Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04066-3

HYBRID ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING TECHNOLOGIES

Density-based semi-supervised online sequential extreme learning machine

Min Xia[1,2] · Jie Wang[2] · Jia Liu[2] · Liguo Weng[2] · Yiqing Xu[3]

Received: 17 September 2018 / Accepted: 28 January 2019

Abstract
This paper proposes a density-based semi-supervised online sequential extreme learning machine (D-SOS-ELM). The proposed method can realize online learning of unlabeled samples chunk by chunk. Local density and distance are used to measure the similarity of patterns, and the patterns with high confidence are selected by the 'follow' strategy for online learning, which can improve the accuracy of learning. Through continuous patterns selection, the proposed method ultimately achieves effective learning of unlabeled patterns. Furthermore, using local density and relative distance can effectively respond to the relationship between patterns. Compared with the traditional distance-based similarity measure, the ability to deal with complex data is improved. Empirical study on several standard benchmark data sets demonstrates that the proposed D-SOS-ELM model outperforms state-of-art methods in terms of accuracy.

Keywords Semi-supervised learning · Extreme learning machine · Online sequential learning · Fast density clustering

1 Introduction

In recent years, deep learning has attracted more and more attention from researchers in the field of machine learning. Deep learning has a strong ability to express features, especially for the feature representation of unstructured data. However, for a lot of structured data with low feature dimensions, the ability of deep learning is not better than shallow learning, and the efficiency of deep learning is lower than that of shallow learning. In reality, there is a large number of low-dimensional structured data. Therefore, it is necessary to establish an efficient learning model according to the characteristics of such data. Single-layer feedforward network (SLFN) is considered to be an effective shallow learning structure, which has been extensively studied in the past decades. The traditional back propagation algorithm [1] and the Levenberg–Marquardt algorithm [2] both use gradient method to optimize the weights of the network, so the network is easy to fall into the local optimum, and the learning rate is low. In the past two decades, the theory of extreme learning machine (ELM) [3–6] has gained more and more attention in the field of machine learning. Compared with existing methods, the hidden node parameters of *ELM* can be randomly generated by any continuous probability distribution without any prior knowledge, and the output weights are determined analytically by using the Moore–Penrose generalized inverse [7]. ELM is faster than the gradient descent iteration and the quadratic programming problem in standard SVM [8]. Compared with gradient-based algorithm, ELM is more efficient and usually leads to better generalization performance [9–11].

In the classical ELM optimization problem, batch mode is used for training. When the classical ELM deals with the data that arrives sequentially, batch learning must integrate new data and old data together to retrain the model, which is very time consuming. In order to solve the above problem, the online sequential extreme learning machine (OS-ELM) was proposed [12–16]. In OS-ELM, the training data are learned chunk by chunk, and the output weight is only

✉ Min Xia
xiamin@nuist.edu.cn

1 Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China

2 Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China

3 College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China

Published online: 04 February 2019

updated by newly arriving data. Although OS-ELM can address the problem of online learning, it must be trained based on a large number of labeled data. With the enhancement of data collection and storage capacity, data acquisition is getting easier, but it is difficult to label data sets. In reality, many data sets only contain very little labeled data, and the others are unlabeled data. If only labeled data are used, the useful information of unlabeled data is wasted; the obtained model is difficult to have a good generalization performance. Therefore, how to achieve efficient semi-supervised online learning has become a hot topic in the field of machine learning. In order to effectively utilize the information of unlabeled data, some semi-supervised ELM (SS-ELM) methods were proposed [17–21]. Based on the OS-ELM and SS-ELM, semi-supervised online sequential extreme learning (SOS-ELM) was proposed [22]. The above semi-supervised learning is realized by establishing graph Laplacian built from both labeled and unlabeled data [17]. In this process, the measure of pattern similarity is generally based on distance, and K-nearest neighbors (KNN) method is used to mark unlabeled patterns. The existing semi-supervised learning relies on the smoothness assumption and cluster assumption [23]. Smoothness assumption: the class labels of two patterns located close to each other in a dense data area are similar, that is, when two patterns can connected by edges in a dense data area, they have the same class labels under a high probability. Cluster assumption: the data tend to form discrete clusters, and points in the same cluster are more likely to share a label. The hypothesis of the effectiveness of distance-based semi-supervised learning algorithm is that the attributes of the same class are similar in distance, and there is no too much difference between neighbor instances. But many complex data sets do not satisfy this assumption. And for many complex data sets, distance-based methods cannot effectively achieve clustering [26]. In the above methods, unlabeled patterns are labeled once; especially when the number of labeled patterns is insufficient, the error rate is often too high, which leads to the deterioration of semi-supervised learning. At present, most of the semi-supervised methods are distance-based semi-supervised learning, and some methods are density-based semi-supervised learning. Most of the density-based semi-supervised learning methods measure the density of the category rather than the individual density of the pattern [24, 25]. At present, there is no semi-supervised learning method based on confidence ranking by density and distance.

Rodriguez's research [26] demonstrated that clustering data with complex features can be well clustered by local density and distance. In this method, the center of the cluster is such a kind of point: it is surrounded by many points (leading to a large local density) and is far away from the point where the local density is larger than its own. The method adopts the 'follow' strategy in clustering, and pattern is classified into cluster whose nearest neighbors' local density is larger than itself. Drawing on this theory, this paper proposes a semi-supervised online sequential extreme learning machine based on density. In this method, we select the unlabeled patterns with high confidence level and label these patterns to further update the network. Through continuous iteration, the effective learning of unlabeled samples is finally achieved. The proposed method has solved the problem of high error rate for unlabeled patterns in the previous SOS-ELM learning. The proposed model selects unlabeled patterns with high confidence level for learning, which effectively improves the learning accuracy. This method inherits the advantages of the Rodriguez's method, and uses the local density and mutual distance to express the similarity measure, and improves the processing ability of the complex data set. Results on several benchmark data sets show that our model outperforms state-of-art methods.

2 Review of ELM and OS-ELM

2.1 Extreme learning machine

The *ELM* is a flexible computing framework for a broad range of nonlinear problems. A single hidden-layer feed-forward network (*SLFN*) is the most widely used model in classifier modeling. The model is characterized by a network of three layers of simple processing units connected by acyclic links. The hidden layers can capture the nonlinear relationship among variables. Each layer consists of multiple neurons that are connected to neurons in adjacent layers. Suppose there are N distinct samples (X_i, Y_i), where $X_i = [x_i^1, x_i^2, \ldots, x_i^n]^{\mathrm{T}} \in \mathbb{R}^n$ and $Y_i = [y_i^1, y_i^2, \ldots, y_i^m]^{\mathrm{T}} \in \mathbb{R}^m$. The *SLFN* with L hidden neurons and an activation function vector $g(x) = (g_1(x), g_2(x), \ldots, g_L(x))$ are described as

$$\sum_{i=1}^{L} \beta_i g_i(W_i \cdot X_j + b_i) = Y_j, \quad j = 1, 2, \ldots, N, \tag{1}$$

where $W_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^{\mathrm{T}}$ is the weight vector connecting the ith hidden neuron and the input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^{\mathrm{T}}$ is the weight vector connecting the ith hidden neuron and the output neurons, and b_i is the threshold of the ith hidden neuron. The operation $W_i \cdot X_j$ is the inner product of W_i and X_j. If the SLFNs can approximate these N samples with a zero error, it is equivalent to $\sum_{i=1}^{N} \| Y_i - T_i \| = 0$. Thus, there also exist parameters W_i, β_i, and b_i such that

$$\sum_{i=1}^{L} \beta_i g_i(W_i \cdot X_j + b_i) = T_j, \quad j = 1, 2, \ldots, N, \tag{2}$$

Thus, above equations can be compactly described as

$$\mathbf{H}\beta = T, \tag{3}$$

where $\mathbf{H}$ is the hidden-layer output matrix, the ith column of $\mathbf{H}$ denotes the ith hidden neuron output with respect to inputs $X_1, X_2, \ldots, X_N$

$$\mathbf{H} = \begin{bmatrix} g_1(W_1 \cdot X_1 + b_1) & \ldots & g_L(W_L \cdot X_1 + b_L) \\ \vdots & \ldots & \vdots \\ g_1(W_1 \cdot X_N + b_1) & \ldots & g_L(W_L \cdot X_N + b_L) \end{bmatrix}_{N \times L} \tag{4}$$

Unlike the traditional function approximation theories which require to adjust input weights and hidden-layer biases, the input weights and hidden biases are randomly generated. Thus, training an *SLFN* is simply equivalent to finding a least squares solution $\hat{\beta}$ of the linear function $\mathbf{H}\beta = Y$:

$$\| \mathbf{H}\hat{\beta} - Y \|^2 = \min_{\beta} \| \mathbf{H}\beta - Y \|^2, \tag{5}$$

If regularized optimization is used, the optimal weights are given by the solution of the following optimization problem

$$\min_{\beta} \frac{1}{2} \| \mathbf{H}\beta - Y \|^2 + \frac{C}{2} \| \beta \|^2, \tag{6}$$

where C is the regularization parameter. Solving this error function is equivalent to the ridge regression problem, and its solution is described as follows.

$$\hat{\beta} = \left(\frac{I}{C} + H^T H \right)^{-1} H^T Y \tag{7}$$

ELM can learn much faster with a higher generalization performance than the traditional gradient-based learning algorithms and solve the problems of stopping criteria, learning rate, learning epochs, local minima.

2.2 Online sequential extreme learning machine

In real applications, the patterns may arrive sequentially in a chunk-by-chunk manner. In the basic ELM algorithm, when the new data are added, all the existing data will be trained, which will lead to the increase in training time of the model. When the data set scales up, batch learning approaches' calculation is too large. In order to solve this problem, the online sequential extreme learning machine (OS-ELM) had been proposed by Liang [12].

Suppose that there is an initial training set $S_0 = \{(x_i^0, y_i^0) \mid x_i^0 \in \mathbb{R}^n, y_i^0 \in \mathbb{R}^m\}_{i=1}^{N_0}$. The initial model is trained using the batch ELM, and the first approximation of the weights is given by

$$\beta^{(0)} = P_0 H_0^T Y, \tag{8}$$

where $P_0 = \left(\frac{I}{C} + H_0^T H_0\right)^{-1}$. For the inverse to exist, $| S_0 | \geq rank(H_0)$ is needed, where $| S_0 |$ is the cardinality of set S_0. Suppose that the sequence chunk of data is available by $S_j = \{(x_i^j, y_i^j) \mid x_i^j \in \mathbb{R}^n, y_i^j \in \mathbb{R}^m\}_{i=1}^{N_j}, j = 1, 2, 3, \ldots, q$. For each new arrived batch S_k, the new weights of the $\beta^{(k)}$ are updated as

$$P_k = P_{k-1} - P_{k-1} H_k^T (I + H_k P_{k-1} H_k^T)^{-1} H_k P_{k-1}, \tag{9}$$

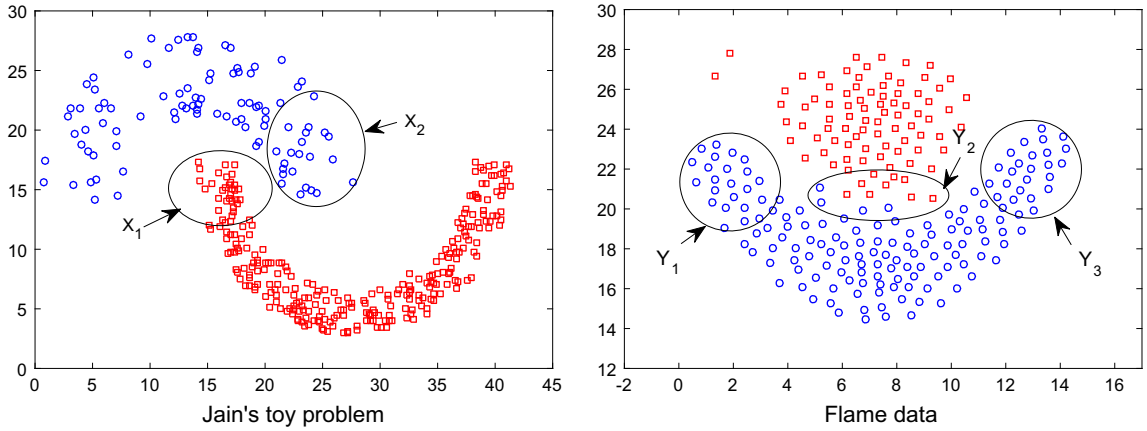
$$\beta^{(k)} = \beta^{(k-1)} + P_k H_k^T (Y_k - H_k \beta^{(k-1)}). \tag{10}$$

where H_k can be calculated by new added pattern set:

$$\mathbf{H_k} = \begin{bmatrix} g_1(W_1 \cdot X_1^k + b_1) & \ldots & g_L(W_L \cdot X_1^k + b_L) \\ \vdots & \ldots & \vdots \\ g_1(W_1 \cdot X_{N_k}^k + b_1) & \ldots & g_L(W_L \cdot X_{N_k}^k + b_L) \end{bmatrix}_{N_k \times L} \tag{11}$$

3 Proposed D-SOS-ELM

It is obvious that OS-ELM only utilizes labeled data. In order to use the sequential unlabeled data, Jia [22] proposed a semi-supervised online sequential extreme learning machine method based on Huang's work [17] and Liang's work [13], which integrates the both advantages from the online sequential learning and semi-supervised ELM. The previous semi-supervised learning is based on two assumptions: (1) both the labeled data and the unlabeled data are drawn from the same marginal distribution and (2) the change of data is generally smooth, that is to say if two points x_1 and x_2 are close to each other, then the conditional probabilities $P(y|x_1)$ and $P(y|x_2)$ should be similar as well. In the previous semi-supervised learning, the measure of pattern similarity was based on distance, but the distance cannot reflect the similarity of the patterns in many cases [26]. Figure 1 shows two examples that distance-based semi-supervised learning cannot handle. The point distribution of Jain's toy data [27] is shown in Fig. 1a, the unlabeled patterns in part X_1 and X_2 are easy to be misclassified when using the previous distance-based semi-supervised learning method, because the average distance from X_1 and X_2 to another category is smaller than that of their real classes. Another example of Flame data [28] in Fig. 1b also shows that the distance-based method may



(a) Point distributions of Jain's toy data

(b) Point distributions of Flame data

Fig. 1 Two examples (Jain's toy data [27] and Flame data [28]) that the distance-based method cannot handle for semi-supervised learning. Blue circles and red squares indicate two different categories. **a** X_1 and X_2 belong to the red squares category and the blue circle category, respectively, which are easily misclassified based on distance-based method; **b** both Y_1 and Y_3 belong to the blue circle category, Y_2 contains two types of points in different categories. The patterns of Y_1, Y_2, and Y_3 are easily misclassified based on distance-based method (color figure online)

cause the misclassification in the parts of Y_1, Y_2, and Y_3, making semi-supervised learning worse. Taking this cue, a new SOS-ELM based on density and distance is proposed in this work, which can effectively solve the above problem.

In this work, we apply the idea of density and distance to semi-supervised learning. Rodriguez's work [26] indicated that if pattern x_2 is the closest point to x_1 in all samples with a density larger than x_1, then the conditional probabilities $P(y|x_1)$ and $P(y|x_2)$ should be similar as well, which redefines the concept of smoothness of data set. Given a data set $S = \{x_i\}_{i=1}^{V}$, which includes both labeled data and unlabeled data. In this work, the relationship between patterns is characterized by two parameters of density and distance. $d_{ij} = dist(x_i, x_j)$ is the distance between pattern x_i and pattern x_j, which can be described by Euclidean distance or Gaussian distance. In this work, the Gaussian distance is used to represent the distance between patterns, which is defined as following equation.

$$d_{ij} = dist(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\delta^2}} \tag{12}$$

where δ is the scale parameter. In this work, we use the Gaussian kernel to describe the pattern's density[1]. The Gaussian density is defined as:

$$\rho_i = \sum_{j \in I_s \backslash i} e^{-\left(\frac{d_{ij}}{d_c}\right)^2} \tag{13}$$

[1] The cutoff kernel can also be used for density description, but this method can lead to the same density of many points, which brings difficulties to subsequent operations. (cutoff kernel: $\rho_i = \sum_{j \in I_s \backslash i} \chi(d_{ij} - d_c)$, where $\chi(x) = 1$ if $x < 0$, else $\chi(x) = 0$).

where ρ_i is the density of pattern i, $I_s = \{1, 2, ..., V\}$ is the tag set. $d_c > 0$ is the cutoff distance. For a labeled pattern x_i, if the unlabeled pattern x_j is the closest pattern to x_i and the density is larger than that of x_i, then the conditional probabilities $P(y|x_j)$ and $P(y|x_i)$ should be similar. On the other hand, for a unlabeled pattern x_p, if the labeled pattern x_q is the closest pattern to x_p and the density is larger than that of x_p, then the conditional probabilities $P(y|x_p)$ and $P(y|x_q)$ should also be similar.

Suppose there is a data set $S = \{s_i\}_{i=1}^{N}$ including labeled set $S^l = \{s_i^l\}_{i=1}^{N_l}$ and unlabeled set $S^u = \{s_i^u\}_{i=1}^{N_u}$ satisfying $N_l + N_u = N$. Based on the above assumptions, the proposed density-based semi-supervised online sequential extreme learning machine (D-SOS-ELM) in this paper can be realized through the following steps:

Step 1: Data normalization. For real data sets, the features are independent, but multiple features may be very different in numerical value. Normalization of feature sets cannot only improve the computing speed, but also improve the learning accuracy.

Step 2: Calculate the distance matrix d_{ij} and determine the truncation distance d_c. Calculate the distance $d_{ij}(1 \le i, j \le N)$ between patterns i and j based on Eq. 12. The size of the cutoff distance d_c directly affects the experimental results. Too large d_c will make the density of each data larger and reduce the difference between patterns. On the contrary, if d_c is too small, the density of patterns will become very small, which reduces the separability between the patterns. Arrange the distance in

ascending order $d_1 \leq d_2 \leq \cdots \leq d_M \quad (M = \frac{1}{2}N(N-1))$, then in this work d_c is set as $d_k \; k = 2\% \times M$ [26][2].

Step 3: Calculate the density of each pattern according to Eq. 13.

Step 4: Network initialization. According to the labeled samples $S^l = \{s_i^l\}_{i=1}^{N_l}$, the number of input layer nodes, hidden-layer nodes and output layer nodes is determined. Train initial network parameters P_0, H_0, and $\beta^{(0)}$ according to Eqs. 4 and 8.

Step 5: SOS-ELM based on density and distance.

5.1. Select unlabeled patterns with high confidence level. A temporary pattern set Z is established. For each labeled pattern s_i^l searching for a point, if the unlabeled pattern s_j^u is the closest pattern to s_i^l and the density is larger than that of s_i^l, then the unlabeled pattern s_j^u is included in Z. According to the principle of density clustering, the probability that s_i^l and s_j^u belong to the same class is the highest[3]. On the other hand, for each unlabeled pattern s_i^u, if the labeled pattern s_j^l is the closest pattern to s_i^u and the density is larger than that of s_i^u, then the unlabeled pattern s_i^u is included in Z.

5.2. Unlabeled patterns in Z are inserted into the neural network to generate labels. Remove the patterns in Z from set S^u.

5.3. Update the network model with the result of the labeled patterns in 5.2. Update P and β based on Eqs. 9 and 10. Put the patterns in Z into the pattern set S^l, and empty the pattern set Z.

5.4. If the pattern set S^u is empty, terminate the network training, else transfer to 5.1.

The algorithm flow is described in Fig. 2. Through the calculation of the pattern's density and distance, the unlabeled patterns with high confidence are selected as the patterns for semi-supervised learning, and then continue to be iterated until all the unlabeled patterns are learned. The proposed method is more conducive to improving the reliability and accuracy of network learning.

[2] As a rule of thumb, one can choose d_c so that the average number of neighbors is around 1% to 2% of the total number of points in the data set. If d_c is too large, the density of each data will be so large that the discrimination is not high. If the value is too small, then one category may be split into multiples, and the extreme case is that each of the data is a separate class. Locking the range at 1–2% is the empirical value of Rodriguez's work based on several data sets and multiple trials.

[3] Rodriguez's work suggests that in density clustering, a pattern X should belong to the same class as the pattern closest to the pattern X and with a larger density than pattern X.

4 Numerical simulations studies

In order to verify the effectiveness of the semi-supervised online learning method based on density, several comparative experiments are conducted. Firstly, this work compares the proposed method with the traditional semi-supervised online learning method in the Jain's toy data set and the Flame data data set to verify that the proposed method can effectively inherit the advantages of density clustering method. In this work, the semi-supervised extreme learning algorithm (SS-ELM) proposed by Huang [17] is compared with the algorithm proposed in this paper. This work firstly selects 10% of the data as labeled data, 40% as unlabeled data, and 50% as test data. Figure 3 demonstrates the comparison diagram of the proposed D-SOS-ELM method and SS-ELM method. It is obvious that the SS-ELM algorithm mistook the red point as blue point for the Jain 's toy problem data set, and the SS-ELM algorithm produces more misjudgments in the upper left corner of Fig. 3c for the Flame data set. Here, the average accuracy of 20 independent experiments is taken as the classification accuracy. On the Jain's toy problem data set, the average test accuracy of proposed algorithm is 98.40%, and the average test accuracy of SS-ELM algorithm is 85.03%. For the Flame data set, the average test accuracy of D-SOS-ELM method is 98.35%, and 90.83% for SS-ELM algorithm. Figure 3 shows that the distribution of Jain's toy problem data set and Flame data data set is irregular. Therefore, the traditional distance-based method is difficult to get good learning results. The density-based method can well reflect the distribution of samples, so it can achieve better learning results than distance-based method. The simulations indicate that the proposed D-SOS-ELM method inherits the advantages of density clustering method.

In order to further verify the performance of the density-based online semi-supervised classification algorithm, another three sets of experimental comparisons are given. This paper uses numerical real data sets selected from UCI machine learning repository, and data properties are shown in Table 1.

According to Theorem 2.2 in literature [29], the upper bound of the required number of hidden nodes is the number of distinct training samples. In this work, we choose the number of hidden neurons following this rule. Among the choice of the number of hidden-layer nodes, we first set a smaller number of nodes to initialize a smaller network, and then continuously add new nodes until a satisfactory network is generated. We have tried many times to select a better parameter through cross-validation, and generate a satisfactory network, which improves the stability and generalization ability of the network. The

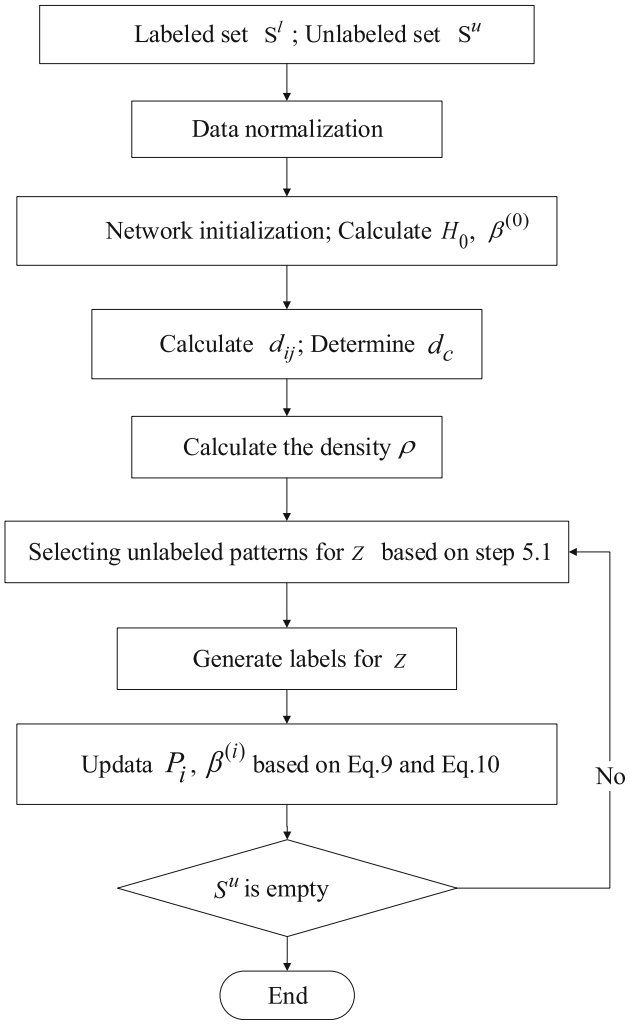


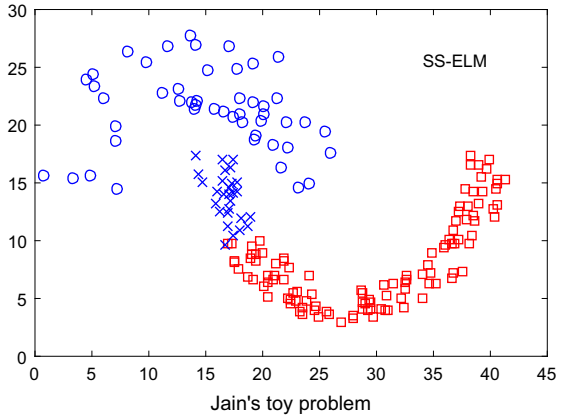
Fig. 2 The algorithm flow chart of proposed D-SOS-ELM

number of hidden-layer nodes for different data sets of D-SOS-ELM algorithm is described in Table 2. In this work, the number of hidden-layer nodes of ELM and SS-ELM is set same as D-SOS-ELM.

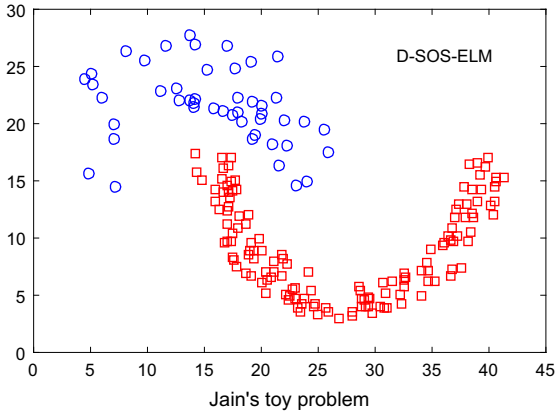
All the tests in this article randomly divide the data set into 3 parts, the first part is 10% of the data set, the second part is 70% of the data set, and the third part is the 20% of the data set. In order to further verify the D-SOS-ELM, we compare the D-SOS-ELM algorithm with the traditional ELM method. Table 3 shows the average classification accuracy of the 100 tests for different training methods. The proposed D-SOS-ELM algorithm selects the first part as labeled data to initialize the model, selects second part as unlabeled data for auxiliary training, and selects third part of data as a test set. The traditional ELM model is divided into two types of training: one selects the first part of labeled data sets (ELM_{TestA}) to train the model, and another simulation uses the first part and second part labeled data (ELM_{TestB}). All simulations are tested with the third part of data. The data sets are randomly scattered before each experiment.

It can be seen from Table 3 that ELM_{TestB} has the best experimental result, ELM_{TestA} is the worst. In terms of the number of patterns used in training, the ELM_{TestA} model is the least, so the result is the worst. In this paper, the number of training patterns for D-SOS-ELM is the same as that of ELM_{TestB}. The difference is that the ELM_{TestB} model is trained by labeled patterns, while D-SOS-ELM has only a very small number of real labeled data and most of the patterns are unlabeled. Thus, the result of D-SOS-ELM method is not as good as that of ELM_{TestB}. Moreover, the accuracy of D-SOS-ELM method is close to the accuracy of ELM_{TestB}. Based on density and distance, the new patterns are screened from the second unlabeled part of data set, and the labels are predicted to retrain the model. Therefore, the performance of the D-SOS-ELM model is better than that of ELM_{TestA}. Figure 4 shows the comparison of the 15 experiments in three ways. Abscissa represents the number of experiments. In Fig. 4, the curve of ELM_{TestB} is the most stable and ELM_{TestA} fluctuates greatly. The D-SOS-ELM algorithm is basically in the middle of two curves of ELM_{TestA} and ELM_{TestB}.

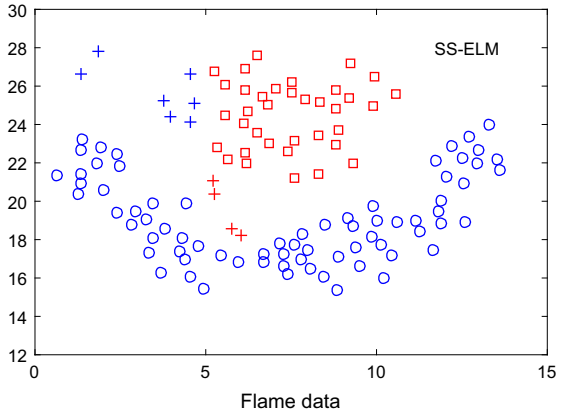
In above simulations, the proposed method is tested with labeled data equal to 10%. Using different proportion of labeled patterns for training, the results are different. In this work, four different proportion of labeled patterns are used for comparison. Test 1 is with 5% labeled data and 75% unlabeled data for training; Test 2 is with 10% labeled data and 70% unlabeled data for training; Test 3 is with 20% labeled data and 60% unlabeled data for training; Test 4 is with 50% labeled data and 30% unlabeled data for training. For all simulations, remaining 20% data are used for testing. The average accuracy comparison of the 100 tests is described in Table 4. Table 4 shows that the classification accuracy of Test 4 is the highest. The results of Test 3 and Test 4 are very close, and the average test accuracy of Test 4 on 9 data sets is 1.97% higher than that of Test 3. However, the average test accuracy of Test 2 on 9 data sets is 11.34% higher than that of Test 1. Figure 5 shows the comparison of the 15 experiments in four different learning ways. It can be seen that the curve of Test 2 has become smoother and more stable than Test 1. The above results indicate that when the number of labeled patterns reaches more than 10%, the online semi-supervised learning method in this paper can effectively approach the accuracy of supervised learning. When the number of labeled patterns is less than 10%, network initialization is prone to over-fitting, which affects the subsequent semi-supervised learning. The larger the data set, the lower the proportion of labeled samples required.



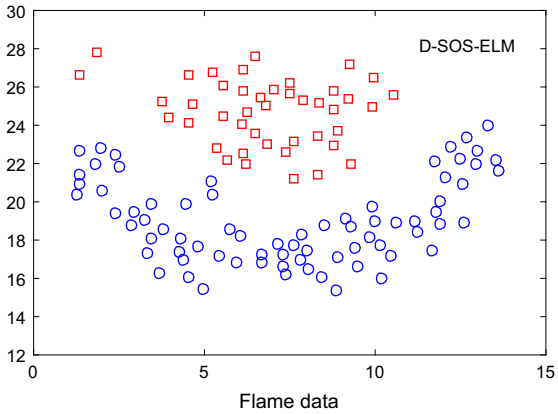
(a) Classification result using SS-ELM



(b) Classification result using D-SOS-ELM



(c) Classification result using SS-ELM



(d) Classification result using D-SOS-ELM

Fig. 3 The comparison results of D-SOS-ELM method and SS-ELM method on Jain's toy problem data set and Flame data data set. The blue cross represents the pattern that red square pattern is wrongly divided into blue circular pattern; the red cross represents the pattern that blue circular pattern is wrongly divided into red square pattern (color figure online)

Table 1 Specifications of benchmark data sets

Data set	Instances	Attributes	Classes
Iris	150	4	3
Haberman	306	3	2
Databanknote	1372	4	2
Transfusion	748	4	2
EEG Eye	1000	14	2
Wine	178	13	3
Seeds data set	210	7	3
Glass	214	9	6
Page-blocks	5473	10	5

Table 2 Numbers of hidden-layer nodes for different data sets of D-SOS-ELM algorithm

Data set	Number of hidden nodes
Iris	7
Haberman	5
Databanknote	10
Transfusion	9
EEG Eye	15
Wine	8
Seeds data set	7
Glass	9
Page-blocks	20

Finally, we will use the following four semi-supervised learning methods to make a comparative analysis with D-SOS-ELM: Selflearning, SemiEM, MultiRankWalk, SS-ELM, and STAR-SVM.

(1) Selflearning [30]: Semi-supervised selflearning framework, use Sklearn's SGD classifier linear model to train labeled data sets. Then the unlabeled patterns are labeled by trained model, and patterns

Table 3 Comparison of average classification accuracy of the 100 tests for D-SOS-ELM and ELM (%)

Data set	D-SOS-ELM	ELM_{TestA}	ELM_{TestB}
Iris	91.33	81.78	**96.89**
Haberman	73.88	65.79	**77.38**
Databanknote	98.52	94.79	**98.54**
Transfusion	78.44	73.82	**80.49**
EEG Eye	99.30	97.60	**99.93**
Wine	87.04	63.89	**94.26**
Seeds	90.00	74.76	**93.97**
Glass	55.66	28.68	**67.29**
Page-blocks	95.58	94.61	**96.28**

Bold values represent the optimal values under different methods

Table 4 Average accuracy comparison of the 100 tests under different percentages of labeled data for training (%)

Data set	Test 1	Test 2	Test 3	Test 4
Iris	74.89	91.33	94.00	**96.22**
Haberman	67.43	73.88	74.16	**74.43**
Databanknote	93.75	98.52	**98.53**	**98.53**
Transfusion	75.11	78.44	78.62	**79.87**
EEG Eye	96.30	99.30	99.30	**99.60**
Wine	73.33	87.04	91.11	**93.96**
Seeds	81.27	90.00	92.38	**93.63**
Glass	43.72	55.66	57.98	**62.93**
Page-blocks	94.50	95.58	95.62	**95.72**

Bold values represent the optimal values under different methods

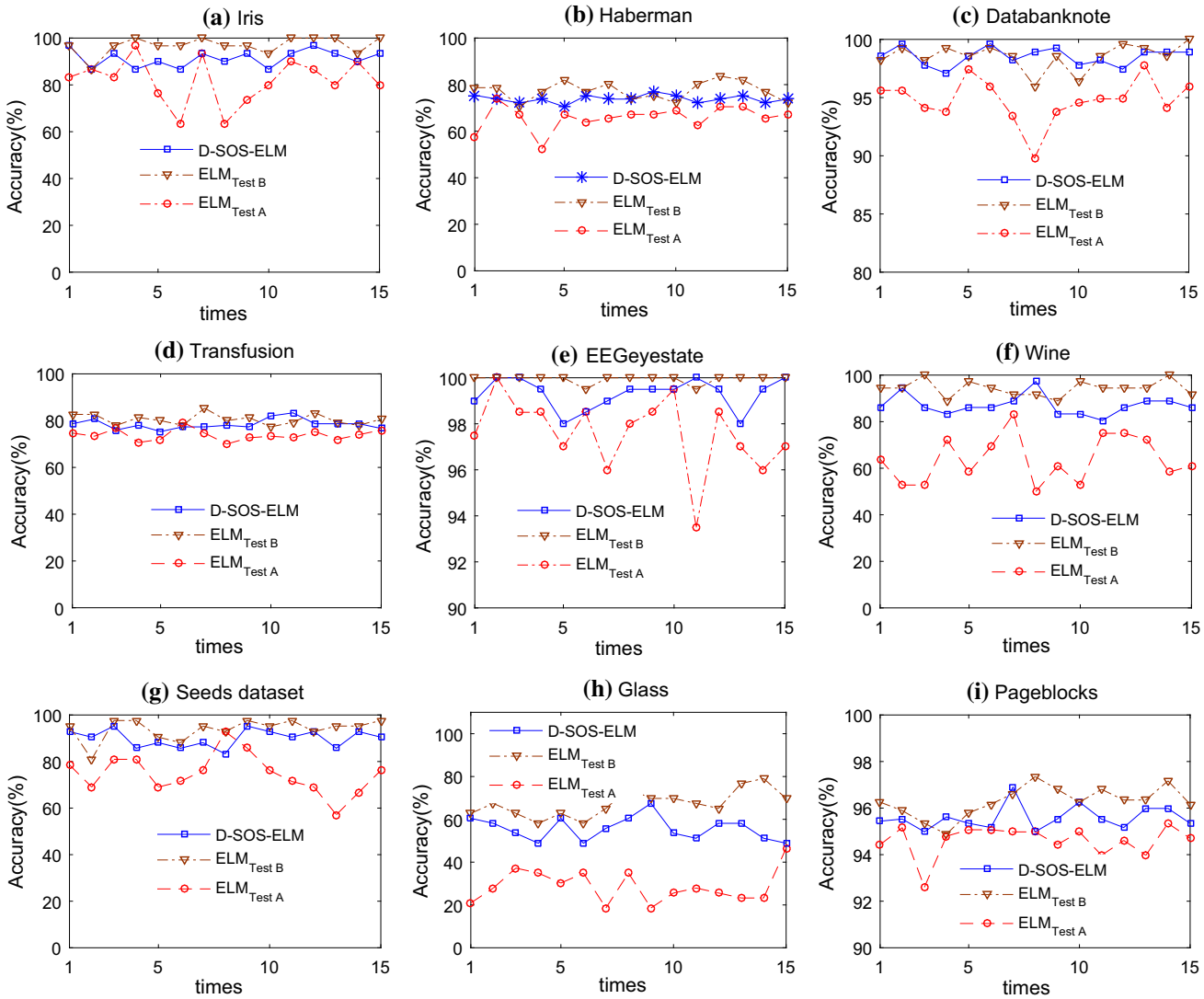


Fig. 4 Comparison of classification results under three cases. The selection of training set and testing set is the same as that of Table 3

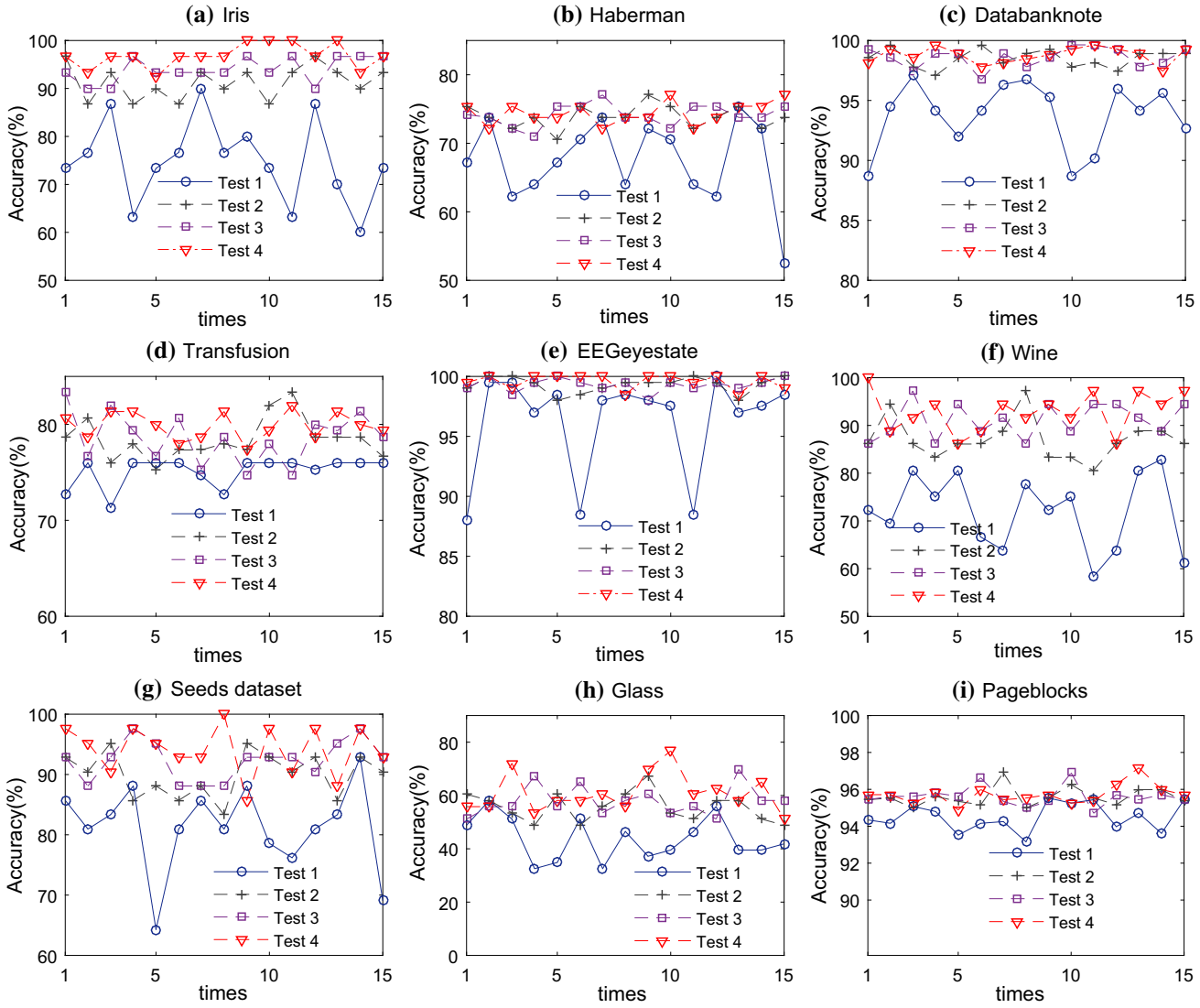


Fig. 5 Comparison of classification results under four cases. The selection of training set and testing set is the same as that of Table 4

with high confidence level are retrained until convergence.

(2) SemiEM [31]: Semi-supervised classification is implemented using EM and polynomial Bayes. Firstly, a Naive Bayesian classifier is trained with the labeled patterns, then the unlabeled data are divided into a reliable set and an unreliable set. The labeled patterns and the reliable patterns are used to update the classifier. Above process is repeated, until the model converges.

(3) MultiRankWalk (MRW)[32]: This method describes the patterns with a graph, the node is the pattern, and the edge indicates the similarity or relation between the patterns. For each labeled pattern, random walks are used to find similar unlabeled patterns, and repeat the above process. For each unlabeled pattern, the label is determined by class with the largest number that finds it.

(4) SS-ELM [17]: The semi-supervised learning algorithm is based on ELM. Use labeled data and unlabeled data to construct Laplace operator, initialize the ELM model, then iteratively select the best until convergence.

(5) STAR-SVM [33]: STAR-SVM is a semi-supervised learning classifier designed to adaptively modify the optimization by adjusting the weights at each iteration. At each iteration, the regularization parameters are adapted to better reflect label confidence, class proportion, and to gradually include more unlabeled points.

Table 5 Comparison of average classification accuracy of the 100 tests. (%)

Data set	D-SOS-ELM	SemiEm	Selflearning	MRW	SS-ELM	STAR-SVM
Iris	**91.33**	65.78	60.00	78.44	84.30	69.36
Haberman	**73.88**	69.03	64.07	53.18	71.22	72.08
Databanknote	**98.52**	70.82	97.97	73.01	95.73	96.90
Transfusion	**78.44**	75.67	60.98	72.99	73.65	74.87
EEG Eye	**99.30**	56.63	52.00	59.85	67.07	94.16
Wine	**87.04**	58.89	49.33	44.01	84.44	67.29
Seeds	**90.00**	39.05	53.21	84.29	85.89	85.96
Glass	**55.66**	37.36	32.99	39.28	44.74	49.31
Page-blocks	**95.58**	91.54	90.55	83.95	84.23	92.82

Bold values represent the optimal values under different methods

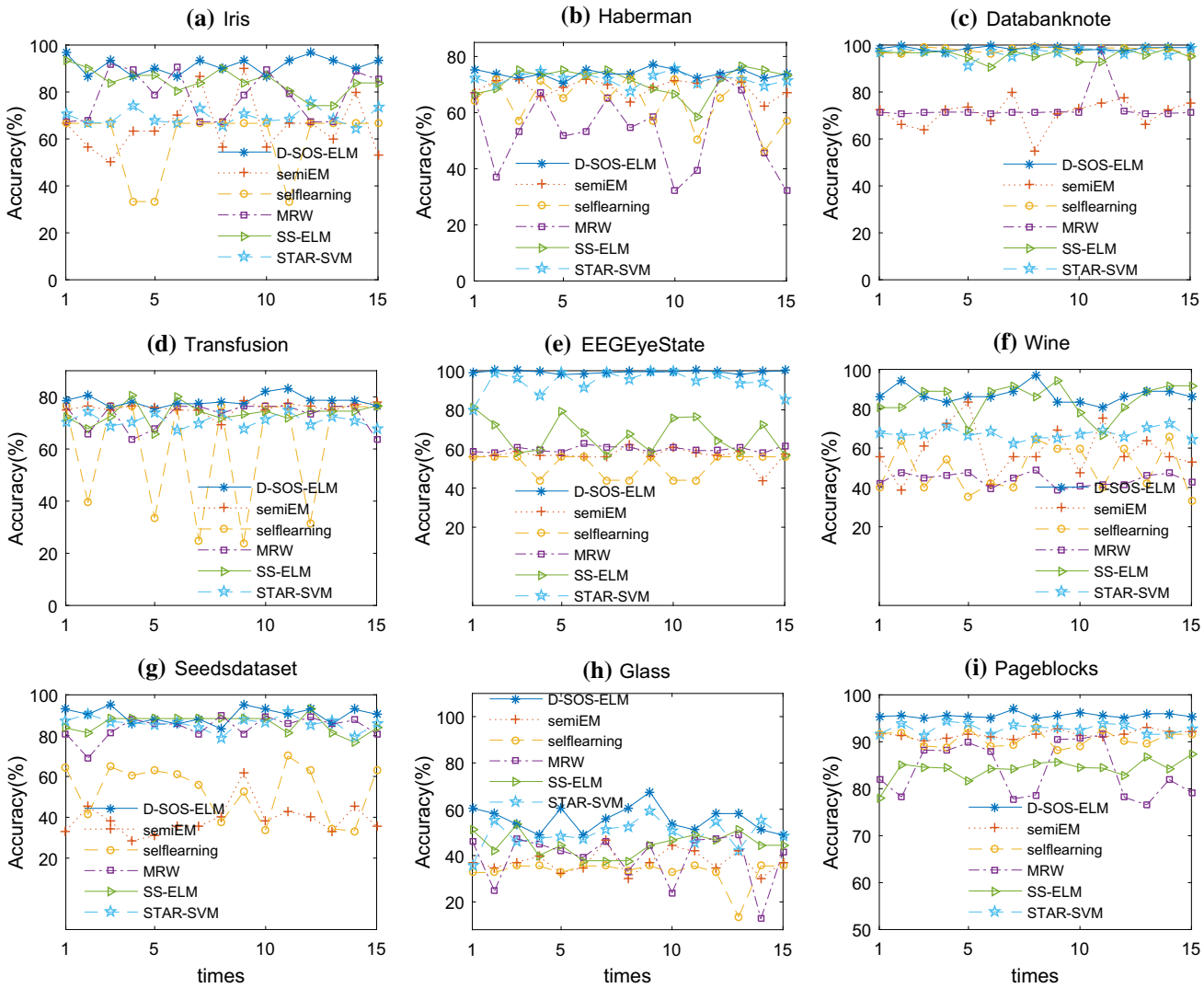


Fig. 6 Comparison of classification results of D-SOS-ELM, SemiEm, Selflearning, MRW, SS-ELM, and STAR-SVM

Table 5 shows the comparison results of six semi-supervised learning methods. It's obvious that D-SOS-ELM is better than other algorithms. The average test accuracy of density-based online semi-supervised extreme learning algorithm on 9 data sets is 13.07% higher than that of traditional semi-supervised extreme learning algorithm. And the average test accuracy of the proposed model is 45.21%, 43.9%, 35.91%, and 12.75% higher than that of

Table 6 Training time (in seconds) comparison of D-SOS-ELM, SemiEm, Selflearning, MRW, SS-ELM, and STAR-SVM

Data set	D-SOS-ELM	SemiEm	Selflearning	MRW	SS-ELM	STAR-SVM
Iris	**0.07**	3.28	0.59	0.27	1.14	0.62
Haberman	**0.25**	3.26	0.71	0.39	1.73	0.54
Databanknote	3.35	3.28	**0.72**	2.61	2.79	1.39
Transfusion	1.20	3.25	**0.62**	0.97	1.93	1.69
EEG Eye	1.97	3.30	**0.75**	1.55	2.34	1.18
Wine	**0.09**	3.23	0.63	0.23	1.12	0.93
Seeds	**0.11**	3.27	0.66	0.29	1.08	0.68
Glass	**0.13**	3.23	0.64	0.31	1.11	0.93
Page-blocks	22.69	**3.48**	7.19	35.67	16.82	28.45

Bold values represent the optimal values under different methods

Selflearning, SemiEm, MRW, and STAR-SVM, respectively. Figure 6 shows accuracy data for 15 experiments and considering all methods. For all data sets, our proposed approach is more stable. This result is attributed to the excellent selection mechanism. Based on density-based clustering principle, D-SOS-ELM is able to accurately select the points with similar categories around each point, which helps to optimize the model correctly. In addition, the proposed method uses unlabeled patterns with high confidence to train at each iteration, which can improve the accuracy of learning. Although this method needs many iterations, the algorithm complexity is not very high. This is due to the use of online learning to update the model, each time only needs to be updated based on the new selected high confidence patterns, without the need to use all selected patterns to relearn the network. In order to evaluate the computational efficiency of the proposed algorithm, we present the training times for D-SOS-ELM, SemiEm, Selflearning, MRW, SS-ELM, and STAR-SVM on different data sets in Table 6. The test environment of experiments is as follows: CPU (2.30 GHz), Memory (8 GB), and Software (MATLAB R2016b). It can be observed that the training time of D-SOS-ELM is the fastest in five data sets. On the one hand, because the proposed method is based on distance and density, the calculation of the distance matrix is very time consuming. On the other hand, online learning of proposed model can effectively improve the speed of training.

5 Conclusion

In this work, a density-based online semi-supervised classification algorithm is proposed. The proposed algorithm uses the pattern distance and the density distribution to screen the unlabeled patterns with high confidence to join online learning to speed up the training efficiency of the learning model and improve the accuracy of the semi-supervised online learning. For many complex data sets, the traditional method of using distance as a similarity index is difficult to realize the correct calculation of the similarity between patterns, which leads to the low accuracy of semi-supervised learning. By analyzing the density and distance of the patterns, this method can effectively calculate the pattern similarity of the complex data set, and can evaluate the reliability of the unlabeled pattern. Because each iteration selects the patterns with high confidence for online semi-supervised learning, the accuracy of learning is greatly improved compared with the existing methods. The average test accuracy of proposed model on 9 data sets is 13.07%, 45.21%, 43.9%, 35.91%, and 12.75% higher than that of SS-ELM, Selflearning, SemiEm, MRW, and STAR-SVM, respectively. This method can make use of a small number of labeled data and a large number of unlabeled data to train a practicable model, which can improve the utilization of data, reduce the manufacturing cost of data labeling, and improve the efficiency of learning.

Acknowledgements This work is supported in part by, the National Natural Science Foundation of PR China (61773219, 61503192), Natural Science Foundation of Jiangsu Province (BK20161533), and Qing Lan Project of Jiangsu Province.

References

1. Mackay DJC (2014) A practical Bayesian framework for back-propagation networks. Neural Comput 4(3):448–472
2. Sarabakha A, Imanberdiyev N, Kayacan E et al (2017) Novel Levenberg–Marquardt based learning algorithm for unmanned aerial vehicles. Inf Sci 416:361–380
3. Ding S, Xu X, Nie R (2014) Extreme learning machine and its applications. Neural Comput Appl 25(3–4):549–556
4. Xue J, Zhou SH, Liu Q et al (2018) Financial time series prediction using l2, 1 RF-ELM. Neurocomputing 277(14):176–186
5. Yang C, Huang K, Cheng H et al (2017) Haptic identification by ELM-controlled uncertain manipulator. IEEE Trans Syst Man Cybern Syst 47(8):2398–2409
6. Martinez-Garcia JA, Sancho-Gomez JL (2018) Performance analysis of No-Propagation and ELM algorithms in classification. Neural Comput Appl 10–12:1–11

7. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
8. Huang GB (2014) An insight into extreme learning machines: random neurons, random features and kernels. Cogn Comput 6(3):376–390
9. Li X, Mao W, Jiang W (2016) Multiple-kernel-learning-based extreme learning machine for classification design. Neural Comput Appl 27(1):175–184
10. Huang GB, Zhou H, Ding X et al (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B 42(2):513–529
11. Zou W, Yao F, Zhang B et al (2017) Improved Meta-ELM with error feedback incremental ELM as hidden nodes. Neural Comput Appl 8:1–8
12. Liang NY, Huang GB, Saratchandran P et al (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17(6):1411–1423
13. Zhao J, Wang Z, Dong SP (2012) Online sequential extreme learning machine with forgetting mechanism. Neurocomputing 87(15):79–89
14. Scardapane S, Comminiello D, Scarpiniti M et al (2015) Online sequential extreme learning machine with kernels. IEEE Trans Neural Netw Learn Syst 26(9):2214–2220
15. Zou QY, Wang XJ, Zhou CJ et al (2018) The memory degradation based online sequential extreme learning machine. Neurocomputing 275:2864–2879
16. Wang B, Huang S, Qiu J et al (2015) Parallel online sequential extreme learning machine based on MapReduce. Neurocomputing 149(PA):224–232
17. Huang G, Song S, Gupta JN et al (2014) Semi-supervised and unsupervised extreme learning machines. IEEE Trans Cybern 44(12):2405–2417
18. Liu S, Feng L, Wang H et al (2016) Extend semi-supervised ELM and a frame work. Neural Comput Appl 27(1):205–213
19. Krishnasamy G, Paramesran R (2016) Hessian semi-supervised extreme learning machine. Neurocomputing 207(C):560–567
20. Bisio F, Decherchi S, Gastaldo P et al (2016) Inductive bias for semi-supervised extreme learning machine. Neurocomputing 174(PA):154–167
21. Yi Y, Qiao S, Zhou W et al (2018) Adaptive multiple graph regularized semi-supervised extreme learning machine. Soft Comput 22(6):1–18
22. Jia X, Wang R, Liu J et al (2016) A semi-supervised online sequential extreme learning machine method. Neurocomputing 174(PA):168–178
23. Olivier C, Scholkopf B, Alexander Z (2006) A discussion of semi-supervised learning and transduction. In: Semi-supervised learning. MIT Press, New York
24. Kawakita M, Kanamori T (2013) Semi-supervised learning with density-ratio estimation. Mach Learn 91(2):189–209
25. Soares RGF, Chen H, Yao X (2018) Efficient cluster-based boosting for semisupervised classification. IEEE Trans Neural Netw Learn Syst 2018(99):1–14
26. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):1492
27. Jain A, Law M (2005) Data clustering: a user's dilemma. Lect Notes Comput Sci 3776:1–10
28. Fu L, Medico E (2007) FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. BMC Bioinform 8(1):3
29. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
30. Pedregosa F, Gramfort A, Michel V et al (2013) Scikit-learn: machine learning in Python. J Mach Learn Res 12(10):2825–2830
31. Nigam K, Mccallum A, Mitchell TM (2006) Semi-supervised text classification using EM. AIAA J 36(36):62–68
32. Lin F, Cohen WW (2010) Semi-supervised classification of network data using very few labels. In: International conference on advances in social networks analysis and mining, pp 192–199
33. Cheung E, Li Y (2017) Self-training with adaptive regularization for S3VM. In: 2017 international joint conference on neural networks (IJCNN), Anchorage, AK, pp 3633–3640