2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Evolutionary Semi-supervised Learning with Swarm Intelligence

Ping He, Lin Lu, Xiaohua Xu*, Heng Qian, Wei Zhang, and Yongsheng Ju

***Abstract*— To address the issue of evolutionary data classification, we propose an evolving swarm classification model. It treats each class as an ant colony carrying different type of pheromone. The ant colonies send their members to propagate their unique pheromone on the unlabeled instances, so as to label them for member recruitment. Meanwhile, the unlabeled instances are treated as unlabeled ants, which also have their preferences for joining one of those labeled colonies. We call it homing feedback, and integrate it into the pheromone update process. Afterwards, the natural selection process is carried out to keep a balance between the member recruitment and the ant colony size maintenance. Sufficient experiments demonstrate that our algorithm is effective in the real-world evolutionary classification applications.**

I. Introduction

THe problem of mining evolutionary data is challenging and appealing in data mining. Evolutionary data, commonly known as dynamic data, comes from many application fields like weblogs and GPS sensors, where the topics and objects evolve with time. The main difference of evolutionary data from static or persistent data lies in that, each instance in the time series corresponds to a different time step and the attributes vary with time. Besides, evolutionary data is also different from streaming data, because there is no constant flow of information. Instead, new data may come at any time, with periods of inactivity in between. Therefore, although the time line in the real world is continuous, we only consider consecutive evolutionary data in discrete time steps.

Based on the partition of unsupervised learning and supervised learning, evolutionary data mining are classified into the two categories, evolutionary clustering and evolutionary classification. Evolutionary clustering [1], especially evolutionary spectral clustering [2] is an emerging research area applied to many real-world applications, such as clustering dynamic Web and blog contents. Evolutionary classification, on the other side, refers to the situation where some instances in the data flow are attached with known labels, and the target is to classify the unlabeled data in the real-time.

A simple method to approach such problems is to apply static clustering or classification methods on the most recent data at each time step. However, this approach is extremely sensitive to noise and produces results that are unstable and inconsistent with the results from adjacent time steps. As a result, evolutionary clustering and classifying methods have been developed, with the goal of producing results that reflect long-term concept drifts in the objects while being robust to short-term variations. Up to now, there are many online algorithms designed for coping with both clustering and classification problems.

As to classifications, a series of data stream classification methods [3]–[13] have been proposed concerning concept drifts, class distribution and temporal smoothness. Despite of the successful application of those methods, the assumption of entire labeled data availability is often violated in the real-world problems, because labels may be scare and not readily available. As a result, the traditional stream classification algorithms have to either update the classifiers with just a few labeled data, which usually results in poor performance, or wait a long time to get enough labeled data, which also affect the classification results due to the usage of outdated data.

Recently, semi-supervised learning methods have been put forward. In 2009, Yangging Jia et al. [14] proposed a semi-supervised classification algorithm for dynamic mail post categorization. They carried out temporal smoothness assumption using temporal regularizers defined in the Hilbert space, and then derived the online algorithm that efficiently finds the closed-form solution to the target function. Later in 2011, H. Borchani et al. [15] proposed a new semi-supervised learning approach for concept-drifting data streams. They aim to take advantage of unlabeled data to detect possible concept drifts and, if necessary, update the classifier over time even if only a few labeled data are available.

However, both the previous works impose a relatively strong assumption that: At any time stamp, at least one labeled instance for each class should be provided. This can be easily violated in the real-world applications. In this paper, we adopt a weaker assumption that users are expected to specify the number of labels and at least one labeled sample for one class at the first time ($t = 0$). Then in the later time steps, the feed of labeled samples is unnecessary.

In this paper, we present a semi-supervised swarm classification model which applies the Darwin's theory of evolution on the classification of evolutionary data. In our work, we treat each data instance as an ant and each class of labeled instances as an ant colony. The whole swarm, or the whole dataset, is composed of all the different colonies and the unlabeled ants, which evolve with time based on the simulation of the natural selection behavior. In other words, our proposed algorithm is 'self-training' in nature. Besides, compared to the previous research, our method can be applied to a more generalized scenario, where the class distribution is arbitrary and the number of labeled instances is unfixed (even down to 0) at each time step.

The rest of this paper is organized as follows: Section II introduces our evolutionary classification model, and describes in detail its classification algorithm. Some simulation and

* Corresponding author: Department of Computer Science, Yangzhou University, Yangzhou 225009, Ph: +8651487978309, Fax: +8651487887937, e-mail: arterx@gmail.com

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

TABLE I

SYMBOLS AND THEIR DEFINITIONS

t	time step
T	number of blocks
X^t	data set at time step t
x^t	an instance with a time stamp t
n_t	the size of X^t, equals to $\|X^t\|$
X_m^t	labeled data set at time step t
m_t	the number of labeled instances at time step t, equals to $\|X_m^t\|$
X_u^t	unlabeled data set at time step t, equals to A_0^t
u_t	the number of unlabeled instances at time step t, equals to $\|X_u^t\|$
Y^t	set of labels corresponding to X^t
c	number of different labels
A^t	set of ant colonies at time step t
A_l^t	the ant colony corresponding to the l^{th} class(label)
a_{li}	the i^{th} member of colony A_l^t

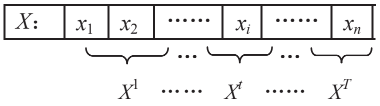


Fig. 1. Producing evolutionary data with T time steps

results are presented in Section III to show its performance. Finally, Section IV concludes the paper.

II. EVOLUTIONARY SWARM CLASSIFICATION MODEL

A. Notations and definitions

We begin with some definitions given in Table I. Each data instance in this paper is associated with not only a label y but also a time stamp t. For simplicity, we assume that the time t takes integer value from 1 to T. For example, posts of a mailing list from each month can be considered as data from one time step. In simulation, the whole data set X is randomly permuted, and then divided evenly into T blocks, as we can see from Fig. 1. Thus we are given a set of data subsets $X = \{X^1, X^2, \dots, X^T\}$ that are from T consecutive time steps. At time step t, $X^t = X_m^t \bigcup X_u^t = \{x_1^t \dots x_{m_t}^t, x_{m_t+1}^t, \dots, x_{n_t}^t\}$, where $X_m^t = (x_i^t)_{i=1\dots m_t}$ is labeled, corresponding to the known label subset $Y_m^t = (y_i^t)_{i=1\dots m_t}$, and $X_u^t = (x_i^t)_{m_t+1\dots n_t}$ is unlabeled, corresponding to the unknown label subset $Y_u^t = (y_i^t)_{m_t+1\dots n_t}$.

Suppose the evolutionary data is to be classified into c classes. The target of semi-supervised learning on evolutionary data is to predict X_u^t instantly at time step t, by the help of previous information $X^{1,\dots,t-1}$ and X_m^t (if provided). Due to the existence of multiple labels, we introduce a $n_t \times c$ label indicating matrix for the time step t.

$$F^t = \begin{bmatrix} F_m^t \\ F_u^t \end{bmatrix} = (f_{ij}^t)_{n_t \times c} \tag{1}$$

where F_m^t and F_u^t respectively denote the states of the known labeled set X_m^t and the unknown set X_u^t. In the beginning, F^t is initialized so that $f_{ij}^t = 1$ if and only if $y_i^t = j$, otherwise $f_{ij}^t = 0$.

To solve this problem, we propose an evolutionary swarm classification (ESC) model, which treats each class as an ant colony, respectively denoted as $A_{l=1\dots c}^t$. Particularly, we let the unlabeled data set X_u^t form a special colony with unknown class, represented by $A_0^t = X_u^t$. Correspondingly its i^{th} member is defined as a_{0i}^t. Then an ant a_{li}^t of colony A_l^t is labeled if $l > 0$ or unlabeled if $l = 0$. By doing so, the swarm at time t is composed by $c+1$ ant colonies and $A^t = \{A_0^t, A_1^t, \dots, A_c^t\}$. In addition, the members in each labeled colony are elite ants that are selected from the history data sets $X^{1,\dots,t-1}$. The determination of the elite ants involves two stages.

Before label prediction (BL): Since X_m^t represents the training data set with already known labels, its instances are directly treated as labeled ants and assigned to ant colonies according to their labels.

After label prediction (AL): Once the labels of unlabeled ants in A_0^t are predicted, those ants showing high fitness to their predicted classes will be regarded as elite ants and added to the corresponding ant colonies.

B. Framework of swarm classification

The framework of our proposed algorithm is illustrated in Fig. 1. We assume that each labeled colony is an ant nest possessing one type of pheromone, thus there are c nests in this swarm model (Step 1-2). To survive in such environment, each nest has to establish itself by recruiting members, i.e. the unlabeled ants, at the **AL** stage (Step 3-9).

As illustrated in Algorithm 1, the member recruitment is done by the pheromone propagation of labeled ants and the preference feedback of unlabeled ants (Step 6). After the pheromone update, all the nests will compete with each other according to the proportion of their pheromone on the unlabeled ants. The nest with the most pheromone can assign its label to the unlabeled ant (Step 7), and check whether or not it is an elite ant (Step 8), which will be discussed later.

Algorithm 1 Evolutionary Swarm Classification Framework

```
1. Initialize parameters
2. Construct swarm model
3. for t = 1 to T do
4.     Receive $\langle X_m^t, Y_m^t \rangle$ and $X_u^t$
5.     Initialize random walk environment
6.     Update pheromones
7.     Predict $Y_u^t$
8.     Swarm evolution
9. end for
```

C. Heuristic value rule

For meta-heuristic algorithms, heuristic value is necessary because it plays the role of guiding function and it also provide a quality measurement for different solutions. Traditional ACO for traveling salesman problem treats the inverse of pairwise distance to be the heuristic value. However in our paper, the transition probability of random walk on graph is more appropriated to be the heuristic value.

1) Distance measure: For evolutionary data, its intrinsic structure is unknown, so we assume that each labeled ant shares a link to all unlabeled ants and the unlabeled ants themselves are all connected. Thus, we provide two kinds of distances: one is the distance between labeled ants and unlabeled ants, the other is the distance among unlabeled ants.

$$d_{ij}^t = \begin{cases} (a_{li}^t - a_{0j}^t)^T (\Sigma_l^t)^{-1} (a_{li}^t - a_{0j}^t) & \text{if } l > 0 \\ \frac{||a_{0i}^t - a_{0j}^t||^2}{2\sigma^2} & \text{if } a_{0i}^t, a_{0j}^t \in A_0^t \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

Here Σ_l^t represents the covariance matrix of the l^{th} colony at time t and σ is a spread parameter. Note that 1) we use Mahalanobis distance rather than Euclidean distance to compute the similarity between an unlabeled ant and a labeled one, 2) we do not take into account of the linkage among labeled ants.

2) Heuristic value: We define the heuristic value as the similarity between two ants. Specifically, given two ants a_{li}^t and a_{lj}^t, their similarity is computed as follows:

$$\eta_{ij}^t = e^{-d_{ij}^t} \tag{3}$$

η is a heuristic value matrix of size $u_t \times n_t$. We divide it into two blocks based on the fact that $n_t = m_t + u_t$. The first block with size $u_t \times m_t$ refers to the similarity between unlabeled ants and labeled ants, the second block with size $u_t \times u_t$ refers the similarity among unlabeled ants themselves. Given such similarity value, any ant will prefer to move to its intimate neighbors.

D. Pheromone update rule

1) Pheromone matrix: For pheromone update, we use aggregation pheromone instead of individual pheromone. In each time step, the pheromone matrix is defined as a $u_t \times c$ matrix τ, where each column vector is the accumulation of pheromone left by one nest on all unlabeled ants. For instance, given a nest label $l(l > 0)$ at time step t, $\tau_{*l}^t = \tau^t(:, l)$ is the column vector whose element $\tau_{jl}^t = \tau^t(j, l)$ indicates the pheromone left by the ants from nest A_l^t on the unlabeled ant a_{0j}.

Suppose τ^{t_s} represents the s^{th} iteration of the pheromone matrix during the pheromone update for time step t, the initial value is set $\tau^{t_0} = F_u^t$ (see eq. (1)). With the heuristic value and pheromone value, the update rule is defined based on the pheromone propagation by the random walk of labeled ants among those unlabeled ants. Given an unlabeled ant a_{0i}^t and a nest $A_l^t(l > 0)$, the updated pheromone intensity on a_{0i}^t is shown in eq. (4).

$$\tau_{il}^{t_{s+1}} \leftarrow \sum_{j=1}^{u_t} \eta_{i(j+m_t)}^t \tau_{jl}^{t_s} + \sum_{k=1}^{m_t} \eta_{ik}^t \tau_{(k+m_t)l}^{t_0} \tag{4}$$

It consists of two parts, respectively corresponding to the two blocks of η^t. In the first term, $\eta_{i(j+m_t)}^t$ represents the similarity between the i^{th} unlabeled ant and the j^{th} labeled ant, while in the second term, η_{ik}^t represents the similarity between the i^{th} and j^{th} unlabeled ants. Therefore, the first term $\sum_{j=1}^{u_t} \eta_{i(j+m_t)}^t \tau_{jl}^{t_s}$ indicates the pheromone propagated from the labeled ants, and the second term $\sum_{k=1}^{m_t} \eta_{ik}^t \tau_{(k+m_t)l}^{t_0}$ indicates the pheromone propagated form the unlabeled ants. The reason for using $\tau_{(k+m_t)l}^{t_0}$ instead of $\tau_{(k+m_t)l}^{t_s}$ is because we do not want to rely on the propagated pheromone on the unlabeled ants except for their initial values.

2) Homing feedback: The unlabeled ants are intelligent agents as well, so they can choose one nest as its home. The nest selection of unlabeled ants is called **Homing**, which is also take into account as a feedback on our pheromone update rule. We assume that an unlabeled ant intends to move to its nearest nest, and use K-Nearest Neighbor (KNN) to select nests for unlabeled ants. KNN is robust in the scenario of evolutionary data classification, where data is local, dynamic and with varying distribution, because it is a non-parametric method and only has one assumption that similar input has similar output.

To integrate the homing feedback into pheromone update, we introduce another heuristic value $\tilde{\eta}$, whose first block of size $u_t \times m_t$ differs from that of η in that each row only keeps the k largest similarity values remained while the others are replaced by 0. In other words, each unlabeled ant only keeps track of its k nearest labeled ants.

By combining the pheromone propagation and the homing feedback, the final pheromone update equation is given in eq. (5).

$$\tau_{il}^{t_{s+1}} \leftarrow \sum_{j=1}^{u_t} \eta_{i(j+m_t)}^t \tau_{jl}^{t_s} + \sum_{k=1}^{m_t} (\alpha \eta_{ik}^t + (1 - \alpha) \tilde{\eta}_{ik}^t) \tau_{(k+m_t)l}^{t_0} \tag{5}$$

If a_{lk}^t is one of a_{0i}^t's k nearest neighbors, then $\tilde{\eta}_{ik}^t = \eta_{ik}^t$, else $\tilde{\eta}_{ik}^t = 0$. The weight parameter α is introduced to balance the two different heuristic values.

When the pheromone matrix converges $\tau^t = \tau^{t_\infty}$, an arbitrary unlabeled ant a_{0i}^t is assigned to the colony that leaves the maximal pheromone on it.

$$y_i^t = \max_l \tau_{il}^t \tag{6}$$

E. Swarm evolution

Swarm evolution is inspired by the natural selection process, where the fitter individuals have larger chance to survive. Given a colony A_l^t belonging to class $l_{1,\ldots,c}$ at time step t, we define the fitness of $a_{l_i}^t \in A_l^t$ as

$$fitness(a_{li}^t) = \frac{1}{|A_l^t|} \sum_{a_{lj}^t \in A_l^t, i \neq j} e^{-(a_{li}^t - a_{lj}^t)^T (\Sigma_l^t)^{-1} (a_{li}^t - a_{lj}^t)} \tag{7}$$

where $|A_l^t|$ is the capacity of A_l^t, Σ_l^t is the covariance matrix of A_l^t, and $(a_{li}^t - a_{lj}^t)$ is computed using the attribute values (included in X^t) of a_{li}^t and a_{lj}^t.

Based on the fitness evaluation, the natural selection process includes two stages.

1) **Member Addition**. For each unlabeled ant, if its fitness value to its predicted class is higher than a acceptable level $\beta \in (0, 1]$, then this ant will be added to the predicted colony as the training set for the next generation of label prediction.
2) **Member Deletion**. The unlimited expansion of ant colonies is unrealistic and should be avoided. Besides, some previous elite members in one colony may turn unfit with time goes by. Therefore, once the size of a colony reaches maximum $(MaxColonySize)$, the deletion process will be carried out to remove those members with the least fitness values.

Algorithm 2 illustrates the Ant Evolutionary Classification at time step t. The detailed description of the natural selection process is provided in step 9-20. This procedure can help the ant colonies evolve from one generation to the next smoothly, as well as avoiding the excessive expansion of each colony.

Algorithm 2 ESC at time stamp t

```
Input: $\langle A^t, Y_C^t \rangle$, $\langle X_M^t, Y_M^t \rangle$, $X_U^t$, $k$, $\alpha$, $\beta$, $MaxColonySize$
Output: $A^t, Y_A^t, Y_U^t$
1.  if $X_M^t$ is not empty then
2.      $A^t = A^{t-1} \bigcup X_M^t$, $Y_A^t = Y_A^{t-1} \bigcup Y_M^t$
3.  end if
4.  if $X_U^t$ is not empty then
5.      Compute the heuristic matrix $\eta^t$
6.      Find the $k$ nearest labeled ants of each unlabeled ant, represented by $\tilde{\eta}^t$
7.      Update the pheromone matrix according to eq. (5) until it converges
8.      Predict the label vector $Y_U^t$ of $X_U^t$ according to eq. (6)
9.      for all $a_{0i}^t \in X_U^t$ do
10.         Let $l = y_i^t \in Y_U^t$
11.         if $fitness(a_{0i}^t) > \beta$ then
12.             $A_l^t = A_l^t \cup a_{0i}^t$
13.         end if
14.     end for
15.     for each colony $A_l^t$ do
16.         while $|A_l^t| > MaxColonySize$ do
17.             Find the member with the least fitness value
18.             Remove it out of $A_l^t$
19.         end while
20.     end for
21. end if
```

III. Experiments

We test our algorithm on three data sets, whose details are summarized in Table II. Twomoons is a synthetic data set including two classes of intertwining moons. Mushroom and Hyperplane data sets from the UCI repository are used to simulate the concept drift problem.

TABLE II
Summarization of Data sets

Dataset	#Size	#Attributes	#Classes
Twomoons	2000	2	2
Mushroom	8124	22	2
Hyperplane	10000	10	2

We use the default 1% labeled ratio to generate the training data and randomly distribute them into T time blocks. As a result, the provided labeled data may vary with different times blocks, and even maybe absent. For the evaluation of evolutionary classification performance, we adopt both overall classification accuracy and local classification accuracy. The overall accuracy refers to the percentage of the overall correctly classified instances after T times of local predictions. To give a reliable result, 50 runs of random simulation are carried out to produce an average overall classification accuracy. The local classification accuracy can also be called block accuracy, which refers to the classification accuracy within each time block.

A. Demo

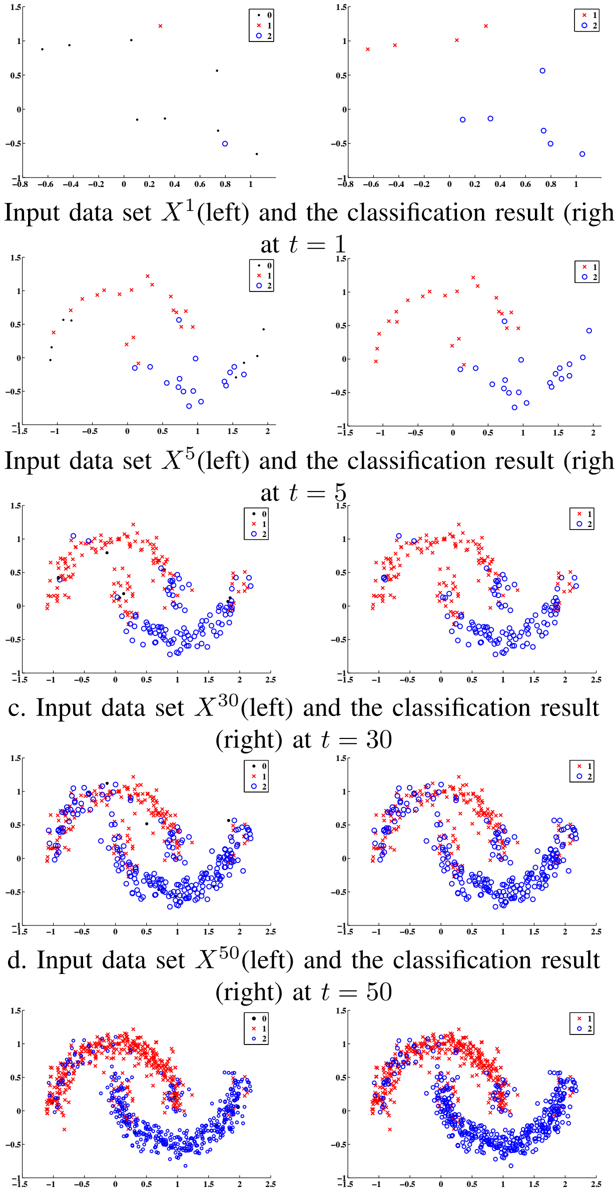
Before the evaluation of our algorithm, a demo is provided to classify the Twomoons data set for a better view of our method. At first, the data set is divided into $T = 100$ time intervals. Fig. 2 shows the the evolutionary classification process in five ascending time steps. The red crosses and blue circles respectively denote the two different classes of data, and the black dots represent the unlabeled instance. The subfigures at the left side depict the input data including the previous classification result, while the subfigures at the right side depict the predicted labels of those black dots in the left subfigures. As we can see, when $t = 1$, only two labeled instances are provided and we cannot recognize the intrinsic structure of the input data. Later, after more instances are provided, our algorithm gradually assigns labels to the unlabeled data points and then discovers the manifold structure of two moons.

B. Parameter issues

Since our proposed method involves several parameters, it is necessary to study the degree of influence that those parameters exert on the classification performance of our algorithm. To this end, we investigate each parameter by fixing the other parameters and plotting the relationship of the parameter and the classification accuracy.

1) The number of blocks T: In our experiment, we split each data set into 10 blocks. For each time step t, the labels of the unlabeled instances are instantly predicted by colony A^t. In this section, we define the block accuracy to be the ratio of correctly classified unlabeled instances in X_u^t.

In Fig. 3, a histogram is displayed to study the instantly-prediction performance of our model. At each block, there are three bars denoting the block accuracies corresponding respectively to the three data sets. As we can see, for Twomoons, the block accuracy gradually increases as time goes by, but for Mushroom data, it first reaches its peak value



a. Input data set X^1(left) and the classification result (right) at $t = 1$

b. Input data set X^5(left) and the classification result (right) at $t = 5$

c. Input data set X^{30}(left) and the classification result (right) at $t = 30$

d. Input data set X^{50}(left) and the classification result (right) at $t = 50$

e. Input data set X^{100}(left) and the classification result (right) at $t = 100$

Fig. 2. Classification simulation on Twomoons data set

at step 3, then goes down until the stable state. One thing in common shared by the three data sets is that they all tend to maintain at a certain level at the last few steps.

Taking the runtime and time steps into consideration, we list the average running time in Table III with respect to four T values. On the one hand, the larger the number of time blocks, the less time spent by our ESC model; On the other hand, the larger the size of data set, the more time is needed. As a result, to reduce the run time, it is better to ensure that each time block has a small size (about 100) of input data.

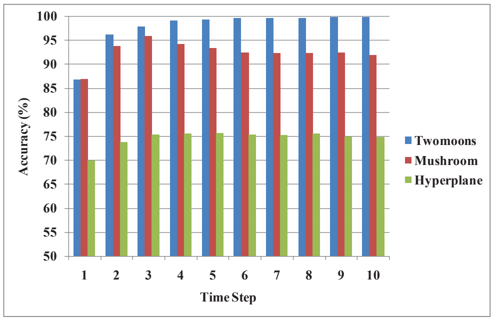


Fig. 3. Average block accuracy from block 1 to 10 on three data sets

TABLE III

THE AVERAGE RUNTIME(S) WITH DIFFERENT TIME STEPS

Dataset	$T = 10$	$T = 50$	$T = 100$	$T = 150$
Twomoons	2.187	1.310	1.246	1.247
Mushroom	79.623	21.175	20.166	19.782
Hyperplane	152.832	29.617	26.992	26.396

2) $MaxColonySize$: In our algorithm, we let each ant colony maintain a fixed size. To test the influence of the parameter $MaxColonySize$, we adopt five values (50, 100, 150, 200, 500) as the max colony size. Fig. 4 presents three subfigures respectively corresponding to the three data sets, where each figure has five curves under five values denoting the block accuracy from time step 1 to 10. We can note that this parameter exerts obvious influence on Mushroom data set.

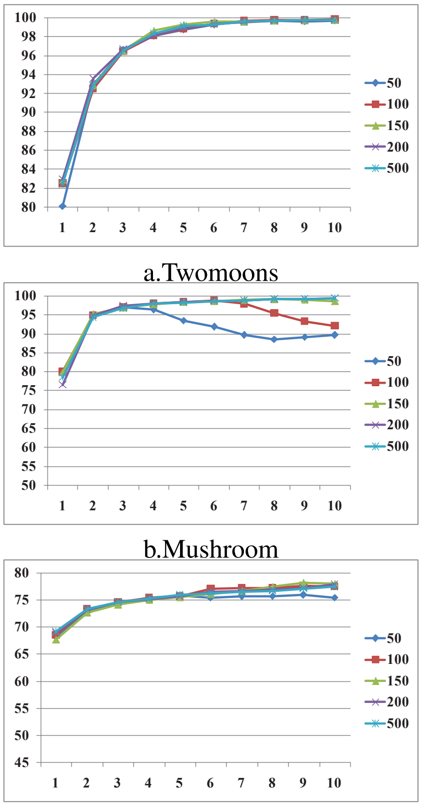
In Table IV, each row shows the average overall accuracy on one data set with five different $MaxColonySize$ values. We note that Twomoons and Hyperplane data sets perform best at size 150. It indicates that 150 might be a good choice for $MaxColonySize$. In addition, the setting of this parameter should also take into account of the physical memory and the runtime cost.

TABLE IV

AVERAGE OVERALL ACCURACY WITH DIFFERENT $MaxColonySize$

	50	100	150	200	500
Two moons	97.17	97.86	**97.98**	97.71	96.98
Mushroom	92.76	95.01	96.26	**96.7**	96.6
Hyperplane	74.41	75.50	**75.65**	74.86	75.19

3) The number of nearest neighbors k: In the homing feedback of unlabeled ants, the number of nearest neighbors k influences the pheromone update rule (see eq. (5)). In Table V, we test k with five different values (1, 2, 3, 4, ∞), where $k = \infty$ means $\tilde{\eta}^t = \eta^t$. We can see that the best performance occurs mostly when $k = 1$. Fig. 5 also presents three subfigures in correspondence to the three data sets, where each figure has five curves under five k values denoting the block accuracy from time step 1 to 10. It again validates that $k = 1$ produces the best classification result on both Twomoons and Mushroom data sets. Besides, in all the subfigures, $k = \infty$ shows the worst performance among the five.



a.Twomoons

b.Mushroom

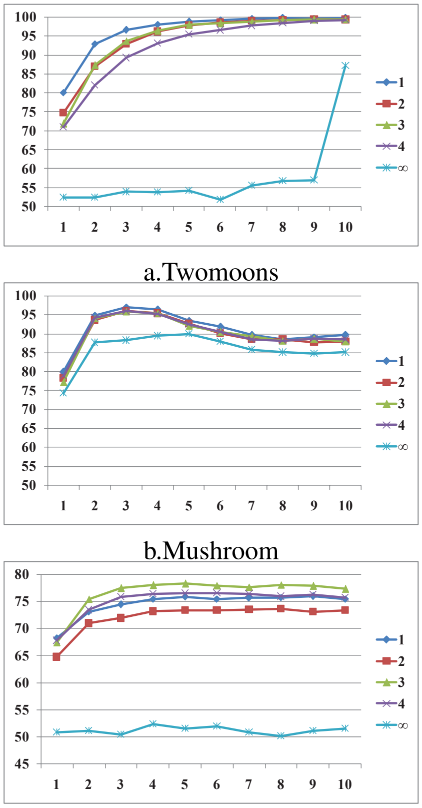
c.Hyperplane

Fig. 4. Average block accuracy with different $MaxColonySize$ on three data sets

TABLE V

AVERAGE OVERALL ACCURACY WITH DIFFERENT k

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = \infty$
Two moons	**97.17**	94.75	95.10	92.76	57.19
Mushroom	**92.76**	92.30	92.70	92.66	92.40
Hyperplane	74.41	72.54	**76.73**	75.14	66.99



a.Twomoons

b.Mushroom

c.Hyperplane

Fig. 5. Average block accuracy with different k on three data sets

TABLE VI

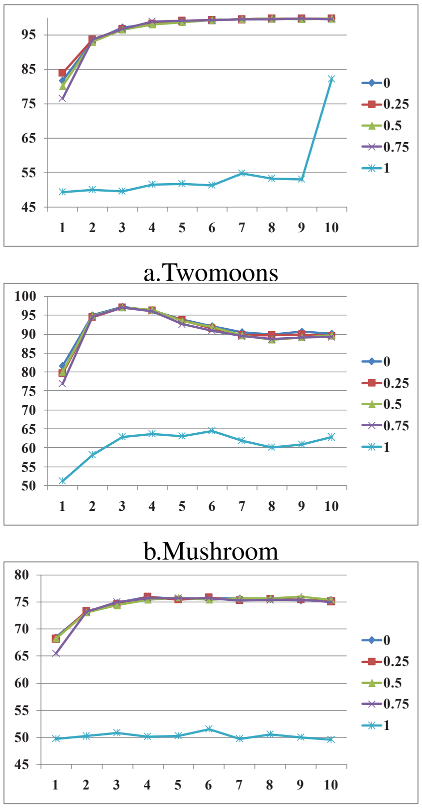
AVERAGE OVERALL ACCURACY (AOA)WITH DIFFERENT α

	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1.0$
Two moons	97.56	**97.92**	97.17	96.77	54.27
Mushroom	**95.47**	92.00	91.91	92.08	91.93
Hyperplane	74.58	**74.95**	74.41	73.95	52.00

4) Balance weight α: The parameter α controls the balance in the integration of homing feedback (see eq. (5)). The value of α is implicitly is restricted in the range [0,1]. Here we consider five different values, $\alpha = (0, 0.25, 0.5, 0.75, 1)$. If $\alpha = 1$, then the homing feedback is not incorporated into the pheromone update. When $\alpha = 0$, then the homing feedback is incorporated while the pheromone propagation from the unlabeled ants is discarded. Fig. 6 illustrates three subfigures in correspondence to the three data sets, where each figure has five curves under five different α values denoting the block accuracy from time step 1 to 10. We can see that when $\alpha = 1$, all the three subfigures show the worst classification performance. On the other side, the other four different values of α show similar accuracy curves.

Table VI reports the average overall accuracy of parameter α. It demonstrates that α is necessary for improving the classification performance of our algorithm. When $\alpha = 0.25$, both Twomoons and Hyperplan data sets achieve the best performance, outperforming both only pheromone propagation ($\alpha = 1$) and only homing feedback ($\alpha = 0$).

5) Addition threshold β: The parameter β controls the fitness threshold for adding a predicted instance into the corresponding colony as a training data. The value of β is also within the range of [0,1]. To investigate the impact of parameter β, we use five different values $\beta = (0.6, 0.7, 0.8, 0.9, 1.0)$. Fig. 7 illustrates three subfigures in correspondence to the three data sets, where each figure has five curves under five different β values denoting the block accuracy from time step 1 to 10. Table VII reports the average overall accuracy of parameter β. It can be seen that the highest overall accuracy value for Twomoons happens when $\beta = 1$, which turns to be supervised learning because no unlabeled instances are utilized. For the other two data sets, their best classification performances occur at $\beta = 0.7$.

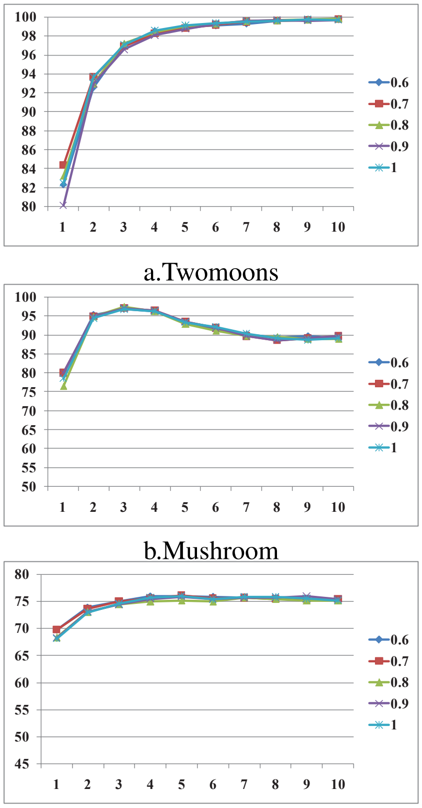


a.Twomoons

b.Mushroom

c.Hyperplane

Fig. 6. Average block accuracy with different α on three data sets



a.Twomoons

b.Mushroom

c.Hyperplane

Fig. 7. Average block accuracy with different β on three data sets

TABLE VII

AVERAGE OVERALL ACCURACY WITH DIFFERENT β

	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$	$\beta = 1.0$
Two moons	97.25	97.37	97.12	97.17	**97.80**
Mushroom	92.79	**92.99**	92.70	92.76	92.83
Hyperplane	74.79	**74.87**	74.4	74.41	74.56

In conclusion, we find that the $MaxColonySize$ parameter contributes more than the other parameters in producing the best classification accuracy. The best classification result of our algorithm on the three data sets are respectively 97.98%, 96.70% and 76.73%.

IV. CONCLUSION

Learning evolutionary data is a new challenge in the data mining area, where the concept drifts and smoothness assumption should be both addressed. To this end, we present a dynamic classification model inspired by swarm intelligence. In our model, labeled instances are treated as ant colonies carrying different types of pheromone. Through a random walk among the unlabeled instances (or called unlabeled ants), the labeled ant colonies propagate their pheromone on the unlabeled ants and recruit them into their colonies. Experimental results demonstrate the effectiveness of our proposed algorithm on the real-world applications.

ACKNOWLEDGMENT

This work was supported in part by the Chinese National Natural Science Foundation under Grant nos. 61003180, 61379066 and 61103018, Natural Science Foundation of Education Department of Jiangsu Province under contracts 13KJB520026 and 09KJB20013, Natural Science Foundation of Jiangsu Province under contracts BK2010318 and BK2011442, Research Innovation Program for College Graduates of Jiangsu Province (CXLX12_0917), and the New Century Talent Project of Yangzhou University.

REFERENCES

[1] D. Chakrabarti, R. Kumar, and A. Tomkins: Evolutionary clustering. In KDD, pp. 554–560 (2006)

[2] Y. Chi, X. Song, D. Zhou, K. Hino et al: Evolutionary spectral clustering by incorporating temporal smoothness. In KDD, pp. 153C162 (2007)

[3] Wang P.,Wang H. X.,Wu X. C.,et al. A low-granularity classifier for data streams with concept drifts and biased class distribution. J. IEEE Transactions on Knowledge and Data Engineering, 19(9), 1202–1213 (2007)

[4] Gao J.,Ding B, L.,Han J. W.,et al: Classifying Data Streams with Skewed Class Distributions and Concept Drifts. J. IEEE Internet Computing, 12(6), 37–49 (2008)

[5] Anagnostopoulos C.,Tasoulis D. K.,Adams N. M.,et al: Temporally adaptive estimation of logistic classifiers on data streams, J. Advances in Data Analysis and Classification,3(3), 243–261 (2009)

[6] Kuncheva L. I.,Zliobaite I: On the window size for classification in changing environments, J. Intelligent Data Analysis, 13(6), 861–872 (2009)

[7] Peng Z.,Xingquan Z., JianlongT.,et al.: Classifier and Cluster Ensembles for Mining Concept Drifting Data Streams. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 1175–1180. Sydney, Australia, Dec 13-17 (2010)

[8] Zhang P., Li J.,Wang P.,et al. Enabling fast prediction for ensemble models on data streams. Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 177-185. San Diego, CA, USA, Aug 21-24 (2011)

[9] Peng Z,, Gao B. J.,Xingquan Z.,et al: Enabling Fast Lazy Learning for Data Streams. In: 2011 IEEE 11th International Conference on Data Mining (ICDM), pp. 932–941. Vancouver, Canada, Dec 11-14 (2011)

[10] Zhang P. , Gao B. J.,Liu P., et al.: A framework for application-driven classification of data streams, J. Neurocomput., 92(0), 170–182 (2012)

[11] Lines J.,Davis L. M.,Hills J. , et al: A shapelet transform for time series classification. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 289-297. Beijing, China, Aug 12-16 (2012)

[12] Masud M. M.,WoolamC.,Gao J. , et al: Facing the reality of data stream classification: coping with scarcity of labeled data, J. Knowledge and Information Systems, 33(1), 213–244 (2012)

[13] Li L. J.,Zou B.,Hu Q. H.,et al.: Dynamic classifier ensemble using classification confidence, J. Neurocomput., 99(1), 581–591 (2013)

[14] Yangqing Jia, Shuicheng Yan and Changshui Zhang: Semi-Supervised Classification on Evolutionary Data. In: Proceedings of the 21st international joint conference on Artifical intelligence (IJCAI'09), pp. 1083–1088 (2009)

[15] Borchani H.,Larranaga P.,Bielza C.: Classifying evolving data streams with partially labeled data, J. Intelligent Data Analysis, 15(5), 655–670 (2011).