochastic reasoning, distributed control, self-organization, and Turing's reaction-diffusion model were integrated. And the hormone model was also used by Mendao [29] to coordinate the completion from different tasks by an autonomous robot. In his presentation, tasks were de-

fined as glands and hormones can be released by them at a fixed speed. Once the hormones are greater than a given threshold, they are released and disseminated over the network environment. Walker [30] expanded the model for task assignment to a multiple-autonomous-robots environment. Each task was simply assigned to a robot and each robot can choose a suitable response to the external environment. Nowadays, artificial endocrine systems are widely used in multiprocessor system control, real time task allocation, human-machine communication and management systems in autonomic networks.

In previous work [31], we have proposed a lattice based artificial endocrine system (LAES), in which the system was abstracted as a quintuple model, LAES = (Ld, EC, TC, H, A).

(1) The environment space Ld denotes a bounded area in which all the LAES elements live, die, and communicate within it. "d" is the number of dimensions in the space.
(2) The endocrine cell EC is the fundamental component of the proposed model. Each endocrine cell has a perceiving sensor (AE) and a releaser (BE). The sensor AE can perceive the hormone concentrations of the location it lies in. The releaser BE can release a certain quantity of hormone in a discrete manner. In addition, endocrine cells were divided into different classes. Each kind of endocrine cell can release a special kind of hormone.
(3) The target cell TC is an organ or cell that can accept stimulation from endocrine cells. The receptor AE of cell TC can perceive the hormone concentrations of the location where TC lies.
(4) Hormones are bio-active substances which can be used to adjust the activities of TC and EC. The cumulative hormone concentration at location L_{xy} is equal to the sum of the concentrations coming from different endocrine cells.
(5) Algorithm A is an iterative procedure with which all elements of LAES carry out their assigned task.

2.3 Data stream classification

The problem of data stream classification can be mathematically described like this:

A data stream is a continuous sequence of samples: $\{x_1, \ldots, x_{\text{now}}\}$, where each x_i, is a d-dimensional feature vector. x_{now} is the newly arrived sample. Each sample x_i is associated with a class label c_i and a sequence label t_i, respectively. Let $t_{i+1} = t_i + 1$ and $t_1 = 1$. Given a latest sample x_{now} (the class label is unknown), the task for a classification algorithm is to predict the class label of it before the next sample $x_{\text{now}+1}$ arrives.

Many algorithms have been proposed in this field [32, 33]. Most of them require very large amounts of samples for training a classifier and are based on a learning manner of batch, which makes them incapable of dealing with concept drift effectively and efficiently [34–36].

Domingos [11] introduced an incremental decision tree algorithm, VFDT, in which Hoeffding bounds was used to guarantee that the output of the classifier is asymptotically nearly identical to that of a batch learner. However, the algorithm is based on the assumption of stationary distribution of the samples. But most of current data streams violate this assumption. So, Hulten [12] introduced the CVFDT system to deal with time-changing data streams, which works by efficiently updating a decision tree with a window of examples. To further improve the classification accuracy of the classifier based on decision tree, Abdulsalam [2] proposed an algorithm which combines ideas of decision trees and Random Forests. Xu [37] proposed a clustering feature decision tree model, CFDT, in which a micro-clustering algorithm was used to provide the statistical summaries of the data for incremental decision tree induction. However, the problem all data stream classification algorithms built with decision trees must face is that even a slight change of the given concept may lead to substantial changes of the built tree. The learning and classifying efficiency of these approaches must be carefully thought over. Rule-based approaches have also been proposed to classify the time-changing data streams [10]. In these algorithms, some data structures were firstly constructed to contain all valid rules of a temporary window. And then, the classifier was updated by inserting or deleting some rules from the data structure to keep changing with the concept drifts. However, to accurately calculate the support and confidence of a rule, the algorithms have to maintain an appropriate window and the classifier requires rapid variation due to concept drifts. In this respect, nearest neighbor classifier is more feasible for time-changing data streams because of their simplicity. However, it is too expensive to keep all of the samples in a classifier. In fact, many methods have been developed to reduce the size of the sample set while maintaining the advantages of the nearest neighbor classifier. But most of them can not be used directly for classifying a time changing data stream. So, Masud proposed a data stream classification method [2], in which different classes of data can be distinguished and the emergence of a novel class can be detected effectively. In this algorithm, k-nearest neighbor classifier was used to classify a new unlabeled sample. To make the classifier more efficient, semi-supervised clustering technique was used to build K clusters with the training data. Aggarwal [7] developed an on-demand classifier by adapting micro-clustering model to the classification problem, in which the micro-clusters and class statistics were used in conjunction with a nearest neighbor classification

process in order to classify a un-label sample. In the two algorithms above, cluster technology was used to simplify the structure of the classifier. But the clustering process of the algorithms was affected by the size of the window and the maintenance process of the class statistic was very complex. So, in this paper, we proposed an AES based sample condensing method for classifying a data stream. The classifier can be updated immediately every time a new record arrives. This will make the classifier more effective and efficient.

3 Data stream classification with artificial endocrine system

The AES algorithm proposed in this paper remains only the samples on the boundaries between different classes and the amount of the boundary samples K_{total} is decided a prior. Interior samples (points) in a class are seen as unimportant points which should be discarded in this process. The feature space of records is defined as environment Ld. Each record is seen as a position that can accommodates only one artificial endocrine cell. K_{total} endocrine cells are initialized in the initial stage. With the evolution of the data stream, the endocrine cells keep on changing their positions to track the changing boundaries between different classes (concept drifts).

Each endocrine cell has a perceiving sensor and a releaser. Only one kind of hormone can be released by an endocrine cell $\text{EC}_{i,k}$ and the class of the hormone released was decided by the position where the cell $\text{EC}_{i,k}$ resided in. Each endocrine cell can perceive all kinds of hormones coming from other endocrine cells. When the same kind of hormones is perceived by an endocrine cell, the hormone concentration will be accumulated. On the contrary, when different kinds of hormones are perceived, the concentration will be reduced.

Assume there are M record classes, numbered 1, $2, \ldots, M$. Let each record be defined in an N-dimensional feature space. Given a data stream of records, each of them is a pair (x_i, c_i), $c_i \in \{1, 2, \ldots, M\}$ denotes the record class. Let $\text{EC}_{i,k}$ be the kth endocrine cell whose class label is "i", $\text{Ld}(\text{EC}_{i,k})$ be a sample on the boundaries where the endocrine cell $\text{EC}_{i,k}$ resides in, $\text{C}(\text{EC}_{i,k})$ be the label of the sample where endocrine cell $\text{EC}_{i,k}$ resided in. Let $x_{i,k} = (x_{i,k,1}, x_{i,k,2}, \ldots, x_{i,k,N})$ be the set of feature values for a record $\text{Ld}(\text{EC}_{i,k})$.

Definition 1 (Hormone concentration $\text{H}_{i,k-i'k'}$) The hormone concentration $\text{H}_{i,k-i'k'}$ is the perceived hormone concentration by the endocrine cell $\text{EC}_{i,k}$, which was secreted by endocrine cell $\text{EC}_{i',k'}$. When the class labels of the two endocrine cells are the same, $\text{C}(\text{EC}_{i,k}) = \text{C}(\text{EC}_{i',k'})$, and $\text{EC}_{i,k} \neq \text{EC}_{i',k'}$, it can be expressed like this:

$$\text{H}_{i,k-i',k'} = \frac{a_1}{\sqrt{(x_{i,k,1} - x_{i',k',1})^2 + (x_{i,k,2} - x_{i',k',2})^2 + \cdots + (x_{i,k,N} - x_{i',k',N})^2}} \tag{3}$$

When the class labels of the two endocrine cells are different, $\text{C}(\text{EC}_{i,k}) \neq \text{C}(\text{EC}_{i',k'})$, and $\text{EC}_{i,k} \neq \text{EC}_{i',k'}$, it can be expressed like this:

$$\text{H}_{i,k-i',k'} = \frac{-a_2}{\sqrt{(x_{i,k,1} - x_{i',k',1})^2 + (x_{i,k,2} - x_{i',k',2})^2 + \cdots + (x_{i,k,N} - x_{i',k',N})^2}} \tag{4}$$

where "a_1" and "a_2" be a decay constant initialized in the beginning of the algorithm, "$x_{i,k,1}, x_{i,k,2}, \ldots, x_{i,k,N}$" and "$x_{i',k',1}, x_{i',k',2}, \ldots, x_{i',k',N}$" are the feature vectors of the records where endocrine cell $\text{EC}_{i,k}$ and $\text{EC}_{i',k'}$ resided in. "$\text{EC}_{i,k} \neq \text{EC}_{i',k'}$" indicates that the endocrine cell $\text{EC}_{i,k}$ can not perceive the hormone secreted by itself. When "a_1" and "a_2" were set to 1, the hormone concentration is just the inverse of the Euclidean distance measure commonly used by the k-NN algorithm.

Definition 2 (Total hormone concentration $\text{H}_{i,k}$) The total hormone concentration $\text{H}_{i,k}$ is the perceived hormone concentration by the endocrine cell $\text{EC}_{i,k}$, which is secreted by all of the other endocrine cells. It can be expressed like this:

$$\begin{aligned}\text{H}_{i,k} = &\sum_{k'=1;k\neq k'}^{K_i} \frac{a_1}{\sqrt{(x_{i,k,1} - x_{i,k',1})^2 + (x_{i,k,2} - x_{i,k',2})^2 + \cdots + (x_{i,k,N} - x_{i,k',N})^2}} \\ &+ \sum_{i'=1;i\neq i'}^{M} \sum_{k'=1}^{K_{i'}} \frac{-a_2}{\sqrt{(x_{i,k,1} - x_{i',k',1})^2 + (x_{i,k,2} - x_{i'k',2})^2 + \cdots + (x_{i,k,N} - x_{i'k',N})^2}}\end{aligned} \tag{5}$$

Fig. 1 Pseudocode for algorithm DELETE_NOISE

```
Algorithm DELETE_NOISE
Input: P: the population of endocrine cells, EC_{i,k} (0<i≤M;0<k≤K_i);
Output: P`: the population of endocrine cells, EC`_{i,k`} (0<i≤M;0<k`≤K_i-1));
1  let EC_{i,k} (0<i≤M;0<k≤K_i) be an endocrine cell, Ld(EC_{i,k}) be the position
   it resided in, C(EC_{i,k}) be the class label of it, H_threshold is a threshold of
   hormone concentration, ΔH_{Ld(ECi,k), r*} be the variation of hormone
   concentration caused by noise r*;
2  H_MIN=10000: k`=0: i`=0
3  for i = 1:M do
4      for k = 1:K_i do                          //to find a record Ld(EC_{i`,k`})
5          if H_MIN> H_{Ld(ECi,k)} then          //which has the minimal
6              H_MIN= H_{Ld(ECi,k)}              //hormone concentration.
7              k`=k: i`=i
8          endif
9      endfor
10 endfor
11 if H_{Ld(ECi`,k`)} < H_threshold then          //noise judgement
12     r*= Ld(EC_{i`,k`})
13     for i = 1:M do                            //eliminate the disturbances of
14         for k = 1:K_i do                      //the hormone concentrations
15             H_{Ld(ECi,k)} = H_{Ld(ECi,k)} - ΔH_{Ld(ECi,k), r*}   // coming from noise.
16         endfor
17     endfor
18     Ld(EC_{i`,k`}) =Ld(EC_{i`,Ki})            //discard the noise from
19     K_i =K_i-1                                //populations of endocrine cells.
20 endif
```

3.1 Algorithm design

The proposed algorithm contains two phases. In the first phase, the total K_{total} endocrine cells is initialized by the first K_{total} records. In the second phase, when a new record (target cell) arrived, if the class label of it is unknown, the classifier calculates the distance between it with all the positions where endocrine cells lived in. The class label of the nearest position will be assigned to the new record. On the contrary, if the class label of it is known, the classifier will adjust the model to make it more suitable for the changes of data streams (concept drifts). The main algorithm in the second phase was made up of four modules: DELETE_NOISE, CLASSIFICATION, DETECT_CONCEPT_DRIFT and UPDATE.

DELETE_NOISE Module DELETE_NOISE was used to delete noise records in classifier. According to formula (5), if the hormone concentration perceived by an endocrine cell is far less than zero, we believe that the location (record) it resides in is a noise point. So, it should be deleted. In fact, if the hormone concentration is far greater than zero, the point should be an inner point in a class. We can see this from Fig. 8 and Table 1. The pseudocode for algorithm DELETE_NOISE is list in Fig. 1.

CLASSIFICATION Module CLASSIFICATION was used to assign the new arrived record a class label. In this process, classifier calculates the distances between the new arrived record and all of the endocrine cells. The class label of the nearest position (record) where the endocrine cell resided in will be set to the new record. The classification error is calculated every time a labeled record arrived.

DETECT_CONCEPT_DRIFT Module DETECH_CONCEPT_DRIFT was used to detect whether or not a concept drift takes place. If the classification error is greater than a threshold, Δerror, we believe that a concept drift takes place and the classifier should be updated. If a concept drift was detected and the time label of a record in which an endocrine cell resided was too old, the record will be deleted from the classifier and the endocrine cell will move to a new arrived record.

UPDATE Module UPDATE was used to update a classifier to make it more suitable for tracking a time-changing stream. The algorithm keeps on checking the classifier to find a non boundary record and to discard it. We represented two kinds of updating methods in this paper. The first one, called UPDATE-1, is in an ensemble update manner, in which all of the artificial endocrine cells re-calculate their hormone concentrations every time a new labeled record arrives. We can see it from Fig. 2. The second one, called UPDATE-2, is in an incremental update manner, in which every endocrine cell only accumulatively calculate the hormone changes caused by the new arrived record and the record discarded. The two updating manner are very different, so as to far difference in the updating time. We can see it from Figs. 10–12. The whole procedures of the two update methods were described in Fig. 2 and Fig. 3.

3.2 Algorithm analysis

Data stream classification algorithm must be online and the arrived record must be examined in amortized time. In fact,

Fig. 2 Pseudocode for algorithm UPDATE-1

```
Algorithm UPDATE-1
Input: r_n: the new arrived record;
Output: r* : the unfit position (record) where an endocrine cell has resided in;
1  let EC_{i,k} (0<i ≤ M, 0<k≤ K_i) be an endocrine cell, Ld(EC_{i,k}) be the position
   it resided in, C(EC_{i,k}) be the class label of it, T(EC_{i,k}) be the time label of it;
2  while r_n ≠ Ø do
3     for i = 1:M do
4        for k = 1:K_i do
5           Compute H_{Ld(ECi,k)}          // calculate the total hormone
6        endfor                            // concentration perceived by EC_{i,k} ;
7     endfor
8      H_MAX= -10000                       // initialize H_MAX to a small value;
9     for i = 1:M do
10       for k = 1:K_i do
11          if H_MAX< H_{Ld(ECi,k)} then
12             H_MAX= H_{Ld(ECi,k)}        //to find a positon with the highest
13             r*= Ld(EC_{i,k})            //hormone concentration;
14             i`=i;  k`=k
15          endif
16       endfor
17    endfor
18    Ld(EC_{i`,k`}) =r_n
19 endwhile
```

Fig. 3 Pseudocode for algorithm UPDATE-2

```
Algorithm UPDATE-2
Input: r_n: the new arrived record;
Output: r* : the unfit position (record) where an endocrine cell has resided
in;
1  let EC_{i,k} (0<i ≤ M, 0<k≤ K_i) be an endocrine cell, Ld(EC_{i,k}) be the position
   it resided in, C(EC_{i,k}) be the class label of it, T(EC_{i,k}) be the time label of
   it, ΔH_{Ld(ECi,k), r_n} be the variation of hormone concentration caused by r_n,
   ΔH_{Ld(ECi,k), r*} be the variation of hormone concentration caused by r*;
2  while r_n ≠ Ø do
3     i`=C(r_n): H_MAX=-10000
4     for k = 1:K_i · do                  //to find an inner record r*
5        if H_MAX< H_{Ld(ECi`,k)} then    //which will be deleted in the
6           H_MAX= H_{Ld(ECi`,k)}         //next step.
7           k`=k
8        endif
9     endfor
10       r*=  Ld(EC_{i,k`})
11    for i = 1:M do                      //calculate the variations of the
12       for k = 1:K_i do                 //hormone concentrations
13          H_{Ld(ECi,k)} = H_{Ld(ECi,k)} - ΔH_{Ld(ECi,k), r*}   // caused by r* and r_n.
14          H_{Ld(ECi,k)} = H_{Ld(ECi,k)}+ΔH_{Ld(ECi,k), r_n}
15       endfor
16    endfor
17    Ld(EC_{i`,k`}) =r_n
18 endwhile
```

for most real-life data streams, the arrival rates are very high, so the complexity of the classification algorithm must be low. In our AES algorithm, the time it takes to update a classifier depends on: (1) the time to detect and delete a noise record in the classifier, (2) the time to classify a sample and to calculate the classification accuracy of it, (3) the time to detect whether or not a concept drift takes place, (4) the time to search the most useless record in the classifier and to replace it by the new arrived record.

The time to update the classifier is the sum of times cost by the four modules. In module DELETE_NOISE, the time to find a noise depends on finding an endocrine cell, the hormone concentration of which is the minimum, and on recovering the hormone concentration disturbed by the noise. The number of the endocrine cells are K_{total}, so the time complexity for delete a noise is $O(2 \times K_{\text{total}})$.

The time cost in module CLASSIFICATION depends on calculating the distances between the new arrived point and

all kinds of endocrine cells and on calculating the error rate of the classifier. The time complexity of it is $O(K_{\text{total}})$.

The time cost in module DETECT_CONCEPT_DRIFT depends on judging whether or not a concept drift appears and on recovering the hormone concentrations which contained the contribution coming from the outdated endocrine cells that will be discarded. The time complexity of it is $O(K_{\text{total}})$.

There are two kinds of updating manners in the module UPDATE. For the first one, the time depends on recalculating all the hormone concentrations of the endocrine cells appeared in the classifier and on finding a maximum one. According to formula (3)–(5), the time complexity of it is $O(2 \times K_{\text{total}})$. For the second updating manner, the time depends on finding the most useless inner record r^* and on calculating the concentration variations coming from the new arrived record r_n and the record r^*. The time complexity of it is $O(K_{i'} + K_{\text{total}})$. The difference of the two time complexities is $O(K_{\text{total}} - K_{i'})$.

4 Experiments

The experiments were conducted on a 3.0 GHz Pentium 4 with 1.0 GB of memory running Microsoft Windows XP. All code was compiled using Microsoft Visual C++ 6.0. In order to examine the performance of the proposed algorithm, we carried out several experiments on synthetic and real datasets. The AES algorithm can deal with ordinal or numerical attributes, the ranges of which are known.

4.1 Data sets and parameters

4.1.1 Dataset-1

In dataset-1, we produced three kinds of records, each of them can be expressed by pairs (x_i, c_i), $c_i \in \{1, 2, 3\}$, $x_i = (x_{i,1}, x_{i,2})$. Parameter "a" is a constant and the value of it was set to 600. The ranges of variables $x_{i,1}$ and $x_{i,2}$ are 0 to 600. The dataset-1 can be described as following:

$$
\begin{aligned}
&x_{i,1} = a \times \text{Rnd}() \\
&x_{i,2} = a \times \text{Rnd}() \\
&c_i = 1, \quad \text{if } \sqrt{(x_{i,1} - 400)^2 + (x_{i,2} - 400)^2} \le 100 \\
&c_i = 2, \quad \text{if } x_{i,1} \in [100, 300] \text{ and } x_{i,2} \in [100, 300] \\
&c_i = 3, \quad \text{otherwise.}
\end{aligned}
$$

4.1.2 Dataset-2

We randomly produced a data stream with 1000 sample pairs (x_i, c_i), $c_i \in \{1, 2\}$, $x_i = (x_{i,1}, x_{i,2})$, $i \in \{1, 2, \ldots, 1000\}$. Parameter "$a$" is a constant. Let $a = 600$. There are two kinds of records in this dataset and the ranges of variables $x_{i,1}$ and $x_{i,2}$ are 0 to 600. In this stream, concept shifts take place every 100 records. The dataset-2 can be described as following:

$$
\begin{aligned}
&x_{i,1} = a \times \text{Rnd}() \\
&x_{i,2} = a \times \text{Rnd}() \\
&c_i = 1, \quad \text{if } x_{i,2} \ge (i \backslash 100 + 1) * 100 \\
&c_i = 2, \quad \text{otherwise.}
\end{aligned}
$$

4.1.3 Dataset-3

We get the dataset Nursery from UCI machine learning repository [38], which consists of 12960 samples in 5 classes with 8 attributes. The original set is repeatedly queued to produce a stream for classifying. To make concept changes, we randomly select an attribute for every 12000 records and change its values in a circular way like this "$a_1 \to a_2 \to \cdots \to a_n \to a_1$".

4.1.4 Dataset-4

Let $\sum_{i=1}^{d} a_i x_i = a_0$ be a hyper-plane in a d-dimensional space. Records satisfying $\sum_{i=1}^{d} a_i x_i < a_0$ were labeled "0", otherwise, they were labeled "1". We use the changes of hyper-plane to simulate the concept drifts [12, 39]. We generate random points in the d-dimensional space. Parameters "a_i" were initialized with values in [0, 1], and parameter a_0 was set to $\sum_{i=1}^{d} a_i/2$. We randomly generate noise points by switching the class labels of examples.

4.1.5 Dataset-5

We get the dataset Cover Type from UCI machine learning repository. There are 7 classes, 54 attributes, and 581012 instances in the dataset which contains some geospatial information about several forests. No missing values were found. So, we directly use it in our experiments after normalization. We randomly select several attributes for every 12000 records and change its values in a circular way like Dataset-3.

4.1.6 Dataset-6

We get the dataset Letter Recognition from UCI machine learning repository. There are 26 classes, 16 attributes, and 20000 instances in this dataset. No missing values were found. The original set is repeatedly queued to produce a stream for mining. To make concept changes, we randomly select several attributes for every 80000 records and change its values in a circular way like Dataset-3.

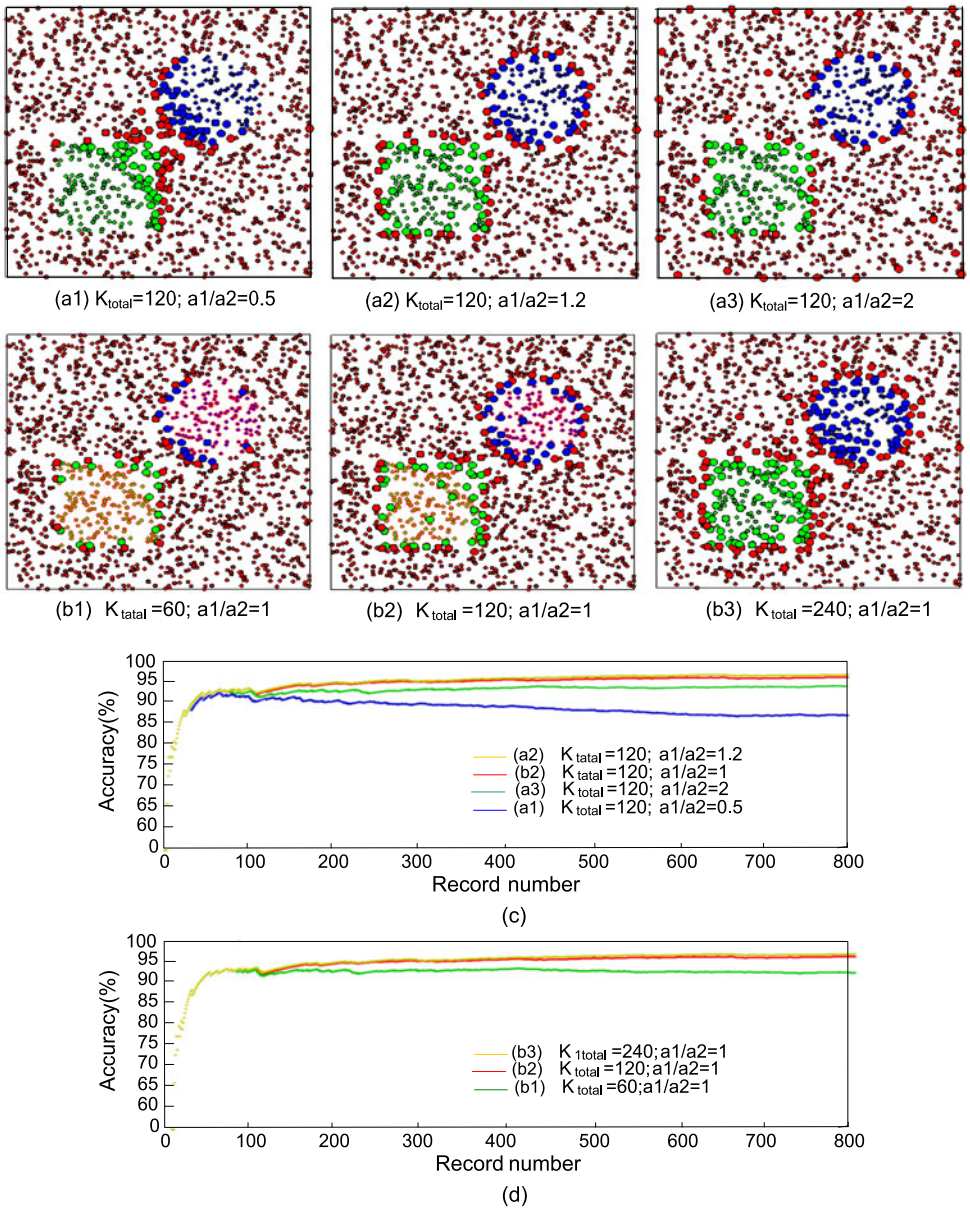

Fig. 4 Effects of different population sizes

4.2 Parameters study

In this section, we study the effects of parameters a_1 and a_2. The data set used is dataset-1. There are 800 sample records and 3 classes in it. When parameter "a_1/a_2" was set to 0.5, 1, 1.2 and 2, we indicate the boundary records contained in a classifier with the big circles, in which the red ones are boundary records in class "3", the green ones are boundary records in class "2", and the blue ones are boundary records in class "1". (The small circles in red, green and blue color are inner records in class "3", "2" and "1".) From Fig. 4(a1)–(a3), we can see that when the parameter "a_1/a_2" was set to a large value "2" or a small value "0.5", the obtained cells are not the boundary cells. In fact, the parameter "a_1/a_2" can be used to adjust the effectiveness of hormones coming from different kinds of cells. "$a_1/a_2 = 1$" means different hormones have the same affections on a cell. In general, the value span of it should to set to [1–1.5].

From Fig. 4(b1)–(b3), we can also find that when the population size of endocrine cells was set to a small value "60", many boundary records can not be tracked by this classifier. Thus, the classification accuracy of it was very poor. On the contrary, when the population size was set to a very big one "240", a lot of inner points were contained in the classifier. Thus, the efficiency of the classifier will be very low. The population scale should be decided by the complexity of the boundaries between different classes.

We further examined the accuracy of the classifier. In Fig. 4(c), we set the total scale of the cells to 120. We get

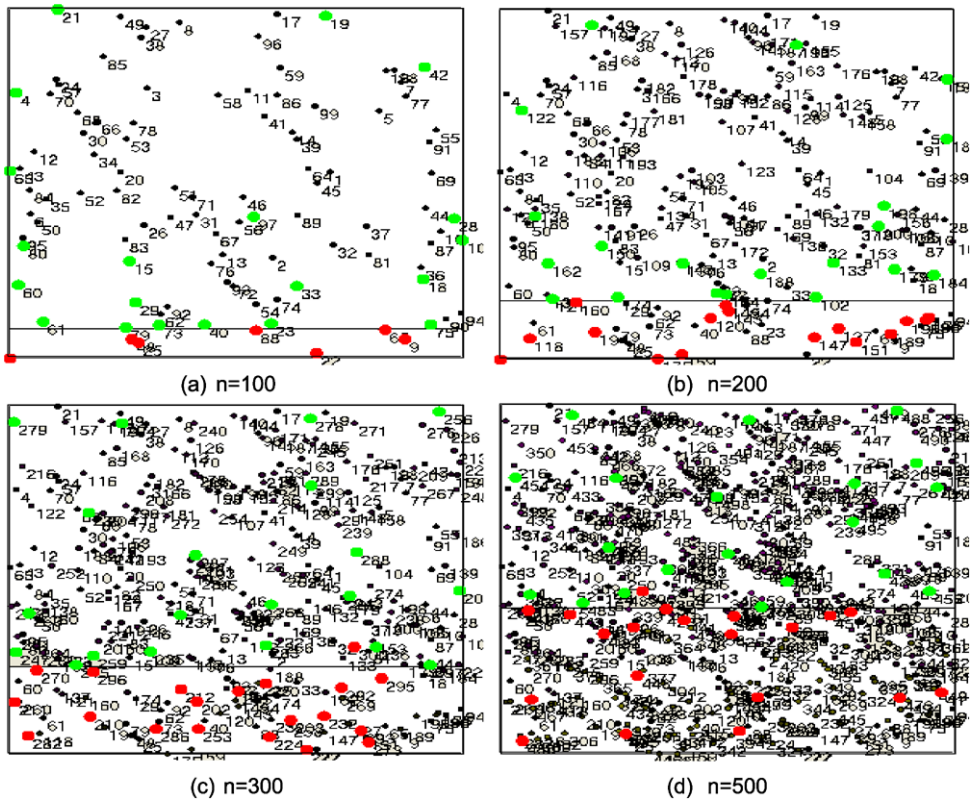

Fig. 5 Dealing with concept drifts

the best accurate classifier when parameter "a_1/a_2" was set to "1.2". What was the reason? It is because that when parameter "a_1/a_2" was set to "1" different hormones have the same affections on a cell so as to unbalanced distribution of the boundary cells. So, when there are two or more classes of records appeared in a classifier, the value of parameter "a_1/a_2" should be greater than "1". In Fig. 4(d), the parameter "a_1/a_2" was set to 1. It is very obvious that, with the increasing of the cell number the classification accuracy of the classifier keeps improving.

4.3 Dealing with concept drifts

In this section, we show the updating process of a classifier in the process of handling concept drifts. The data set we used is Dataset-2. In the synthetic data set, the class label was decided by parameter $x_{i,2}$. We introduce concept drifts by moving the boundaries for every 100 records. No noise records were introduced into this experiment. Every time a new record arrives, the algorithm checks the time interval between the oldest record and the new arrived record. If the time interval is greater than the threshold T_{update} and the accuracy of the classifier is less than the accuracy threshold T_{accuracy}, the oldest record in the classifier will be deleted. When the 100th, 200th, 300th, and 500th record arrive, we show the classifiers and the classification boundaries (black line) in Figs. 5(a), (b), (c), and (d). The results were obtained with parameters $K_{\text{total}} = 60$, $T_{\text{accuracy}} = 95\%$, $T_{\text{update}} = 100$.

In order to further show the effectiveness of different update interval and to simulate the real data stream environment, we generated more records with the function used in dataset-2 and introduce a concept drift by moving the boundary for every 10000 records. No noise records were introduced in this experiment and the classification accuracy was calculated every 100 records. The update interval T_{update} was set to 100, 1000, and 10000. Other parameters were the same as the former experiment.

From Fig. 6, we can see that with the increasing of the update interval T_{update} the classification accuracy got more and more poor. Every time the concept drift takes place, a burst of classification error will appear. After the burst, the accuracy of it will go back to a high level. It can be seen from (a) and (b). However, when the frequency of the update interval T_{update} exceeds/reaches the frequency of concept drift, in Fig. 6(c), the classification accuracy of the classifier will get worse and worse. It can no longer go back to a high level. It can also be seen from Fig. 6(d), that without the accuracy threshold, too fast updating frequency will make the classifier more unstable and the classification accuracy will drop simultaneously. So, to get an accurate classifier, the frequency of the update interval should be greater than that of the concept drifts.

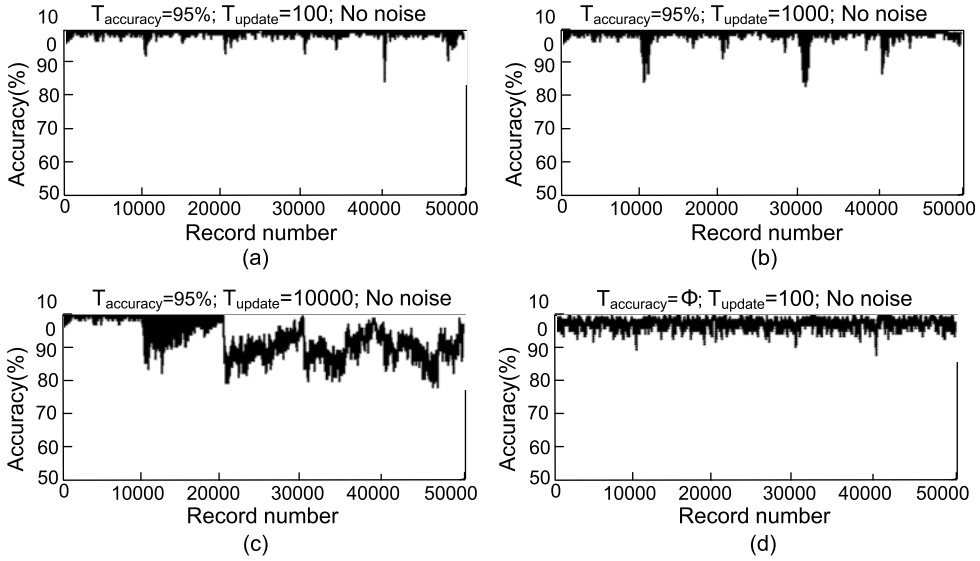

Fig. 6 Different update intervals

4.4 Detecting and deleting noise

In fact, most of the real-life data streams have noise records. In this section, we show how the proposed algorithm deals with noise. Firstly, we introduced noises by randomly switching the labels of 20 percent of the examples generated by functions in dataset-2, and totally 50000 sample records were generated. Then, we recorded the state of the classifier and the data stream. When the first 10000, 20000, and 30000 samples arrived, we separately wrote down the samples and it is shown in Fig. 7(a), (b), and (c). In the three pictures, the green points are records in class 2 and the red ones are records in class 1. The boundary between the two classes is very clear and it keeps changing with the increasing of the record number. What should be noticed is that the separate points which hided in many other kinds of samples are noise samples.

In Fig. 7(d)–(l), we shown the endocrine cells remained in the classifier when the first 10000, 20000, and 30000 records arrived. Threshold $H_{threshold}$ was set to Φ, 0, and -0.05. $H_{threshold}$ is a threshold of hormone concentration, which was used to delete records in a classifier. "$H_{threshold} = \Phi$" means the parameter $H_{threshold}$ is not used in this algorithm so as to no noise record will be discarded. $H_{i,k}$ is the hormone concentration of the record that accommodates endocrine cell $EC_{i,k}$. When the value of the parameter $H_{i,k}$ is greater than 0, a large value means that the record is an inner point in a record class. On the contrary, when the value of the parameter $H_{i,k}$ is less than 0, the small value means that the record is a noise record which should be discarded. Having met the first 10000 records, the endocrine cells remained in the classifier were shown in Fig. 7(d) and their hormone concentrations were shown in Table 1.

As we have known, the hormone concentrations of the noise records appeared in classifiers are always less than zero. Are all the records with negative hormone concentrations noise? It is obviously not the truth. Occasionally, records near the boundary have negative value. So, how to identify the noise records? We introduced a parameter $H_{threshold}$ to discriminate the noise. When the concentration of a record is less than the threshold, we believe that the record is a noise record and should be discarded from the classifier. When the threshold was set to three different values Φ, -0.05, and 0, we got three different experiment results which were shown in Fig. 8. "$H_{threshold} = \Phi$" means that the codes for deleting a noise were not executed. From Fig. 8(a), we can see that, when no noise record was generated, the accuracy of the algorithm is very high. From Fig. 8(b), it can be found that, when 20% noise records were added to the dataset and no noise-deleting program was executed, the accuracy of the classifier was bad. But, when hormone threshold $H_{threshold}$ was introduced into this algorithm, the accuracy of the classifier was improved. We can see it from Figs. 8(c) and 8(d). In fact, when the value of $H_{threshold}$ was less than zero, large value of it means long distance between the two class boundaries. We can see it from Fig. 7(d)–(l), when the value of $H_{threshold}$ was set to 0, noise cells in different classes were deleted obviously. But several boundary cells were deleted at the same time. That leads to a decline of classification accuracy. However, when the value of $H_{threshold}$ was set to Φ, no clear boundaries can be shown. The noise cells were remained in different classes. The classification accuracy was still very low. So, keeping an appropriate value of $H_{threshold}$ is very important for a classifier.

4.5 Dealing with biased class distribution

For data stream classification, skewed distributions and concept drifts are two common properties which make the training process very difficult [40]. In this section, we compare

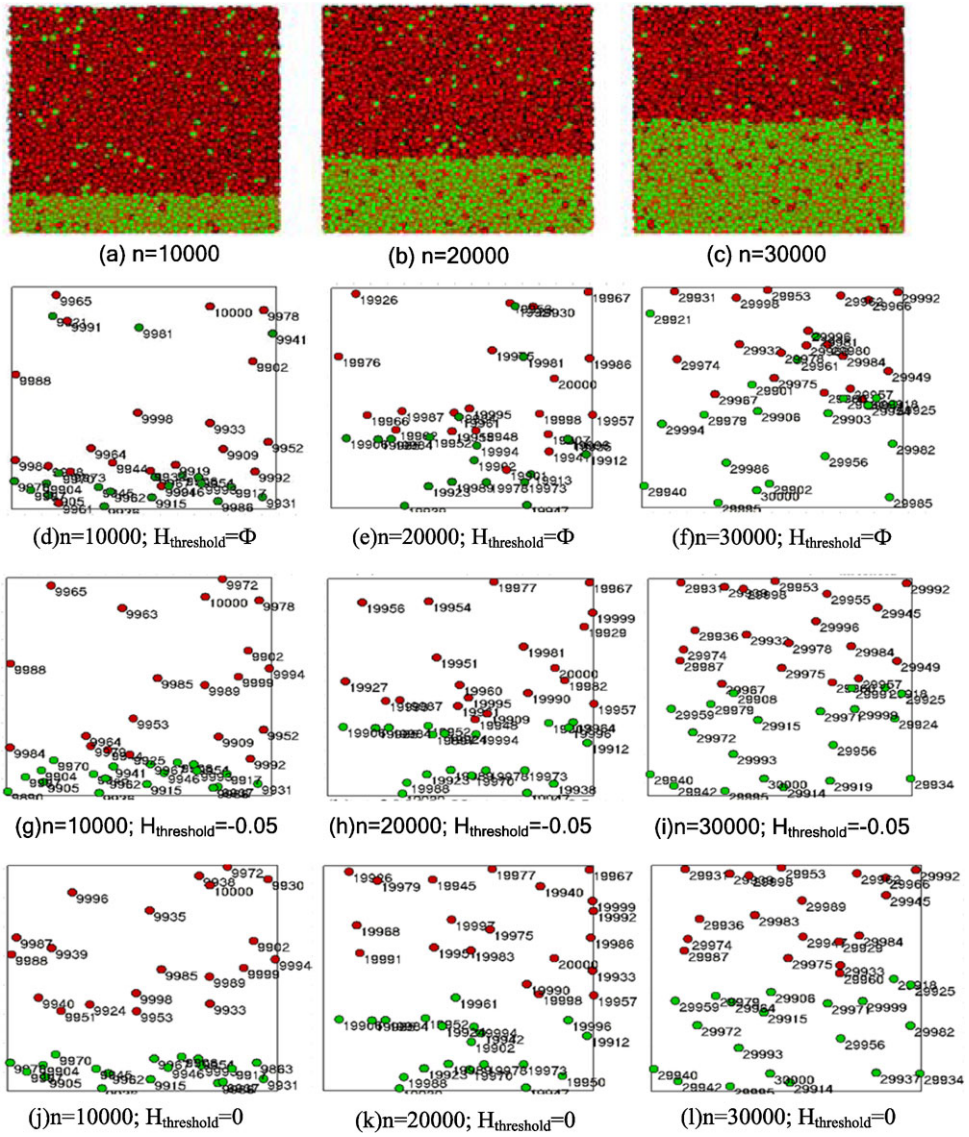

Fig. 7 Dealing with noises

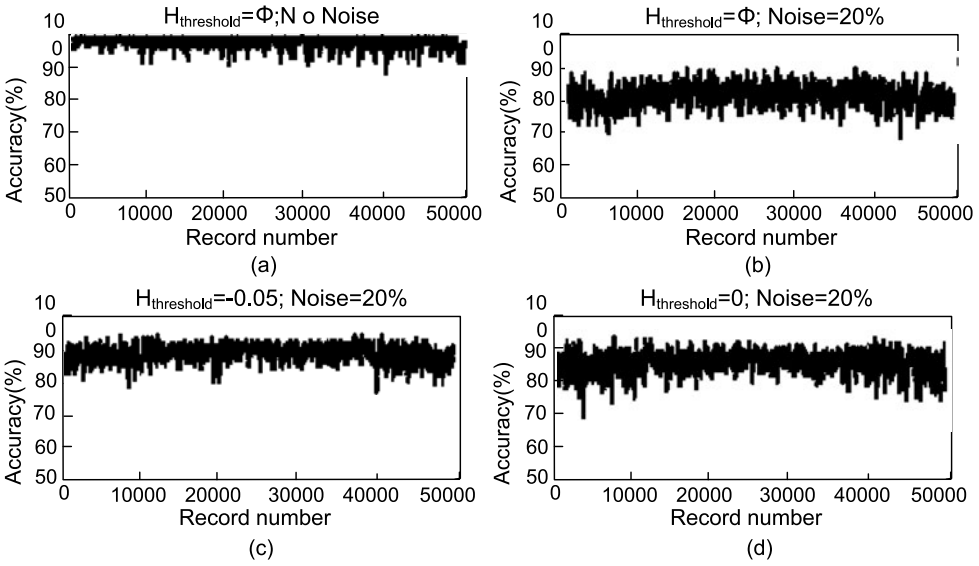

Fig. 8 Different $H_{threshold}$

Table 1 The hormone concentrations of endocrine cells

$H_{i,\text{Class-1}}$	Value	$H_{i,\text{Class-1}}$	Value	$H_{i,\text{Class-2}}$	Value	$H_{i,\text{Class-2}}$	Value
$H_{1,\text{Class-1}}$	1.397719E−03	$H_{11,\text{Class-1}}$	−2.421871E−03	$H_{1,\text{Class-2}}$	2.457238E−02	$H_{11,\text{Class-2}}$	2.020129E−02
$H_{2,\text{Class-1}}$	4.351127E−03	$H_{12,\text{Class-1}}$	−3.690987E−02	$H_{2,\text{Class-2}}$	4.598669E−02	$H_{12,\text{Class-2}}$	6.661846E−02
$H_{3,\text{Class-1}}$	−.0253661	$H_{13,\text{Class-1}}$	9.458338 − 03	$H_{3,\text{Class-2}}$	−1.730304E−02	$H_{13,\text{Class-2}}$	7.666826E−02
$H_{4,\text{Class-1}}$	−.1232346	$H_{14,\text{Class-1}}$	−1.075305E−02	$H_{4,\text{Class-2}}$	−4.727303E−02	$H_{14,\text{Class-2}}$	7.538774E−02
$H_{5,\text{Class-1}}$	−3.050148E−04	$H_{15,\text{Class-1}}$	−6.870199E−02	$H_{5,\text{Class-2}}$	−4.435137E−02	$H_{15,\text{Class-2}}$	9.256607E−02
$H_{6,\text{Class-1}}$	.012975	$H_{16,\text{Class-1}}$	−1.884288E−02	$H_{6,\text{Class-2}}$	1.463604E−02	$H_{16,\text{Class-2}}$	5.650862E−02
$H_{7,\text{Class-1}}$	.0146478	$H_{17,\text{Class-1}}$	2.267312E−03	$H_{7,\text{Class-2}}$	5.182039E−02	$H_{17,\text{Class-2}}$	.0353908
$H_{8,\text{Class-1}}$	−2.747295E−02	$H_{18,\text{Class-1}}$	1.357215E−02	$H_{8,\text{Class-2}}$	7.022131E−02	$H_{18,\text{Class-2}}$	5.146193E−02
$H_{9,\text{Class-1}}$	−5.181288E−02	$H_{19,\text{Class-1}}$	−5.501501E−02	$H_{9,\text{Class-2}}$	6.524275E−02	$H_{19,\text{Class-2}}$	9.562109E−02
$H_{10,\text{Class-1}}$	−1.012014E−02	$H_{20,\text{Class-1}}$	1.967972E−02	$H_{10,\text{Class-2}}$	−1.565104E−02	$H_{20,\text{Class-2}}$	5.275288E−02

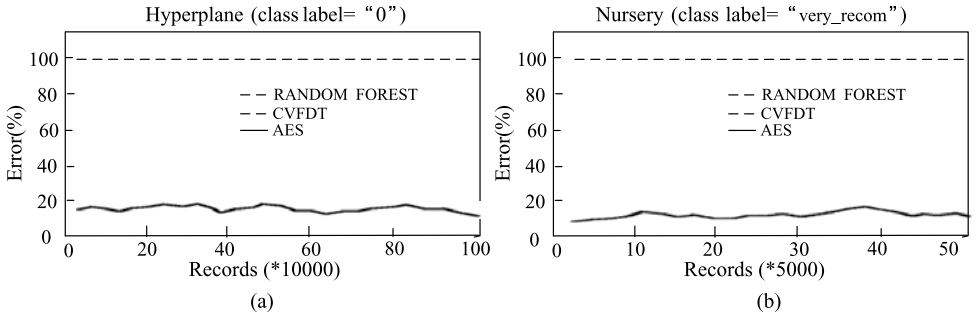

Fig. 9 Classification accuracy on biased class distribution

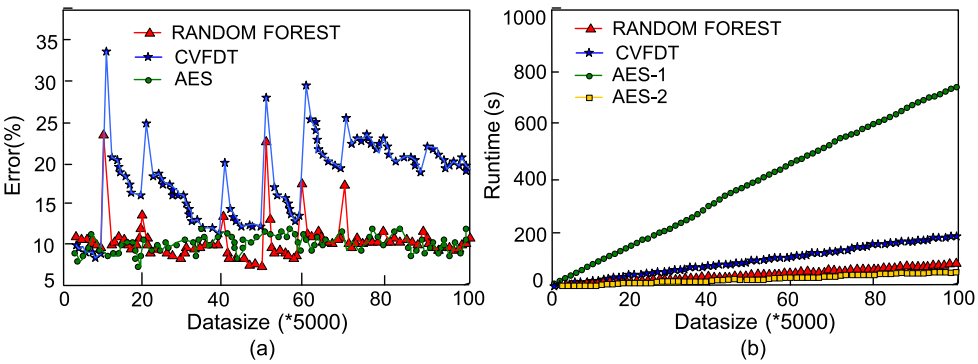

Fig. 10 Accuracy and runtime on dataset-4

the accuracy of AES algorithms, CVFDT [11] and Streaming Random Forest [2] on the dataset with biased class distribution. For dataset-4, we select appropriate parameter to make the number of records in class "0" be one percent of the total record number. For dataset nursery, there are five class labels in the dataset. The percentage of the records in class "very_recom" is 2.531%. Let error rate in Fig. 9 is the ratio of the records (belonging to class "0" or "very_recom") that are not classified correctly. From Fig. 9 we can see that only AES algorithm can correctly classify most of the records belonging to the rare class, while the other two algorithms can not. This is due to the limitation of the decision tree, which stops splitting a node when the ration of a class in this node is lower than a threshold. If the frequency of a class is very low, it is possible that the trained decision tree does not contain any node whose label is the rare class. In this case, the decision tree cannot correctly classify any record belonging to the rare class. However, for AES algorithm, the number of the boundary cells has no relation with the ration of a class. So, AES algorithm has similar accuracy for datasets in biased or unbiased class distribution.

4.6 Comparison with other algorithms

Without standard real-life data sets, it is very hard to compare the performance of different methods. However, we can compare them with artificial data sets with known noise. Most existing algorithms test their classifiers with data sets which has only two kinds of records. And these datasets

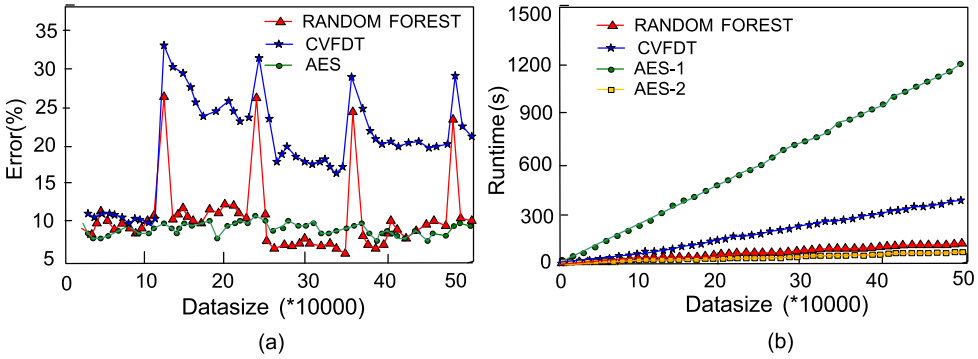

Fig. 11 Accuracy and runtime on dataset-5

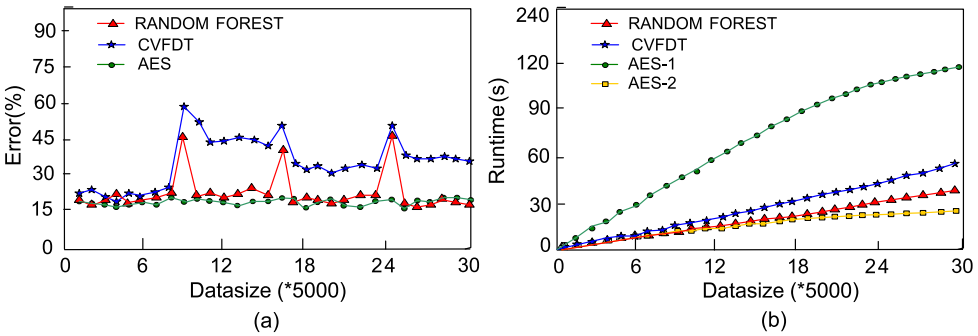

Fig. 12 Accuracy and runtime on dataset-6

have only a single decision boundary. In fact, the algorithm AES can be used to build effective classifiers for multiclass datasets. In Figs. 10–12, we compare our results with algorithms CVFDT [11] and Streaming Random Forest [2]. The sample buffers for algorithms CVFDT and Streaming Random Forests were set to 5000 or 10000. The error rate and runtime were reported for every 5000 or 10000 records. Figures 10(b)–12(b) illustrate incremental curves of runtime of the rival methods on datasets Hyper Plane, Cover Type and Letter Recognition, respectively. Having seen enough records, the two algorithms had a classification error of at least double the error percentage of the synthetic data. It can be seen from Fig. 10, every time a concept drift takes place, the error rates of the two algorithms changed largely. But for algorithm AES, the error curve changed smoothly and the mean variance of it is less than that of the other two algorithms. With the real-life dataset Dataset-5, and Dataset-6, we get a similar result. The reason is that, for AES algorithm, the classifier was updated every time a new record arrived and the buffers for data chunk are no longer needed. The fast updates improved the accuracy of the classifier. There are two kinds of updating manners in the proposed AES algorithm: AES-1 and AES-2. The first one is in an ensemble update manner and the second one is in an incremental manner. The difference of the running time cost by the two update algorithms is very large. We can see it from Fig. 10(b). The time cost by algorithm AES-1 is nearly four times greater than algorithm CVFDT. However, in AES-2, the time cost by it is nearly a third of the time cost by algorithm CVFDT. But the error rates of the two updating methods are the same.

5 Discussions and conclusion

In this paper, we represented a stream classification algorithm which is designed to classify a sample under the condition of limited memory resource. In this algorithm, artificial endocrine system was used to dynamically build a classifier. Every time a new record arrives, the endocrine cell moves from the most unfit record (position) to the new record (position). With the moving of the endocrine cells, the classifier updates dynamically. The algorithm can successfully handle concept drifts and can get more smooth classification results than other traditional algorithms.

The key feature of our algorithm is that, on the one hand, the classifier built with AES can be updated every time a new record arrives and the big data buffer of samples for training a real-time classifier is no longer needed. On the other hand, the stability of the classifier was improved and no burst of classifying error takes place. The run time of the algorithm AES-2 is similar with traditional algorithms. Experiments on synthetic and real datasets have shown that the algorithm can classify data streams with lesser memory space and classification error.

Acknowledgements We are grateful to the anonymous referees for their invaluable suggestions to improve the paper. This work was supported by Excellent Doctoral Dissertation Foundation of Xi'an University of Technology (116-211102), Natural Science Foundation of Shanxi Province, China (2010JM8028), and the National Natural Science Foundation of China (Grant Nos. 60873035, 61073091, 61100009).

References

1. Masud MM, Gao J, Khan L, Han J, Thuraisingham B (2011) Classification and novel class detection in concept-drifting data streams under time constraints. IEEE Trans Knowl Data Eng 23(6):859–874
2. Abdulsalam H, Skillicorn DB, Martin P (2011) Classification using streaming random forests. IEEE Trans Knowl Data Eng 23(1):22–36
3. Masud M, Gao J, Khan L, Han J, Thuraisingham B (2009) A multi-partition multi-chunk ensemble technique to classify concept-drifting data streams. In: Proc Pacific-Asia conf knowledge discovery and data mining (PAKDD'09)
4. Hassan YF (2010) Rough sets for adapting wavelet neural networks as a new classifier system. Appl Intell 35(2):260–268
5. Lee KK, Yoon WC, Baek DH (2006) A classification method using a hybrid genetic algorithm combined with an adaptive procedure for the pool of ellipsoids. Appl Intell 25(3):293–304
6. Zhang X, Chen G, Wei Q (2011) Building a highly-compact and accurate associative classifier. Appl Intell 34(1):74–86
7. Aggarwal CC, Han J, Wang J, Yu PS (2006) A framework for on-demand classification of evolving data streams. IEEE Trans Knowl Data Eng 18(5):577–589
8. Tsai CJ, Lee CI, Yang WP (2008) An efficient and sensitive decision tree approach to mining concept-drifting data streams. Informatica 19(1):135–156
9. Abdulsalam H, Skillicorn D, Martin P (2008) Classifying evolving data streams using dynamic streaming random forests. In: Proc 19th int'l conf database and expert systems applications (DEXA), pp 643–651
10. Wang P, Wang H, Wu X, Wang W, Shi B (2007) A low-granularity classifier for data streams with concept drifts and biased class distributio. IEEE Trans Knowl Data Eng 19(9):1202–1213
11. Domingos P, Hulten G (2000) Mining high-speed data streams. In: Proc sixth ACM SIGKDD, pp 71–80
12. Hulten G, Spencer L, Domingos P (2001) Mining time-changing data streams. In: Proc seventh ACM SIGKDD int'l conf knowledge discovery and data mining (KDD'01), pp 97–106
13. Kasabov N (2002) Evolving connectionist systems: methods and applications in bioinformatics, brain study and intelligent machines. Springer, New York
14. Ihara H, Mori K (1984) Autonomous decentralized computer control systems. Computer 17:57–66
15. Miyamoto S, Mori K, Ihara H (1984) Autonomous decentralized control and its application to the rapid transit system. Comput Ind 5:115–124
16. Shen WM, Will P, Galstyan A et al (2004) Hormone-inspired self-organization and distributed control of robotic swarms. Auton Robots 17:93–105
17. Avila-Garcia O, Canamero L (2005) Hormonal modulation of perception in motivation-based action selection architectures. In: The AISB'05 symposium. SSAISB Press, New York, pp 9–16
18. Kravitz EA (1998) Hormonal control of behavior: amines and the biasing of behavioral output in lobsters. Science 241:1175–1181
19. Avila-Garcia O, Canamero L (2004) Using hormonal feedback to modulate action selection in a competitive scenario. In: Proceeding of the 8th international conference on simulation of adaptive behavior. Springer, Heidelberg, pp 243–252
20. Shen WM (2003) Self-organization through digital hormones. IEEE Intell Syst 18:81–83
21. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory IT-13:21–27
22. Triguero I, García S, Herrera F (2010) IPADE: iterative prototype adjustment for nearest neighbor classification. IEEE Trans Neural Netw 21(12):1984–1990
23. Min YJ, Ta YP, Chen KB (2010) A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. IEEE Trans Geosci Remote Sens 48(3):1279–1293
24. Bermejo S, Cabestany J (2004) Local averaging of ensembles of LVQ-based nearest neighbor classifiers. Appl Intell 20(1):47–58
25. Zhang S (2011) Shell-neighbor method and its application in missing data imputation. Appl Intell 35(1):123–133
26. Huang CC, Lee HM (2004) A grey-based nearest neighbor approach for missing attribute value prediction. Appl Intell 20(3):239–252
27. Miyamoto S, Mori K, Ihara H (1984) Autonomous decentralized control and its application to the rapid transit system. Comput Ind 5:115–124
28. Shen WM, Salemi B, Will P (2002) Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. IEEE Trans Robot Autom 18:700–712
29. Mendao M (2007) A neuro-endocrine control architecture applied to mobile robotics. Dissertation for the Doctoral Degree, Canterbury, University of Kent, pp 1–49
30. Walker J, Wilson M (2008) A performance sensitive hormone-inspired system for task distribution amongst evolving robots. In: 2008 IEEE/RSJ international conference on intelligent robots and systems. IEEE Press, New York, pp 1293–1298
31. Xu QZ, Wang L (2011) Lattice-based artificial endocrine system model and its application in robotic swarms. Sci China Ser F 54(4):795–811
32. Zhang Y, Jin X (2006) An automatic construction and organization strategy for ensemble learning on data streams. SIGMOD Rec 35(3):28–33
33. Masud M, Gao J, Khan L, Han J, Thuraisingham B (2009) A multi-partition multi-chunk ensemble technique to classify concept-drifting data streams. In: Proc Pacific-Asia conf knowledge discovery and data mining (PAKDD'09)
34. Sun Y, Mao G, Liu X, Liu C (2007) Mining concept drifts from data streams based on multi-classifiers. In: Proc 21st int'l conf advanced information networking and applications workshops (AINAW), pp 257–263
35. Wang P, Wang H, Wu X, Wang W, Shi B (2005) On reducing classifier granularity in mining concept-drifting data streams. In: Proc fifth IEEE int'l conf data mining (ICDM'05)
36. Pang S, Ozawa S, Kasabov N (2005) Incremental linear discriminant analysis for classification of data streams. IEEE Trans Syst Man Cybern, Part B, Cybern 35(5):905–914
37. Xu WH, Qin Z, Chang Y (2011) Clustering feature decision tree for semi-supervised classification form high-speed data stream. J Zhejiang Univ Sci C (Comput & Electron) 12(8):615–628
38. Blake C, Merz C (1998) UCI Repository of Machine Learning Databases, Dept of Information and Computer Science, Univ. of California, Irvine
39. Wang H, Fan W, Yu PS, Han J (2003) Mining concept-drifting data streams using ensemble classifiers. In: Proc ninth ACM SIGKDD int'l conf knowledge discovery and data mining
40. Gao J, Ding B, Han J (2008) Classifying data streams with skewed class distributions and concept drifts. IEEE Internet Comput 12(6):37–49

Li Zhao was born in 1974. He received his M.S. degree from Xian University of Technology in 2009. He is currently a Ph.D. candidate at the college of automation and information engineering, Xi'an University of Technology. His current research interests include evolutionary computation and data mining.

Lei Wang was born in 1972. He received his Ph.D. degree from Xidian University in 2001. From 2002 to 2004, he was a post-doctoral researcher at University of Fribourg, Switzerland. Now he is a Professor and doctoral supervisor at Xi'an University of Technology. His current research focuses on artificial intelligence and evolutionary computation.