

Research Article

Semisupervised Particle Swarm Optimization for Classification

Xiangrong Zhang,¹ Licheng Jiao,¹ Anand Paul,² Yongfu Yuan,¹
Zhengli Wei,¹ and Qiang Song¹

¹ Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China

² School of Computer Science Engineering, Kyungpook National University, Daegu 702-701, Republic of Korea

Correspondence should be addressed to Xiangrong Zhang; xrzhang@mail.xidian.edu.cn and Anand Paul; anand@knu.ac.kr

Received 14 February 2014; Accepted 29 April 2014; Published 28 May 2014

Academic Editor: Albert Victoire

Copyright © 2014 Xiangrong Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A semisupervised classification method based on particle swarm optimization (PSO) is proposed. The semisupervised PSO simultaneously uses limited labeled samples and large amounts of unlabeled samples to find a collection of prototypes (or centroids) that are considered to precisely represent the patterns of the whole data, and then, in principle of the “nearest neighborhood,” the unlabeled data can be classified with the obtained prototypes. In order to validate the performance of the proposed method, we compare the classification accuracy of PSO classifier, k-nearest neighbor algorithm, and support vector machine on six UCI datasets, four typical artificial datasets, and the USPS handwritten dataset. Experimental results demonstrate that the proposed method has good performance even with very limited labeled samples due to the usage of both discriminant information provided by labeled samples and the structure information provided by unlabeled samples.

1. Introduction

The particle swarm optimization (PSO) [1, 2], originally proposed by Eberhart and Kennedy, is a population-based stochastic search process. It is inspired by the social interaction behavior of birds flocking and fish schooling. In the context of PSO, a swarm refers to a number of potential solutions to the optimization problem, where each particle represents a potential solution. The particles fly through the search space with a velocity that is dynamically adjusted according to its local information (the cognitive component) and neighbor information (the social component), and trend to fly toward better and better search areas [3]. PSO has been widely applied to acquire the solution to the machine learning problems involved in various fields [4, 5]. In the area of machine learning, traditional learning methods can be divided into two categories: supervised learning and unsupervised learning. In many of the real-world applications of machine learning, a certain number of labeled samples are usually needed to train a classifier so as to perform the learning process, which is usually named as the supervised learning, such as the decision tree [6],

support vector machines (SVMs) [7–10], neural network classifier [11], and the PSO-based classifiers [12–17]. As a matter of fact, labeled instances are difficult, expensive, and time consuming to obtain. In cases that only unlabeled samples are available, the learning can be achieved in an unsupervised way, such as K-means clustering and fuzzy C-means clustering. It is often the case that both labeled and unlabeled samples are available but the labeled samples are too limited to obtain a favorable performance in supervised way, while abundant unlabeled samples are easy to obtain. Therefore, semisupervised learning [18] and reinforcement learning [19] are introduced and have been proved to be quite promising. Semisupervised learning tries to improve the performance via combining limited labeled samples and large amounts of unlabeled ones to perform the classification. It has recently become more and more popular among the variety of problems such as text classification [20], mail category [21], and human action recognition [22], for which the labeled samples are highly limited.

The nearest neighbor (NN) classification is one of the popular classification methods. It is a “lazy” learning method because it does not train the classifier using labeled training

data in advance [23]. The nearest neighbor decision rule assigns an unknown input sample vector to the class label of its nearest neighbor [24], which is measured in terms of a distance defined in the feature space. In this space, each class defines a region, which is called the Voronoi region [25]. When the distance is defined as the classical Euclidean distance, the Voronoi regions are delimited by linear borders. This method can be extended to the K-nearest neighbors when more than one nearest neighbor is considered. In addition, some other distance measures other than the Euclidean distance also can be used.

A further improvement of NN method replaces the original training data by a set of prototypes that correctly “represent” the original data [23]. Namely, the classifier assigns class labels by calculating distances to the prototypes rather than to the original training data. As the number of prototypes is much smaller than the total number of original training data sets, classification of new sample is performed much faster due to the reduced computational complexity of the solution (measured by the number of prototypes). These nearest prototype algorithms are able to achieve better accuracy of the solution than the basic NN classifiers. An evolutionary approach to the prototype selection problem can be found in [26].

PSO has been used in classification in many literatures. Most of the PSO-based classification methods combine PSO with an existing machine learning or classification algorithm such as NN [12, 27], neural network [13], and rough set theory [28]. In [12], an unsupervised learning algorithm is proposed by minimizing the distances within clusters. In [13], an evolutionary approach-based nearest prototype classifier is introduced. In [27], PSO is applied to find the optimal positions of class centroids in the feature space of dataset using the examples contained in the training set.

PSO has shown competitive performance in classification problem. However, it usually needs many labeled data points to obtain the optimal positions of class centroids. Semisupervised learning provides a better solution by making full use of the abundant unlabeled data along with the limited labeled samples to improve the accuracy and robustness of class predictions [18], and [29–31]. In this paper, we propose a semisupervised classification method based on the standard PSO, namely, semisupervised PSO (SSPSO), in which some available supervised information and the wealth of unlabeled data points are simultaneously used to search for the optimal positions of class centroids. The key point in SSPSO is to introduce the unlabeled information to the fitness function of PSO naturally. The advantages of SSPSO can be concluded as follows: firstly, it is a semisupervised learning method which can be applied with limited labeled samples; secondly, with less number of prototypes, the classification of new patterns will be performed faster; thirdly, SSPSO is able to achieve competitive or even better accuracy than the basic NN classifier and SVM.

The rest part of this paper is organized as follows. In Section 2, the theory of nearest neighbor classification and PSO is presented. The classification method based on PSO and our proposed SSPSO are described in Section 3. Experimental results and analysis on UCI datasets, some

typical artificial datasets, and the USPS handwritten dataset are shown and discussed in Section 4. Finally, Section 5 concludes this paper.

2. Review of Related Methods

2.1. K-Nearest Neighbor Algorithm. In pattern recognition, the k-nearest neighbor algorithm (KNN) is a simple method for classification. KNN is a type of lazy learning where the function is only approximated locally and all computation is deferred until classification. The simplest 1-NN algorithm assigns an unknown input sample to the class of its nearest neighbor from a stored labeled reference set. Instead of looking at the closest labeled sample, the KNN algorithm seeks k samples in the labeled reference set that are closest to the unknown sample and applies a voting mechanism to make a decision for label prediction.

Suppose $\mathbf{T} = \{(\mathbf{x}_i, y_i)\}$ is the training set, where $\mathbf{x}_i \in \mathcal{R}^d$ denotes the training example in a continuous multidimensional feature space and $y_i \in \mathcal{R}$ is class label of \mathbf{x}_i . For 1-NN classification, the class label of a test sample $\mathbf{x} \in \mathcal{R}^d$ can be obtained by finding the training example that is the nearest to \mathbf{x} according to some distance metrics, such as the Euclidean distance in (1), and assigning the class label of this training sample to it. For KNN classification, the class label of the test sample can be obtained with a method of majority voting. Consider

$$\|\mathbf{x} - \mathbf{x}_i\| = \sqrt{\sum_j (x_j - x_{ij})^2}. \quad (1)$$

2.2. Particle Swarm Optimization. PSO is based on a swarm of N individuals called particles, each representing a solution to the problem with D dimensions. Its genotype consists of $2D$ parameters, with the first D parameters representing the coordinates of particle’s position and the latter D parameters being its velocity components in the D -dimensional problem space, respectively. Besides the two basic properties, the following properties exist: a personal best position \mathbf{pbest}_i of each particle in the searching space and the global best position \mathbf{gbest} of the whole swarm. A fitness function corresponding to the problem is used to evaluate the “goodness” of each particle. Given a randomly initial position and velocity, the particles can be updated with the following:

$$\mathbf{v}_i^{(t+1)} = \omega \mathbf{v}_i^{(t)} + c_1 r_1 (\mathbf{pbest}_i^{(t)} - \mathbf{p}_i^{(t)}) + c_2 r_2 (\mathbf{gbest}^{(t)} - \mathbf{p}_i^{(t)}) \quad (2)$$

$$\mathbf{p}_i^{(t+1)} = \mathbf{p}_i^{(t)} + \mathbf{v}_i^{(t+1)}, \quad (3)$$

where $\mathbf{p}_i^{(t)}$ and $\mathbf{v}_i^{(t)}$ are the position and velocity of the i th particle at the t th iteration, respectively. The two positive factors c_1 and c_2 , known as the cognitive and social coefficients, control the contributions of the best local solution \mathbf{pbest}_i (cognitive component) and the global best solution \mathbf{gbest} (social component), respectively. r_1 and r_2 are two independent random variables within $[0, 1]$. The inertia

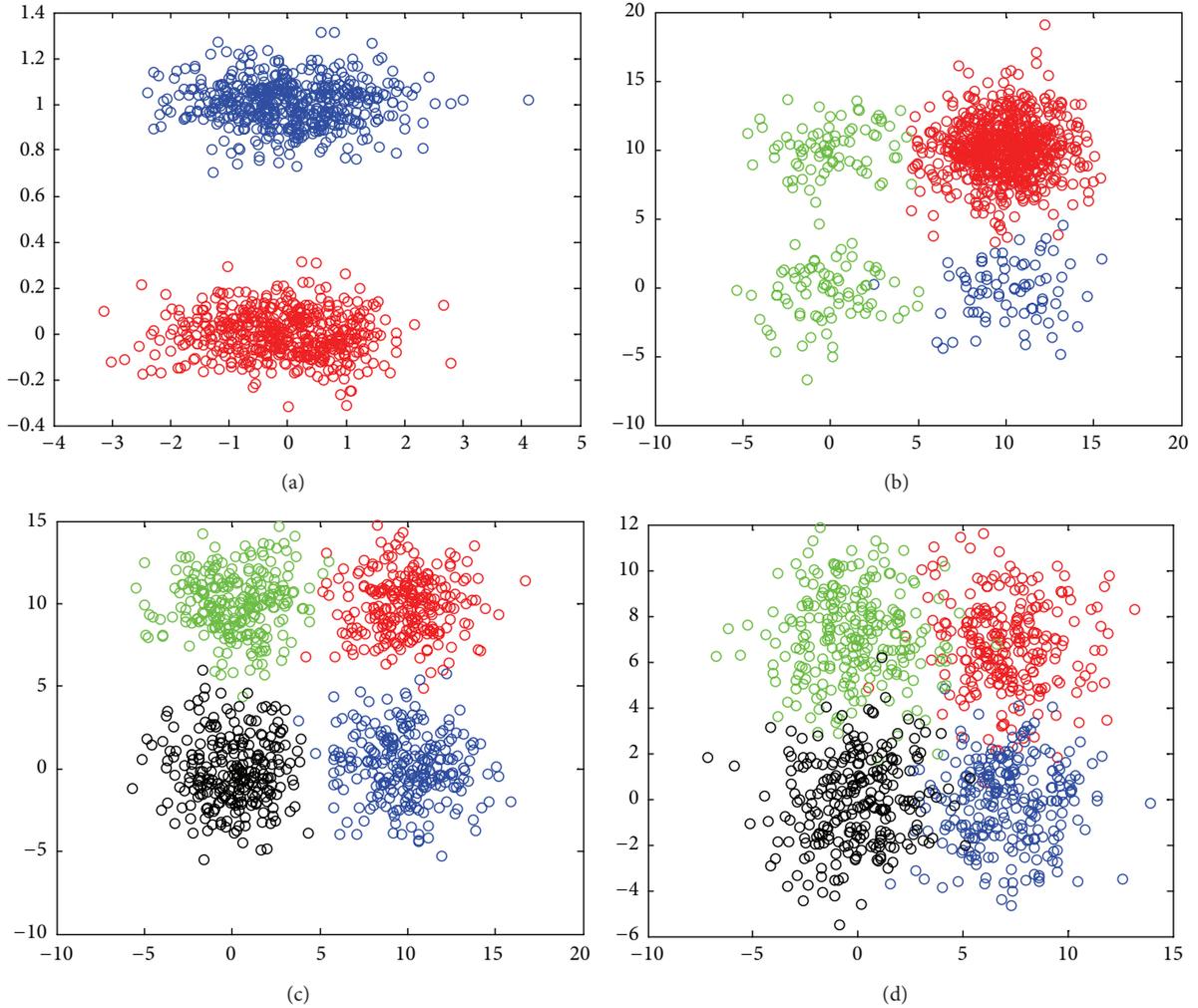


FIGURE 1: The distributions of the artificial datasets used in the experiment: (a) long1, (b) sizes5, (c) square1, and (d) square4.

weight factor ω is used to control the convergence of the swarm. In this paper, a nonlinear changing inertia factor for PSO and SSPSO is used as [32], which is shown in the following:

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})t}{T_{\max}}, \quad (4)$$

where T_{\max} is the maximum number of iterations and t is the current number of iteration. Note that, during the iteration, every dimension of the velocity is defined in the range $[-V_{\max}, V_{\max}]$ to limit the maximum distance that a particle will move.

3. Semisupervised Particle Swarm Optimization for Classification

In the context of PSO-based classification on the dataset \mathbf{X} with C classes and D attributes, classification problem can be seen as that of searching for the optimal positions for the C centroids of data clusters in a D -dimensional space with the labeled samples [23]. Then, NN method is applied as the

classifier that assigns class labels by calculating distances to the centroids to classify the unlabeled instances.

Data to be classified are a set of samples which are defined by continuous attributes, and the corresponding class is defined by a scalar value. Different attributes may take values in different ranges. To avoid one of attributes with large value dominating the distance measure, all the attributes are normalized to the range $[0, 1]$ before classification.

As a particle demotes a full solution to the classification of data with D attributes and C classes, C centroids are encoded in each particle. A centroid corresponds to a class, so it is defined by D continuous values for the attributes. Table 1 describes the structure of a single particle's position. Centroids are encoded sequentially in the particle, and a separate array determines the class of each centroid. Namely, the class for each centroid is defined by its position inside the particle. The total dimension of the position vector is $D * C$, and similarly the velocity of the particle is made up of $D * C$ real numbers representing its $D * C$ velocity components in the problem space. To simplify the notation for representation, we denote $\mathbf{p}_{k,i}$ as the k th class centroid vector $\mathbf{p}((k-1) * D + 1 : k * D)$ encoded in the i th particle.

TABLE 1: Encoding of a set of centroids in a particle for PSO.

	Centroid 1	...	Centroid k	...	Centroid C
Position	$p(1 : D)$...	$p((k - 1) * D + 1 : k * D)$...	$p((C - 1) * D + 1 : C * D)$
Class	1	...	k	...	C

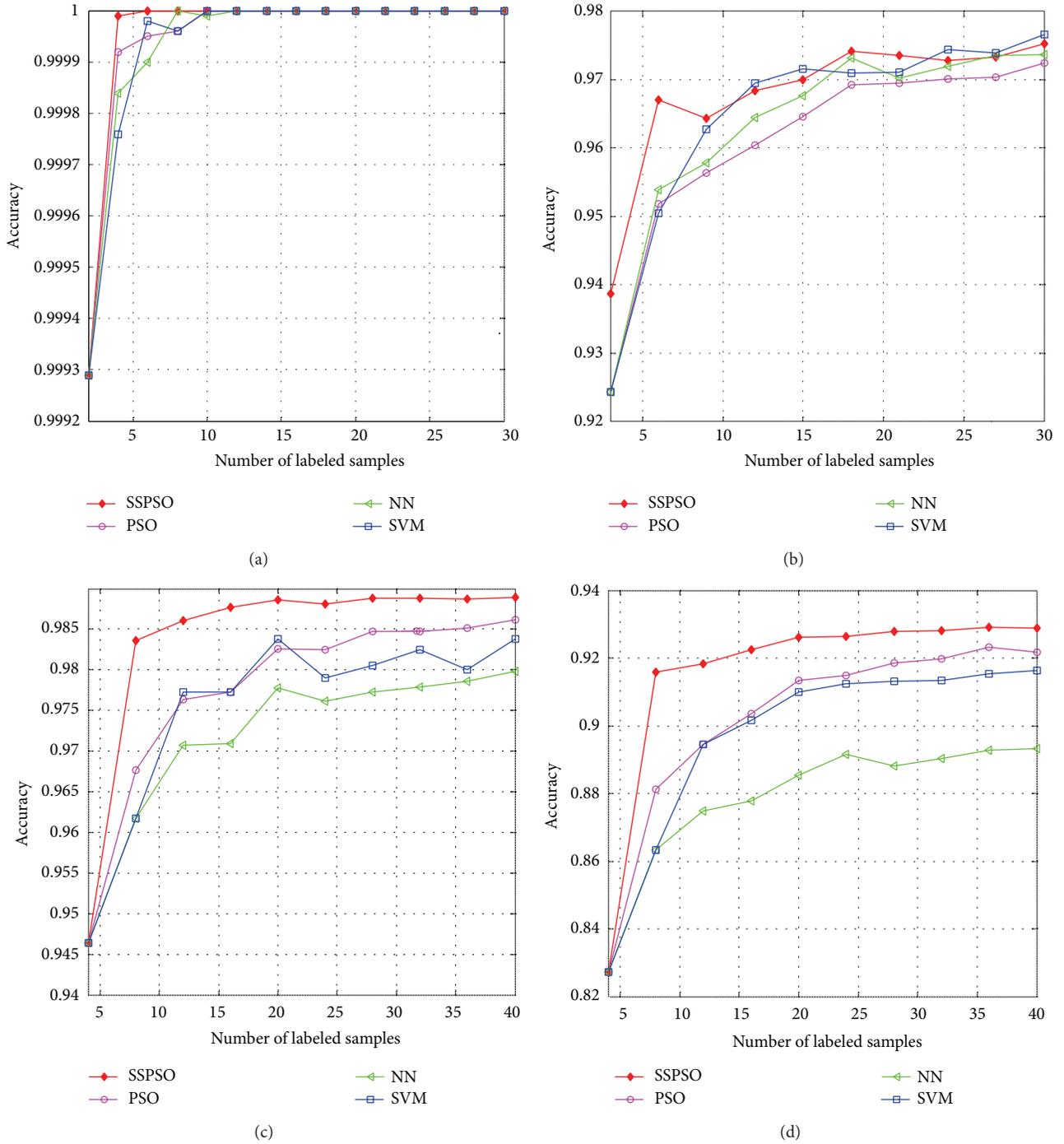


FIGURE 2: Classification results on the artificial dataset with different algorithms: (a) long1, (b) sizes5, (c) square1, and (d) square4.

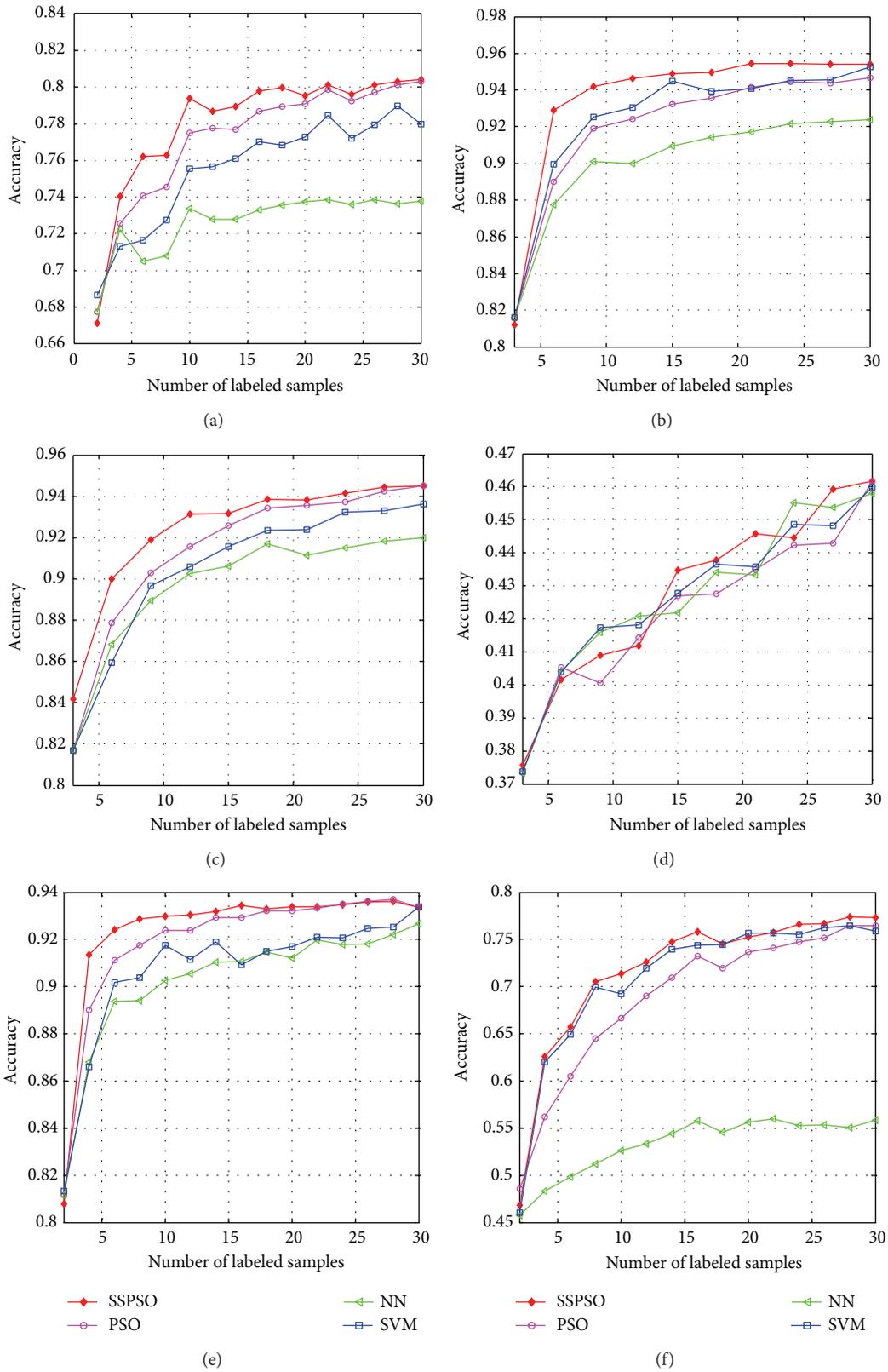


FIGURE 3: Classification results on the UCI datasets with different algorithms: (a) Heart (2 classes); (b) Wine (3 classes); (c) Thyroid (3 classes); (d) Tae (3 classes); (e) SPECT (2 classes); (f) Wdbc (3 classes).

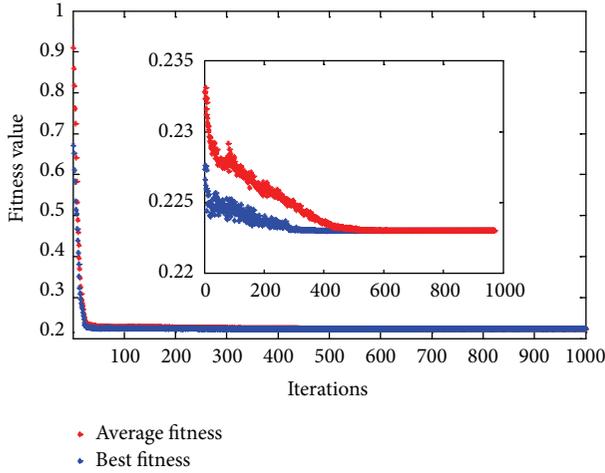


FIGURE 4: Typical behavior of best and average fitness of SSPSO as a function of the number of iterations.

Fitness function plays an important role in PSO. A good fitness function can quickly find the optimization positions of the particles. In [26], the fitness function ψ of the classical PSO classification method is computed as the sum of the Euclidean distances between all the training samples and the class centroids encoded in the particle they belong to. Then, the sum is divided by l , which is the total number of training samples. The fitness of the i th particle is defined as

$$\psi(\mathbf{p}_i) = \frac{1}{l} \cdot \sum_{j=1}^l d(\mathbf{x}_j, \mathbf{p}_{CL(j),i}), \quad (5)$$

where $CL(j)$ denotes the class label of the training sample \mathbf{x}_j , $\mathbf{p}_{CL(j),i}$ denotes the centroid vector of the class $CL(j)$ encoded in the i th particle, and $d(\mathbf{x}_j, \mathbf{p}_{CL(j),i})$ is the Euclidean distance between the training sample \mathbf{x}_j and the class centroid $\mathbf{p}_{CL(j),i}$. Equation (5) only considers the labeled samples which are used to provide the discriminant information. However, in the case that the labeled samples are limited, the labeled samples are too few to represent the real distribution of dataset, while the abundant unlabeled samples are often available and may be helpful to capture the real geometrical structure of the whole dataset. To take full advantage of the existing unlabeled samples, we modify the fitness function by introducing the structure information of unlabeled samples to the original fitness function of PSO classifier. With the assumption of NN method that the neighborhood samples should have the same labels, we propose to use a new fitness function in our proposed SSPSO as follows:

$$\begin{aligned} \psi(\mathbf{p}_i) = & \beta \left(\frac{1}{l} \sum_{j=1}^l d(\mathbf{x}_j, \mathbf{p}_{CL(j),i}) \right) + (1 - \beta) \\ & \times \left(\frac{1}{u} \sum_{k=1}^u \min \{ d(\mathbf{x}_k, \mathbf{p}_{1,i}), \right. \\ & \left. d(\mathbf{x}_k, \mathbf{p}_{2,i}), \dots, d(\mathbf{x}_k, \mathbf{p}_{C,i}) \} \right), \end{aligned} \quad (6)$$

where $\psi(\mathbf{p}_i)$ is the fitness value of the i th particle; β is a weight factor in the range between $[0, 1]$, which controls the ratio of information obtained from the labeled and unlabeled samples; u is the number of the unlabeled samples \mathbf{X}_U ; and l is the number of labeled samples \mathbf{X}_L . The first term on the left side of the fitness function is the discriminate constraint, which means that a good classifier should have a better result on the labeled samples. The second term is the structure constraint, which is helpful to find the real distribution of the whole dataset so as to improve the classification performance. When $\beta = 1$, we obtain the standard PSO algorithm, and when $\beta = 0$, we can obtain an unsupervised PSO clustering method.

The detailed process of the proposed SSPSO is as follows.

Input. The labeled dataset is $\mathbf{X}_L = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l\}$ and the corresponding labels set is $\mathbf{Y}_L = (y_1, y_2, \dots, y_l)^T$; the unlabeled dataset is $\mathbf{X}_U = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_u\}$.

Output. The labels of the unlabeled samples.

Step 1. Load training dataset and unlabeled samples.

Step 2. Normalize the dataset.

Step 3. Initialize the swarm with N particles by randomly generating both the position and velocity vectors for each particle with the entry value between the range $[0, 1]$. It is noticed that the dimension of each particle equals the product of the number of attributes D and the number of classes C .

Step 4. Iterate until the maximum number of iterations is reached.

Substep 1. Calculate the fitness value $\psi(\mathbf{p}_i^{(t)})$ for each particle with (6).

Substep 2. Update the best fitness value $\psi(\mathbf{pbest}_i^{(t)})$ and the best particle of i th particle $\mathbf{pbest}_i^{(t)}$; that is, if $\psi(\mathbf{p}_i^{(t)}) < \psi(\mathbf{pbest}_i^{(t)})$, then $\psi(\mathbf{pbest}_i^{(t)}) = \psi(\mathbf{p}_i^{(t)})$, and $\mathbf{pbest}_i^{(t)} = \mathbf{p}_i^{(t)}$.

Substep 3. If necessary, update the global best particle $\mathbf{gbest}^{(t)}$; that is, $\mathbf{b}^{(t)} = \arg \min_{\mathbf{p}^{(t)}} \{\psi(\mathbf{p}_1^{(t)}), \psi(\mathbf{p}_2^{(t)}), \dots, \psi(\mathbf{p}_N^{(t)})\}$ if $\psi(\mathbf{b}^{(t)}) < \psi(\mathbf{gbest}^{(t)})$, then $\psi(\mathbf{gbest}^{(t)}) = \psi(\mathbf{b}^{(t)})$ and $\mathbf{gbest}^{(t)} = \mathbf{b}^{(t)}$.

Substep 4. Update particles' velocity with (2).

Substep 5. Update particles' position with (3).

Substep 6. Update the inertia factor ω with (4).

Step 5. Use the NN method to obtain the labels of the unlabeled samples \mathbf{X}_U with the optimum centroids represented by \mathbf{gbest} .

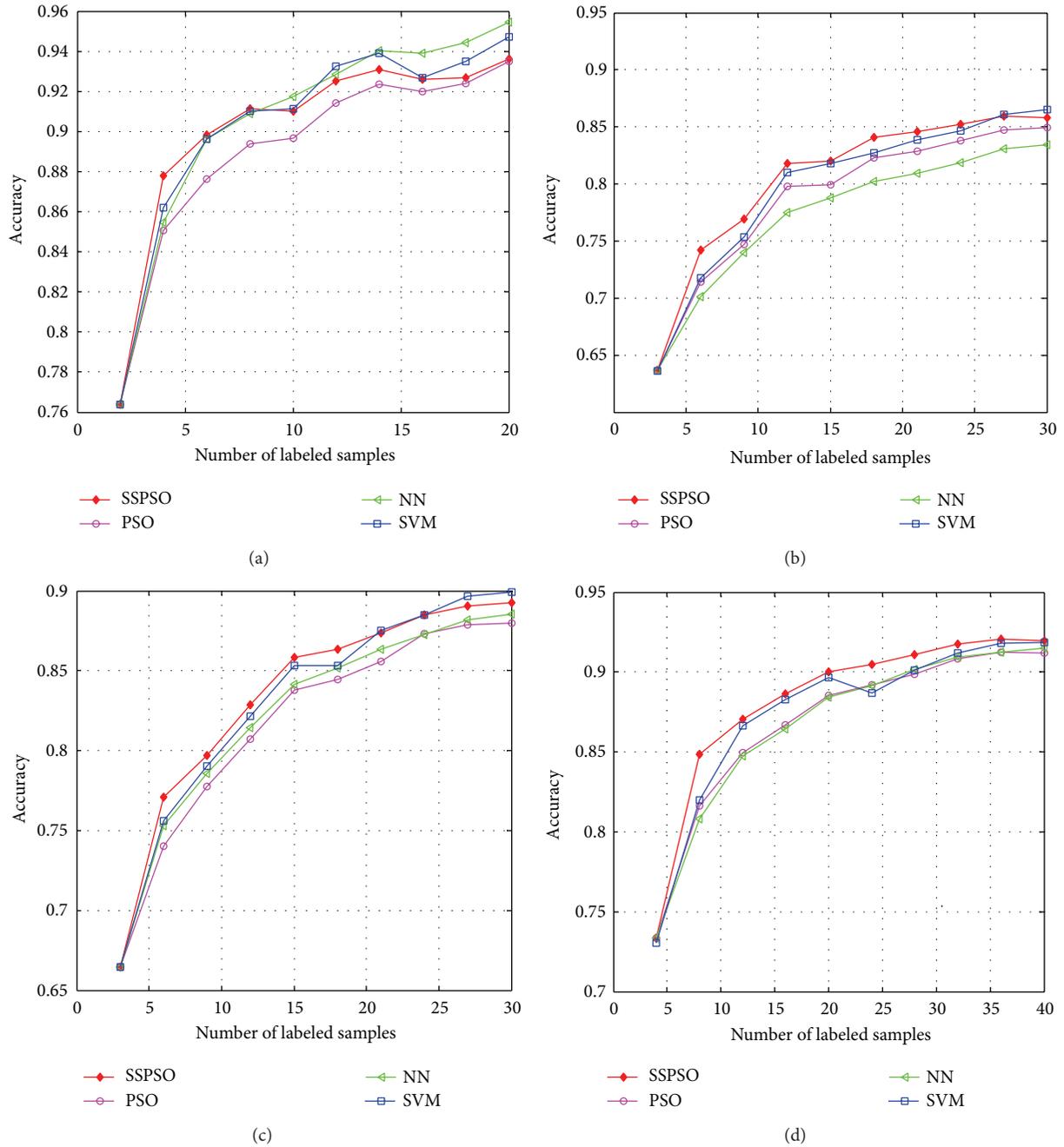


FIGURE 5: Digit recognition on the USPS dataset with different algorithms. (a) Digits “0” and “8,” (b) digits “3,” “5,” and “8,” (c) digits “3,” “8,” and “9,” and (d) digits “1,” “2,” “3,” and “4.”

4. Experimental Results and Analysis

In this section, we assess our proposed method SSPSO on six UCI datasets, four artificial datasets, and the USPS handwritten dataset. The datasets have different attributes and classes, involving different problems including balanced and unbalanced ones.

To evaluate the performance of SSPSO, we make comparisons of the classification results with the PSO-based classifier, the traditional NN classifier, and the classical SVM

classifier. In order to compare the algorithms reasonably, all the parameters of PSO and SSPSO are selected to make them obtain the best results. The parameter settings are as follows. The inertia weight factor ω used in PSO and SSPSO decreases linearly from 0.9 to 0.4. Both c_1 and c_2 are set to 2. The velocity is defined in the range $[-0.05, 0.05]$. The swarm scale N is set to 20 and the maximum number of iterations T_{max} is 1000. The parameters of SVM with Gaussian kernel function are selected by using the gridding search method on the training dataset.

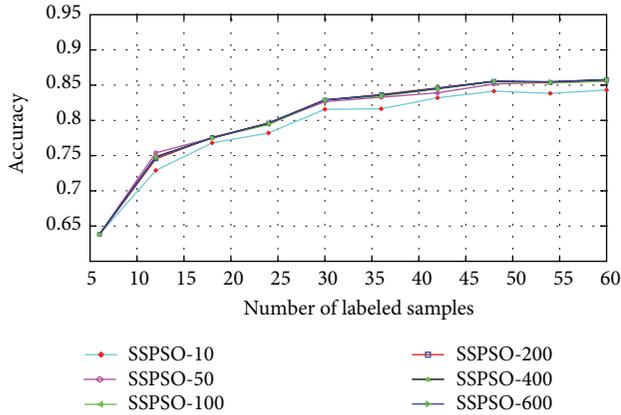


FIGURE 6: {3, 5, and 8} digit recognition on the USPS dataset by SSPSO with different numbers of unlabeled samples.

In addition, we analyze the effect of the number of unlabeled samples on the classification accuracy on USPS dataset. In order to test the robustness of the parameter β in the fitness function to the classification performance, we conduct experiments on UCI datasets with different values of β and analyze the effect of β on the classification performance.

4.1. Artificial Two-Dimension Problems. To test the feasibility of SSPSO for classification, the proposed method is first conducted on four artificial two-dimension datasets, that is, long1, sizes5, square1, and square4. The details are shown in Table 2, and the distributions of the four datasets are shown in Figure 1.

In the experiments, for the first two datasets, we randomly select 1~30 labeled samples per class as the training data, and for the last two datasets we randomly select 5~40 labeled samples per class as the training set, and the rest are used as the test set. Figure 2 plots the curves of accuracy with respect to the number of labeled samples, which shows the average results over 100 runs of the proposed SSPSO comparing with PSO, NN, and SVM on the four datasets. The weight factor β in the fitness function of SSPSO (in (6)) is selected as 0.5. From Figure 2, it can be observed that SSPSO can obtain favorable classification results on the four datasets, which means that SSPSO is feasible for the classification problem. Among the four datasets, long1 is the easiest to classify, on which all the four methods acquire 100% classification accuracy when the number of labeled samples per class exceeds 10. But when the labeled instances are few, for example, only 3 instances per class are labeled, PSO, NN, and SVM cannot classify all the test data correctly, while SSPSO can still obtain 100% classification accuracy. In Figure 2(b), the performance difference among SSPSO, NN, and SVM is not noticeable when the number of labeled samples per class is up to 15, but when the number of labeled instances is small, for example, less than 10, SSPSO can obtain obvious better accuracy than the other methods. It is because SSPSO utilizes the information of unlabeled instances which is helpful to capture the global structure. For square1 and

TABLE 2: Artificial datasets used in experiments.

Data	Class number	Attribute number	Instance number	Normalization
long1	2	2	500/500	Yes
sizes5	3	2	77/154/769	Yes
square1	4	2	250/250/250/250	Yes
square4	4	2	250/250/250/250	Yes

TABLE 3: UCI datasets used in experiments.

Data	Class	Attributes	Instances	Normalization
Heart	2	13	150/120	Yes
Wine	3	13	59/71/48	Yes
Thyroid	3	5	150/35/30	Yes
Tae	3	5	49/50/52	Yes
SPECT	2	22	267	Yes
Wdbc	3	30	357/212	Yes

square4 datasets, the superiority of SSPSO is more apparent; that is, for all the training scenarios, the best performance is achieved by the proposed method.

4.2. UCI Dataset. To further investigate the effectiveness of SSPSO for classification, we also conduct the experiments on six real-life datasets with different numbers of attributes and classes from the UCI machine learning repository [33]. The description of the datasets used in experiments is given in Table 3.

For datasets with 2 classes, we randomly select 1~15 labeled samples per class as the training data, and, for datasets with 3 classes, we randomly select 1~10 labeled samples per class as the training data, and the rest are used as the test set. The results are averaged over 100 runs. The weight factor β in the fitness function of SSPSO (in (6)) is selected as 0.5. Figure 3 shows the classification accuracy with different numbers of training samples on the 6 datasets.

From Figures 3(a), 3(b), 3(c), and 3(e), it can be observed that the proposed SSPSO method outperforms the other three methods on the Heart, Wine, Thyroid, and SPECT datasets, especially when the number of the labeled samples per class is small. It is because that SSPSO uses the information of available unlabeled data which is of benefit to the classification. With the increase of the labeled samples, the superiority becomes weak. From Figure 3(d), it is seen that SSPSO can obtain comparative accuracy with the other three methods. From Figure 3(f), SSPSO is slightly better than SVM, but it is much better than PSO and NN methods. Therefore, it can be concluded that SSPSO works well for some real-life classification tasks especially in the case that the labeled samples are highly limited.

From an evolutionary point of view, in Figure 4 we report the behavior of a typical run of SSPSO in terms of the best individual fitness and average fitness in the population as a function of the number of iterations. It is carried out on the Thyroid database. As can be seen, SSPSO shows two phases. In the first phase with about 50 iterations, the fitness value

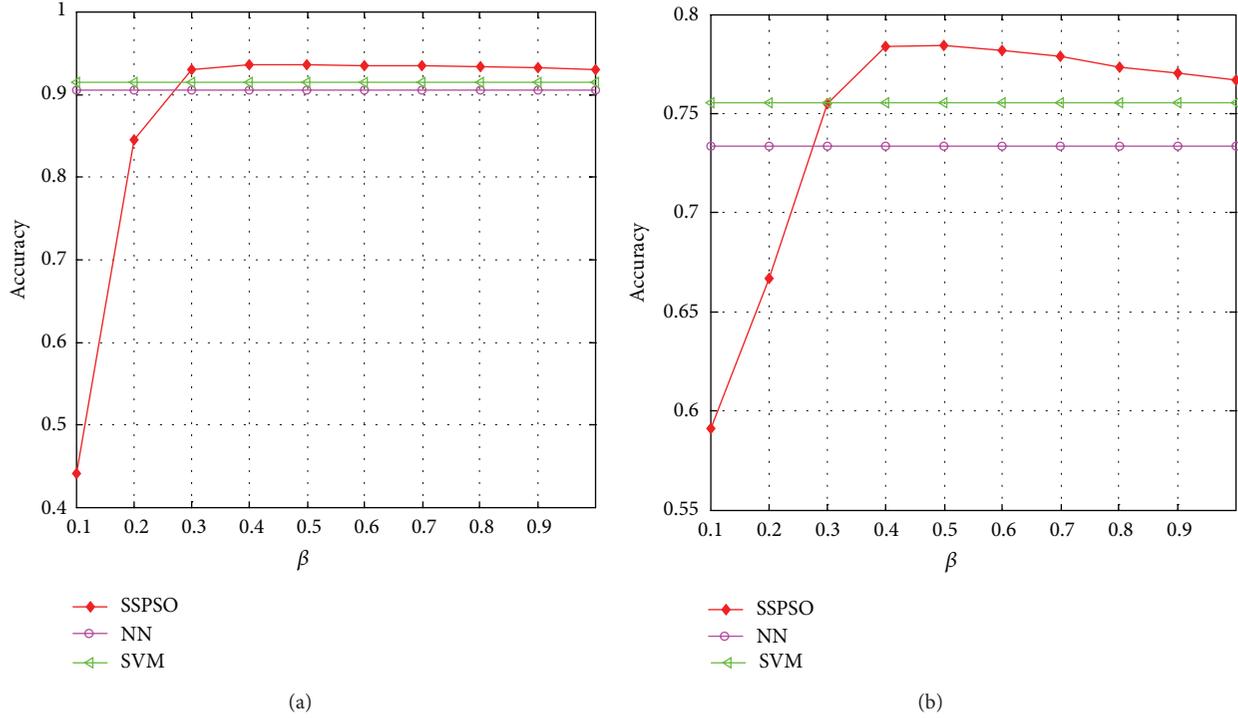


FIGURE 7: Classification accuracy as a function of β in SSPSO on (a) Thyroid and (b) Heart.

decreases sharply, starting from 0.6712 for the best and 0.9192 for the average, and reaching about 0.2305 for the best and 0.2356 for the average. Then, the second phase follows, lasting about 50 iterations, in which both the best and the average fitness values decrease slowly and tend to become closer and closer, until they reach 0.2231 and 0.2247, respectively. And then the average and the best fitness values become more and more similar. Finally, both the two values get to 0.2230.

4.3. USPS Digital Recognition. We also conduct experiments on the USPS handwritten digits dataset to test the performance of SSPSO. This dataset consists of 9298 samples with 10 classes and each sample has the size of 16×16 pixels. Firstly, we apply the principal component analysis on the dataset for feature extraction and select the first 10 principle components as the new features.

We consider four subsets of the dataset in the experiment, that is, the images of digits 0 and 8, with 2261 examples in total, the images of digits 3, 5, and 8, with 2248 examples in total, the images of digits 3, 8, and 9, with a total number of 2429 examples, and the images of digits 1, 2, 3, and 4, with a total number of 3874 examples. We randomly select 1~10 samples per class, respectively, as the training data, and randomly select 200 unlabeled samples to construct the unlabeled sample set \mathbf{X}_U , which is used for semisupervised learning. The weight factor β in the fitness function of SSPSO (in (6)) is selected as 0.7.

The recognition results averaged over 100 independent trials are summarized in Figure 5, where the horizontal axis represents the number of randomly labeled digital images per class in the subset, and the vertical axis represents

the classification accuracy. From Figure 5(a), it is shown that when the number of labeled samples per class is below 14, SSPSO can obtain comparable performance with SVM and KNN and be better than PSO. In particular, SSPSO can outperform the other methods when the labeled samples are few. For the results on the USPS subset of digits 3, 5, and 8 and the subset of digits 3, 8, and 9, shown in Figures 5(b) and 5(c), respectively, one can clearly see that SSPSO method outperforms SVM and is much better than KNN and PSO methods when the number of labeled samples is small. In Figure 5(d), SSPSO still works better than the other methods but the superiority of the proposed SSPSO over the other methods decrease with the increase of labeled samples.

4.4. The Sensitivity Analysis of the Number of Unlabeled Samples. In this section, we validate the effect of the number of the unlabeled samples on the classification accuracy. This experiment is carried on the subset of the USPS dataset with the digit images of 3, 5, and 8. We vary the size of the unlabeled set \mathbf{X}_U to be 10, 50, 100, 200, 400, and 600. Figure 6 illustrates the classification accuracy as the function of the size of the unlabeled set and the number of labeled samples. From Figure 6, one can see that the number of the unlabeled samples affects the accuracy slightly when the number of the labeled samples is small. The plot with 10 unlabeled samples gets much lower accuracy than the other plots, which indicates that the number of unlabeled samples used should not be too small. With the increase of the size of \mathbf{X}_U , SSPSO can obtain better classification accuracy because the proposed method can capture the real structure of the whole dataset more precisely with more unlabeled samples,

but the gaps between the plots of SSPSO with different sizes of X_U become very small. It is noted from the above experiment that, for an unlabeled dataset X_U with a certain scale, when more unlabeled data are added, the classification accuracy of SSPSO may increase a bit, but it will also bring higher computation cost. So in the experiments, 100 to 200 unlabeled samples are proper to use for SSPSO.

4.5. The Sensitivity Analysis of the Parameter β . β in fitness function is an important parameter in our proposed SSPSO, which controls the contributions of information obtained from the labeled and unlabeled samples to the classification. In this section, we will analyze the sensitivity of β in SSPSO. The experiments are conducted on two UCI datasets, that is, Thyroid and Heart. We randomly select 5 samples per class to form the labeled dataset, and the rest are used for test. The mean results over 100 times of randomly selected training datasets with different values of β are shown in Figure 7.

From Figures 7(a) and 7(b), it can be observed that with different values of β , SSPSO is not always better than NN and SVM methods. When β is small, SSPSO may obtain a bad performance; that is, the accuracy is much lower than NN and SVM. When the value of β is small, the effect of the labeled samples is weakened while the effect of the unlabeled samples is strengthened, which is much more like the unsupervised learning. With the increase of β , the accuracy raises sharply. After β gets to 0.4 for the Thyroid dataset and 0.5 for the Heart dataset, the performance keeps stable and even decreases more or less. To balance the effects of the labeled instances and the available unlabeled instances, β is set to 0.5 in our experiments.

5. Conclusions

In this paper, a semisupervised PSO method for classification has been proposed. PSO is used to find the centroids of the classes. In order to take advantage of the amount of unlabeled instances, a semisupervised classification method is proposed based on the assumption that near instances in feature space should have the same labels. Since the discriminative information provided by the labeled samples and the global distribution information provided by the large number of unlabeled samples is used to find a collection of centroids, SSPSO obtains better performance than traditional PSO classification method. In the experiments, four artificial datasets, six real-life datasets from the UCI machine learning repository, and the USPS handwritten dataset are applied to evaluate the effectiveness of the method. The experimental results demonstrated that our proposed SSPSO method has a good performance and can obtain higher accuracy in comparison to the traditional PSO classification method, NN method, and SVM when there are only few labeled samples available.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61272282, 61203303, 61272279, 61377011 and 61373111), the Program for New Century Excellent Talents in University (NCET-13-0948), and the Program for New Scientific and Technological Star of Shaanxi Province (No. 2014KJXX-45).

References

- [1] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, October 1995.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, pp. 1942–1948, December 1995.
- [3] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Congress on Evolutionary Computation (CEC '01)*, pp. 81–86, May 2001.
- [4] A. Paul, A. A. Victoire, and A. E. Jeyakumar, "Particle swarm approach for retiming in VLSI," in *Proceedings of the 46th IEEE Mid-West Symposium on Circuits and Systems*, pp. 1532–1535, 2003.
- [5] X. Zhang, W. Wang, Y. Li, and L. Jiao, "PSO-based automatic relevance determination and feature selection system for hyperspectral image classification," *IET Electronics Letters*, vol. 48, no. 20, pp. 1263–1265, 2012.
- [6] R. J. Lewis, "An introduction to classification and regression tree (CART) analysis," in *Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, Calif, USA*, pp. 1–14, May 2000.
- [7] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Network*, vol. 12, pp. 181–201, 2001.
- [8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [9] L. P. Wang, Ed., *Support Vector Machines: Theory and Application*, Springer, Berlin, Germany, 2005.
- [10] L. P. Wang, B. Liu, and C. R. Wan, "Classification using support vector machines with graded resolution," in *Proceedings of the IEEE International Conference on Granular Computing*, vol. 2, pp. 666–670, July 2005.
- [11] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000.
- [12] J. Trejos-Zelaya and M. Villalobos-Arias, "Partitioning by particle swarm optimization," in *Selected Contributions in Data Analysis and Classification, Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 235–244, 2007.
- [13] M. Sugisaka and X. Fan, "An effective search method for neural network based face detection using particle swarm optimization," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 2, pp. 214–222, 2005.
- [14] Z. Q. Wang, X. Sun, and D. X. Zhang, "A PSO-based classification rule mining algorithm," in *Proceedings of the 3rd International Conference on Intelligent Computing: Advanced*

- Intelligent Computing Theories and Applications*, pp. 377–384, 2007.
- [15] T. Sousa, A. Silva, and A. Neves, “Particle swarm based data mining algorithms for classification tasks,” *Parallel Computing*, vol. 30, no. 5–6, pp. 767–783, 2004.
- [16] Y. J. Zheng, H. F. Ling, J. Y. Xue et al., “Population classification in fire evacuation: a multiobjective particle swarm optimization approach,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 70–81, 2014.
- [17] N. P. Holden and A. A. Freitas, “A hybrid PSO/ACO algorithm for classification,” in *Proceedings of the 9th Annual Genetic and Evolutionary Computation Conference (GECCO '07)*, pp. 2745–2750, July 2007.
- [18] X. Zhu, “Semi-supervised learning literature survey,” Computer Sciences Technical Report 1530, University of Wisconsin-Madison, 2008.
- [19] A. Paul, “Dynamic power management for ubiquitous network devices,” *Advance Science Letters*, vol. 19, no. 7, pp. 2046–2049, 2013.
- [20] J. Wu, “A framework for learning comprehensible theories in XML document classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 1, pp. 1–14, 2012.
- [21] I. Koprinska, J. Poon, J. Clark, and J. Chan, “Learning to classify e-mail,” *Information Sciences*, vol. 177, no. 10, pp. 2167–2187, 2007.
- [22] X. Zhao, X. Li, Y. P. Chao, and S. Wang, “Human action recognition based on semi-supervised discriminant analysis with global constraint,” *Neurocomputing*, vol. 105, pp. 45–50, 2013.
- [23] A. Cervantes, I. M. Galván, and P. Isasi, “AMPSO: a new particle swarm method for nearest neighborhood classification,” *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 39, no. 5, pp. 1082–1091, 2009.
- [24] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [25] F. Aurenhammer and V. Diagrams, “A survey of a fundamental geometric data structure,” *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.
- [26] F. Fernández and P. Isasi, “Evolutionary design of nearest prototype classifiers,” *Journal of Heuristics*, vol. 10, no. 4, pp. 431–454, 2004.
- [27] I. de Falco, A. Della Cioppa, and E. Tarantino, “Evaluation of particle swarm optimization effectiveness in classification,” *Lecture Notes in Computer Science*, vol. 3849, pp. 164–171, 2006.
- [28] K. Y. Huang, “A hybrid particle swarm optimization approach for clustering and classification of datasets,” *Knowledge-Based Systems*, vol. 24, no. 3, pp. 420–426, 2011.
- [29] L. Zhang, L. Wang, and W. Lin, “Semisupervised biased maximum margin analysis for interactive image retrieval,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2294–2308, 2012.
- [30] B. Liu, C. Wan, and L. Wang, “An efficient semi-unsupervised gene selection method via spectral biclustering,” *IEEE Transactions on Nanobioscience*, vol. 5, no. 2, pp. 110–114, 2006.
- [31] X. Zhang, Y. He, N. Zhou, and Y. Zheng, “Semisupervised dimensionality reduction of hyperspectral images via local scaling cut criterion,” *IEEE Geoscience Remote Sensing Letters*, vol. 10, no. 6, pp. 1547–1551, 2013.
- [32] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, 1998.
- [33] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, “UCI Repository of machine learning databases,” Department of Information and Computer Science, University of California, Irvine, Calif, USA, 1998, <http://www.ics.uci.edu/~mllearn/MLRepository.html>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

