Neurocomputing 275 (2018) 180–191

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Self-training semi-supervised classification based on density peaks of data

Di Wu[a,b,1], Mingsheng Shang[a,*], Xin Luo[a,1], Ji Xu[a,c], Huyong Yan[a], Weihui Deng[a], Guoyin Wang[a,*]

[a] Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China
[c] School of Information Science and Technology, Southwest Jiaotong University 610031, Chengdu, China

ARTICLE INFO

Article history:
Received 29 April 2016
Revised 24 May 2017
Accepted 27 May 2017
Available online 31 May 2017

Communicated by Prof. Zidong Wang

Keywords:
Density peaks
Self-training
Semi-supervised classification
Supervised learning

ABSTRACT

Having a multitude of unlabeled data and few labeled ones is a common problem in many practical applications. A successful methodology to tackle this problem is self-training semi-supervised classification. In this paper, we introduce a method to discover the structure of data space based on find of density peaks. Then, a framework for self-training semi-supervised classification, in which the structure of data space is integrated into the self-training iterative process to help train a better classifier, is proposed. A series of experiments on both artificial and real datasets are run to evaluate the performance of our proposed framework. Experimental results clearly demonstrate that our proposed framework has better performance than some previous works in general on both artificial and real datasets, especially when the distribution of data is non-spherical. Besides, we also find that the support vector machine is particularly suitable for our proposed framework to play the role of base classifier.

1. Introduction

Supervised learning (classification) is an active research problem in data mining and machine learning. So far, it has been widely used in power system protection, biological medicine, face recognition, image processing, and object detection, etc [1–6]. Supervised learning relies on the samples with class labels to train a good classifier, through which class labels can be provided for new samples. However, due to extensive expert effort along with time consumption of data labeling, it is hard to obtain sufficient labeled data. On the contrary, unlabeled data are often abundant in the real world. Consequently, having a multitude of unlabeled data and few labeled ones occurs quite often in many practical applications. In this scenario, traditional supervised learning often fails to learn an appropriate classifier with labeled data only [7]. Nevertheless, semi-supervised classification (SSC) is a learning paradigm concerned with finding a way to improve supervised learning by using unlabeled data [8–10]. Hence, in this type of learning, it is not necessary to label all the collected data for training the classifier.

Various approaches of SSC have been proposed and studied all over the world. They are usually classified depending on the different assumptions related to the link between the distribution of unlabeled and labeled data. General models are based on manifold and /or cluster assumption. If data correspond approximately to a manifold of lower dimensionality than the input space, it is suitable for manifold assumption [11]. The most common manifold assumption based models are the graph min-cut problems. The graph construction determines the behavior of the models because two instances connected by a strong edge likely have same labels [12]. The cluster assumption supposes that similar examples should have the same labels. In this case, generative models [13] or support vector machines based models [14] are proposed to achieve SSC. Recently, multiple assumptions in one model have also been addressed by some researchers [15–17].

A successful methodology to tackle SSC problems is self-labeled techniques, which take advantage of a supervised classifier to label instances with unknown class and do not make any specific suppositions about input data [10]. Self-labeled techniques include two well-known methodologies: co-training and self-training. The standard co-training [18] considers the feature space to be two different conditionally independent views. Each view is able to train one classifier and then teach each other to predict the classes perfectly [19,20]. In addition, advanced approaches for co-training are multi-

* Corresponding authors.
E-mail addresses: wudi@cigit.ac.cn, msshang@cigit.ac.cn, wanggy@ieee.org (G. Wang).
[1] Di Wu and Xin Luo contributed equally to this work.

http://dx.doi.org/10.1016/j.neucom.2017.05.072
0925-2312/

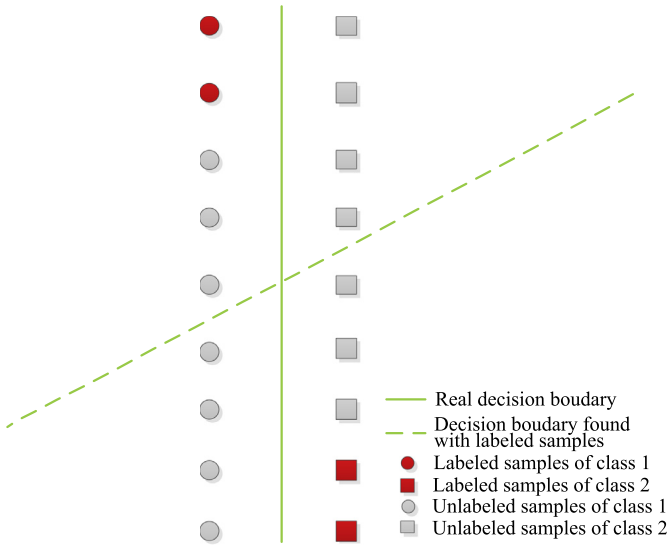

Fig. 1. Illustration of misclassification of using semi-supervised FCM algorithm to improve self-training SSC.

view learning, which does not require explicit feature splits or the iterative mutual-teaching procedure [21–23]. Self-training, as the name implies, attempts to iteratively enlarge the labeled training set [24]. To begin with, a classifier is trained with initial labeled data. Then, unlabeled data, which are selected with the highest confidence, are added incrementally into the labeled training set with their predicted labels. The procedure is repeated until convergence. Self-training has been successfully applied to many real applications, such as word sense disambiguation [25] and subjective nouns [26].

However, self-training method is limited by the number of labeled data and their distribution. When the labeled data cannot roughly represent the underlying structure of the entire data space, the training process will fail to approach the real data space and obtain a bad classifier. Adankon and Cheriet developed an improved version of self-training, called help-training, to train the main discriminative classifier by using a generative model [27]. But, the problems existed in self-training have also not been solved fundamentally because the generative model was trained only by labeled data. Thus, Gan et al. proposed that using semi-supervised fuzzy c-means (FCM) algorithm to improve self-training, where unlabeled data and labeled data were exploited to reveal the actual data space structure through clustering analysis [28]. Nevertheless, Gan's algorithm is not appropriate for the non-spherical distribution of data, which occurs quite often in the real world. Fig. 1 shows an example that Gan's algorithm may not find the real decision boundary. In this case, since the semi-supervised FCM algorithm cannot discover the real data space structure of non-spherical distribution of data, unlabeled samples distributed nearby initial labeled samples have been chosen to reduce the performance of classifier obtained by training.

Recently, Rodriguez and Laio achieved a density peaks clustering algorithm, published on the famous journal of Science, to detect non-spherical clusters and to automatically find the correct number of clusters [29]. For each data point x_i, this clustering algorithm computed two quantities: its local density and its distance from points of higher local density. By researching the two quantities, we found that the real structure of entire data space, no matter spherical or non-spherical distribution of data, can be discovered by making each data point x_i points to its nearest point with higher local density. This result inspired us to further research on SSC. Thus, in this paper, we propose a framework for self-training SSC based on find of density peaks. Our proposed framework consists of two main parts: one part is discovering the real structure of entire data space by searching and finding density peaks of data; the other part is integrating the real structure of entire data space into the self-training process to iteratively train a classifier. Our proposed framework has three advantages: (a) it is not limited by the distribution of initial labeled data and entire data space; (b) it is a constructive model without prior conditions; (c) it is suitable for any supervised algorithm to improve its performance by using the abundant unlabeled data. The main contributions of this work include:

(a) The proposed framework for self-training SSC, which is able to improve the performance of a supervised algorithm by using the abundant unlabeled data.
(b) Detailed algorithm design and analysis for the proposed framework.
(c) Detailed empirical studies conducted on both artificial and real datasets, along with analyses regarding the experimental results.

To the authors' best knowledge, such efforts have never been seen in any prior work.

The rest of the paper is organized as follows: in Section 2, we introduce the method to discover structure of data space based on find of density peaks; in Section 3, we describe our framework and algorithm; in Section 4, we discuss the experimental results on both artificial and real datasets, and in Section 5, we conclude this paper and make some plans for the future.

2. Discovering the structure of data space based on find of density peaks

Clustering is an important unsupervised method for analyzing unlabeled data [30] and can find the underlying structure of data space [28]. As mentioned in [29], a density peaks clustering algorithm was demonstrated based on the idea that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. It has the basis that the similarity between data points is evaluated by the distance measurement. In other word, if two data points have the closer distance, they are more similar. For each data point x_i, the local density ρ_i is defined as:

$$\rho_i = \sum_j \chi(d_{ij} - d_c),\ \chi(x) = \begin{cases} 1, & x < 0 \\ 0, & others \end{cases} \tag{1}$$

where d_{ij} is distances between data points x_i and x_j, and d_c is a cutoff distance without a fixed value. Obviously, ρ_i indicates the number of points that are closer than d_c to point x_i. δ_i is the minimum distance between x_i and any other point with higher local density than ρ_i:

$$\delta_i = \begin{cases} \min_{j:\rho_i<\rho_j}(d_{ij}), & others \\ \max_j(d_{ij}), & \forall j,\ \rho_i \geq \rho_j \end{cases} \tag{2}$$

In the process of calculating δ_i, each data point x_i has a corresponding data point x_j, which is x_i's nearest point with higher local density. Thus, for each data point x_i, it is characterized by three quantities: ρ_i, δ_i, and x_j. By making each data point x_i points to its corresponding data point x_j, the real structure of entire data space, no matter spherical or non-spherical distribution of data, can be discovered. This observation, which is the core of our framework, is illustrated by the simple example in Fig. 2. Firstly, 30 artificial data points (x_1, x_2, ..., x_{30}) with two classes are randomly generated as the follow conditional distributions:

$$\begin{cases} Class\ 1 : x_i(a,b), [(a,b)|a = N(10, 0.3), b = N(10, 3)], i = 1, 2, \ldots, 15. \\ Class\ 2 : x_i(a,b), [(a,b)|a = N(14, 0.3), b = N(10, 3)], i = 16, 17, \ldots, 30. \end{cases} \tag{3}$$

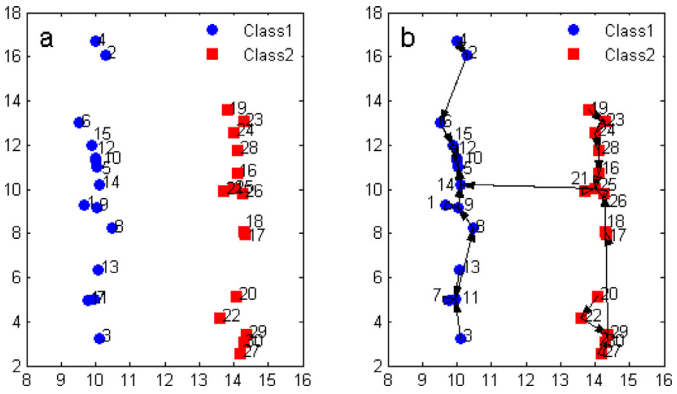

Fig. 2. Illustration of discovering data space: (a) distribution of data points, (b) real structure of entire data space.

where $N(\mu, \sigma^2)$ is normal distribution with mean μ and variance σ^2. Fig. 2(a) shows the distribution of 30 artificial data points generated as formula (3). Next, according to formulas (1) and (2), we compute the three quantities (ρ_i, δ_i, and x_j) for 30 artificial data points, respectively. Finally, making each data point x_i points to its corresponding data point x_j. Consequently, the real structure of entire data space is discovered, as shown in Fig. 2(b). Note that each data point x_i points to its unique next data point x_j in one-way successively until the data point with highest local density. For example, the sequenced relationships of data points x_{28}, x_{16}, x_{25}, and x_{14} are: x_{28} points to x_{16}, x_{16} points to x_{25}, and x_{25} points to x_{14}.The data point x_{14} has the highest local density. In the Section 4.5, we will discuss the selection rules of d_c.

As discussed above, the real structure of entire data space, no matter spherical or non-spherical distribution of data, can be well discovered by computing the two quantities of ρ_i, δ_i, for each data point x_i. In addition, the discovering of the real structure of entire data space is extremely fast because it has no iterative process. All the advantages are very suitable for the self-training SSC. Thus, we choose the density peaks clustering algorithm to discover the real structure of entire data space and achieve our proposed framework.

3. Our framework and algorithm

In this section, our proposed framework for self-training SSC is described, in which the structure of data space discovered by finding the density peaks of data is integrated into the self-training process to iteratively train a classifier. The formal definition of the SSC problem is described as follows: $x_i = (x_i^1, x_i^2, \ldots, x_i^d, \omega)$, which is a sample that belongs to a class ω and a d-dimensional space.x_i^d is the value of the dth feature of the ith sample. L is labeled set with ω known and U is unlabeled set with ω unknown. The $L \cup U$ set forms the training set T_R. In particular, the number of initial unlabeled data is much larger than that of initial labeled data for a typical SSC problem. The aim of SSC is to learn a better classifier C by using T_R instead of L.

Fig. 3 depicts the flowchart of our proposed framework. Step 1, find of density peaks of data is employed to learn the underlying structure of entire data space on the T_R. Next, the structure of entire data space is integrated into the self-training process to iteratively train a classifier, which consists of two similar steps. Step 2: (a) a classifier C is trained based on the L; (b) the unlabeled data, which are the "next" points of labeled data according to the structure of entire data space, are then classified by the classifier C; (c) these unlabeled data with their predicted labels are added into L and are subtracted from U. At the same time, updating L and U; (d) repeating the operations a to c until all the "next" unlabeled data points of labeled data are added into L and are subtracted from U. Step 3: This step has the similar 4 operations, the only difference between step 2 and step 3 is the operation b, which means that the "previous" unlabeled data points of labeled data, not the "next" unlabeled data points of labeled data, are selected to be classified by the classifier C. Here the notions of "next" and "previous" are only used based on the structure of data space discovered by step 1. For example in Fig. 2(b), if the data point x_{16} is the labeled data and the other data points are unlabeled data, then the "next" unlabeled data point of x_{16} is x_{25}, and the "previous" unlabeled data point of x_{16} is x_{28}. After that, a better classifier C is obtained.

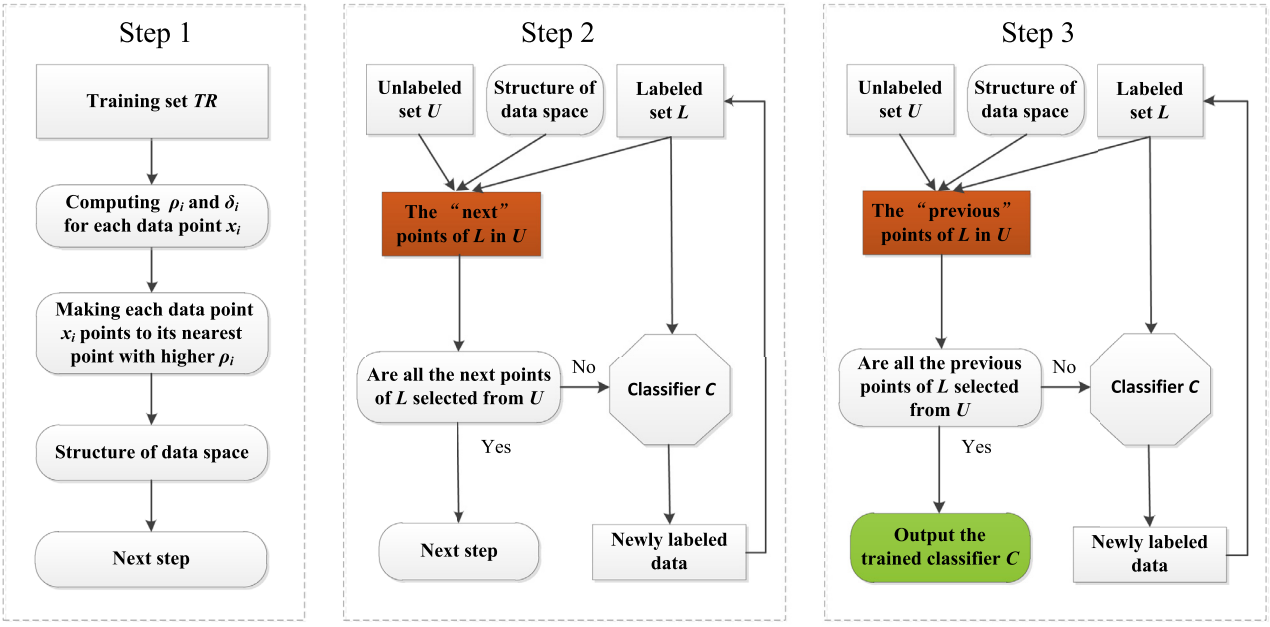

Fig. 3. Flowchart of our proposed framework.

Note that any supervised algorithm, like support vector machine (SVM) [31], k-nearest neighbor (KNN) [32], and classification and regression tree (CART) [33], can be used as the classifier C in our proposed framework. The specific algorithm pseudo-code of our proposed framework is outlined in Algorithm 1. Note that our proposed framework is different from [28]. Gan et al. [28] used semi-supervised FCM algorithm to learn the underlying structure of data space, which is not appropriate for the non-spherical distribution of data. Nevertheless, our proposed framework exploits the find of density peaks to reveal the actual structure of entire data space, no matter spherical or non-spherical distribution of data. Thus, the classifier trained by our proposed framework has better performance than that trained by the [28]. We will discuss this in Section 4.3.

In order to evaluate the performance of Algorithm 1, a test set T_S composed of t number of instances x_i with ω known is used for the test stage. Three quantities of the accuracy rate (AR), the mean accuracy rate (MAR), and the standard deviation of AR (SD- AR), are respectively computed as:

$$AR = \frac{1}{t}\sum_{i=1}^{t}\psi(\omega, f(x_i)),\ \psi(\omega, f(x_i)) = \begin{cases} 1, if\ \omega = f(x_i) \\ 0, else \end{cases} \tag{4}$$

$$MAR = \frac{1}{n}\sum_{k=1}^{n} AR_k \tag{5}$$

$$SD - AR = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(AR_k - MAR)^2} \tag{6}$$

where $f(x_i)$ is the calculate label of x_i, and n is the repeated times of computing AR. MAR represents the classification ability of Algorithm 1, SD-AR represents the robustness of Algorithm 1.

4. Experimental results and discussions

In the experiments, two artificial datasets and two real datasets are used to test the performance of our algorithm. The properties of these datasets are shown in Table 1.

4.1. Artificial datasets

Two artificial datasets, Gauss50 and Gauss50x, are generated like the ways used in [27,28]. They are a two-class problem in a 50-dimensional input space. For Gauss50, each class is generated with equal probability from a Gaussian distribution with a unit covariance matrix. The means of the Gaussian are (0.23, 0.23, ..., 0.23) for class 1 and (−0.23, −0.23, ..., −0.23) for class 2. For Gauss50x, each class is generated with equal probability from a Gaussian mixture distribution. The data in each class have the conditional distributions as follow:

$$\begin{cases} p(x|y=1) = 0.49N(\mu_1, I) + 0.51N(\mu_2, I) \\ p(x|y=-1) = 0.49N(-\mu_1, I) + 0.51N(-\mu_2, I) \end{cases} \tag{7}$$

where $\mu_1 = (0.25, 0.25, ..., 0.25)$, $\mu_2 = (0.25, 0.25, ..., 0.25, -0.25, -0.25, ..., -0.25)$, $N(\mu, I)$ is a Gaussian distribution with mean μ and unit covariance matrix.

Table 1
Summarizes the properties of all the experimental datasets.

Name	#Examples	#Attributes(d)	#Classes
Gauss50	2000	50	2
Gauss50x	2000	50	2
Banknote authentication	1372	4	2
Waveform	5000	21	3

4.2. Real datasets

Two real datasets from the UCI dataset [34], Banknote Authentication and Waveform, are tested in the experiments. Banknote Authentication consists of 1372 samples in a 4-dimensional input space with 2 classes. Waveform is composed of 5000 samples in a 21-dimensional input space with 3 classes.

4.3. Comparisons between our algorithm and some previous works

In order to compare our algorithm with some previous works, two representative self-labeled SSC algorithms, i.e., semi-supervised FCM [28] and semi-supervised Tri-training [39], are chosen to run the experiments. Semi-supervised FCM and semi-supervised Tri-training respectively represent the methodologies of self-training and co-training of self-labeled SSC. Note that all the three self-labeled SSC algorithms (two chosen and our algorithm) are suitable for any supervised algorithm, thus, the supervised algorithms of SVM, KNN, and CART are respectively used as the base classifier for them to test their performances. Table 2 summarizes all the parameters used in these algorithms.

In the experimental phase, we use the 10-fold cross-validation strategy to determine the final experimental results. Firstly, each dataset is split into ten folds, and each one contains 10% of the instances of each dataset. Then, 9 folds are selected to use as the

Algorithm 1.

Input: L = a labeled dataset, U = an unlabeled dataset
Output: A classifier C
Method:
1. Calculate ρ_i for each sample x_i of L and U according to Formula (1)
2. Calculate δ_i for each sample x_i of L and U according to Formula (2)
3. Establish the structure of data space by making each sample x_i points to its unique nearest sample with higher ρ_i according the results of step 2
4. Train the classifier C with L
5. Repeat until all the "next" points of samples of L are selected from U
   -Select a dataset T from U, where each sample x_j is the "next" points of samples of L according to the structure of data space
   -Label the samples of T with the trained classifier C
   -Update the current labeled dataset $L \leftarrow L \cup T$
   -Update the current unlabeled dataset $U \leftarrow U\text{-}T$
  -Retrain the classifier C with L
6. Repeat until all the samples are selected from U
   -Select a dataset T from U, where each sample x_j is the "previous" points of samples of L according to the structure of data space
   -Label the samples of T with the trained classifier C
   -Update the current labeled dataset $L \leftarrow L \cup T$
   -Update the current unlabeled dataset $U \leftarrow U\text{-}T$
  -Retrain the classifier C with L
7. Return the classifier C

Table 2
Descriptions of all the parameters used in the experiments.

Mark	Algorithm	Parameters
/	SVM	*LIBSVM: all the parameters are set as default values* [31].
/	KNN	*Number of neighbors* = 3.
/	CART	*MATLAB2014: all the parameters are set as default values.*
A1	Semi-supervised Tri-training	
A2	Semi-supervised FCM	*Threshold* $\varepsilon_1 = 1/$ *(the number of classes).*
Our-A	Our algorithm	$P_a = 2$, *please refer to Section 4.5, formula (8)*

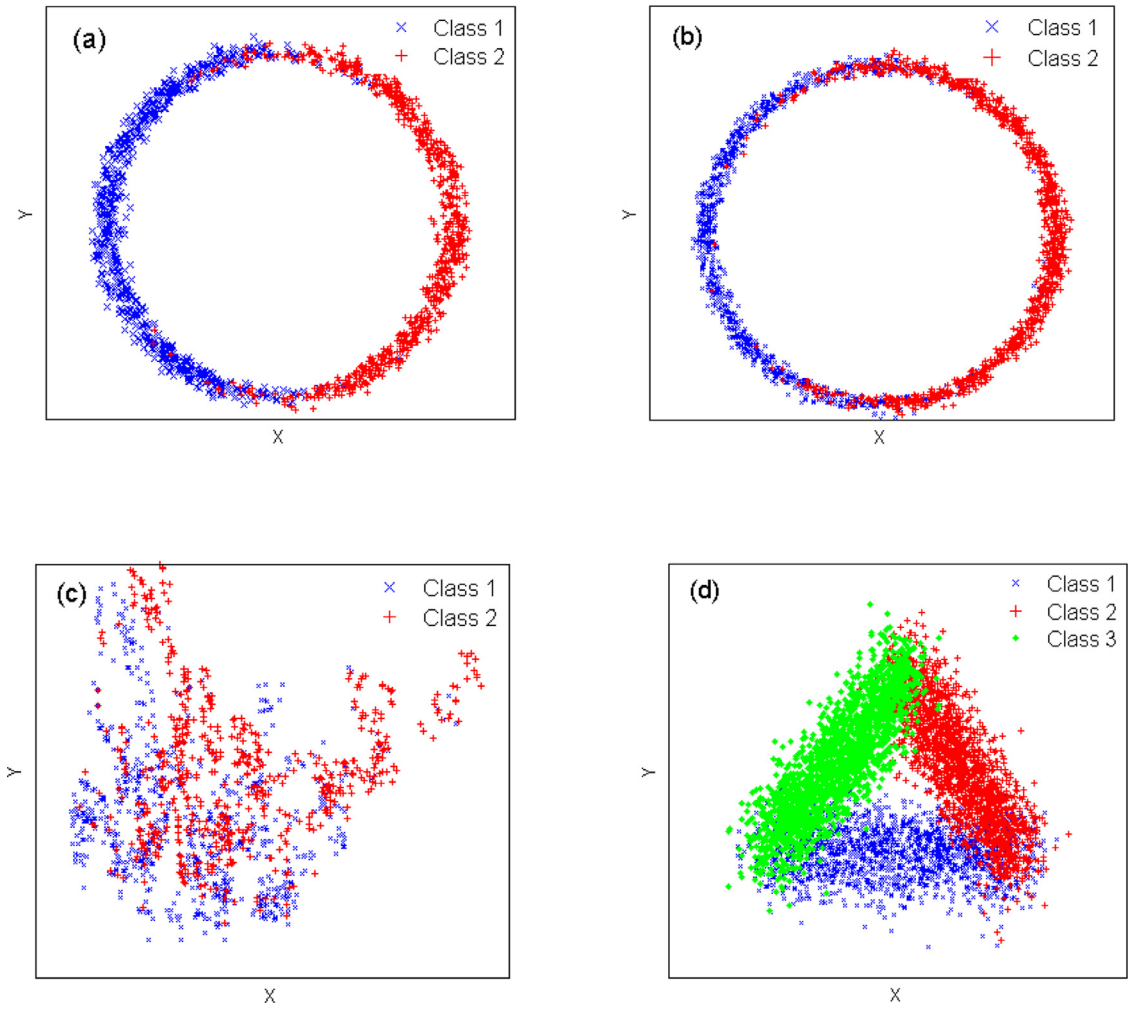

Fig. 4. The point distributions of datasets of: (a) Gauss50, (b) Gauss50x, (c) Banknote authentication, and (d) Waveform.

training set T_R and the remaining one forms the testing set T_S. After that, the T_R is then divided into labeled part L and unlabeled part U by using a random stratified selection, which means that the selected number of instances for each class is proportional to the number of them in the T_R. In addition, we will ensure that at least one representative instance of each class is selected in the L. Thus, each dataset is divided into three parts: L, U, and T_S (L and U form T_R). To make sure each fold can serve as the T_S once, the above steps will be executed ten times. Specifically, an initial ratio 10% of labeled data is adopted over the whole datasets. The comparison results are shown in Tables 3–6.

From Tables 3–6, we observe that all the three self-labeled SSC algorithms give better results than supervised algorithm in general, which means that unlabeled data can improve the generalization capacity of the supervised algorithm. In addition, we also find that our algorithm has the best performances on the three datasets of Gauss50, Gauss50x, and banknote authentication in general. In order to further analyze the experimental results of Tables 3–6, we perform the Friedman test [40] with the significance level $\alpha = 0.05$ to conduct the statistical analysis and the accepted hypotheses are highlighted in bold. The results are recorded in Table 7, where the higher average value of rankings indicates the better performance and the maximum values of each column are highlighted in bold.

As shown in Table 7, when the supervised algorithms of SVM and KNN are used as the base classifier, the statistical results accept the hypothesis that our algorithm has the best performances on datasets of Gauss50, Gauss50x, and banknote authentication except for the case of on Gauss50x with supervised algorithm KNN. However, the p-value of 0.190 also reflects that our algorithm has slightly better performances than the others on the exceptional case. When the supervised algorithm CART is used as the base classifier, although our algorithm is not the best algorithm, it also has slightly better performances than any supervised algorithm on datasets of Gauss50, Gauss50x, and Banknote Authentication.

Table 3
Experimental results of comparisons with 10-fold cross-validation: accuracy rate (%) on Gauss50.

10-fold cross- validation	SVM				KNN				CART			
	SVM	A1 with SVM	A2 with SVM	Our-A with SVM	KNN	A1 with KNN	A2 with KNN	Our-A with KNN	CART	A1 with CART	A2 with CART	Our-A with CART
#1	92.50	91.00	93.50	93.50	81.50	82.50	82.00	85.50	70.50	71.00	69.00	72.00
#2	94.00	93.00	94.00	94.50	83.00	85.50	82.00	88.00	67.50	70.50	70.00	69.00
#3	92.00	92.00	94.00	93.50	81.00	80.50	80.50	82.50	67.50	63.50	71.50	67.00
#4	93.00	94.00	93.50	94.50	81.50	81.00	82.50	84.00	64.50	65.50	61.00	67.00
#5	94.50	95.00	94.50	95.00	83.50	84.50	85.00	90.50	68.00	67.00	69.00	72.00
#6	89.00	93.00	92.50	92.50	81.00	81.50	86.00	85.50	63.00	61.00	64.50	64.50
#7	95.50	96.00	97.00	96.50	87.50	87.50	85.00	88.50	67.50	66.00	65.00	70.00
#8	94.00	96.50	96.00	96.00	86.00	86.00	84.00	79.00	62.50	62.00	60.50	56.00
#9	92.00	93.50	92.50	94.50	83.50	83.50	82.50	85.00	60.50	60.50	62.00	65.00
#10	92.00	93.00	92.50	94.50	87.50	87.50	85.50	88.50	70.50	74.00	69.50	67.50
MAR	92.85	93.70	94.00	94.50	83.60	84.00	83.50	85.70	66.20	66.10	66.20	67.00
SD-AR	1.81	1.72	1.51	1.18	2.56	2.60	1.84	3.38	3.41	4.57	4.09	4.65

Table 4
Experimental results of comparisons with 10-fold cross-validation: accuracy rate (%) on Gauss50x.

10-fold cross- validation	SVM				KNN				CART			
	SVM	A1 with SVM	A2 with SVM	Our-A with SVM	KNN	A1 with KNN	A2 with KNN	Our-A with KNN	CART	A1 with CART	A2 with CART	Our-A with CART
#1	92.00	92.00	93.00	93.00	88.00	88.00	85.00	85.00	66.50	69.50	69.00	70.00
#2	92.50	93.50	93.50	95.00	84.50	84.50	80.50	87.00	74.00	70.00	73.00	70.50
#3	95.50	96.00	98.50	99.00	86.50	86.00	87.50	87.50	71.00	69.50	71.50	76.50
#4	94.50	96.50	96.50	95.50	90.50	91.50	90.50	86.00	67.00	76.00	68.00	62.00
#5	88.50	90.00	93.50	93.50	85.00	84.00	85.00	86.50	63.00	69.00	62.50	62.50
#6	95.50	94.00	95.00	96.00	86.50	86.50	87.50	87.00	67.00	74.00	64.50	70.50
#7	92.50	93.50	94.50	95.00	87.00	87.50	89.00	94.00	73.00	73.50	75.00	71.50
#8	96.00	95.50	96.50	94.00	84.00	83.50	90.50	90.00	69.00	72.00	67.50	71.00
#9	94.50	94.50	93.50	95.00	87.50	88.00	85.50	89.00	70.00	71.50	69.00	71.00
#10	94.00	94.50	95.00	95.00	84.00	86.00	90.00	88.00	69.50	73.50	68.50	71.50
MAR	93.55	94.00	94.95	95.10	86.35	86.55	87.10	88.00	69.00	71.85	68.85	69.70
SD-AR	2.25	1.93	1.76	1.65	2.06	2.36	3.17	2.55	3.27	2.36	3.71	4.33

Table 5
Experimental results of comparisons with 10-fold cross-validation: accuracy rate (%) on banknote authentication.

10-fold cross- validation	SVM				KNN				CART			
	SVM	A1 with SVM	A2 with SVM	Our-A with SVM	KNN	A1 with KNN	A2 with KNN	Our-A with KNN	CART	A1 with CART	A2 with CART	Our-A with CART
#1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	87.59	89.78	87.59	85.40
#2	98.54	98.54	98.54	99.27	99.27	100.00	99.27	100.00	91.24	93.43	90.51	91.24
#3	98.54	99.27	97.81	99.27	98.54	98.54	98.54	98.54	85.40	85.40	85.40	86.13
#4	97.08	99.27	97.08	98.54	99.27	99.27	99.27	99.27	89.05	92.70	89.05	89.78
#5	99.27	98.54	97.08	100.00	98.54	97.08	98.54	100.00	93.43	95.62	93.43	93.43
#6	97.81	97.81	99.27	99.27	97.08	97.08	96.35	98.54	83.94	86.86	83.21	89.78
#7	97.81	98.54	98.54	98.54	99.27	99.27	99.27	99.27	95.62	97.81	95.62	96.35
#8	94.16	94.16	94.16	94.89	98.54	100.00	98.54	100.00	97.08	93.43	96.35	94.16
#9	95.62	96.35	98.54	98.54	99.27	100.00	99.27	99.27	91.24	94.16	94.16	90.51
#10	99.28	97.12	100.00	100.00	97.84	97.84	97.84	98.56	92.81	94.96	91.37	94.24
MAR	97.81	97.96	98.10	98.83	98.76	98.91	98.69	99.35	90.74	92.42	90.67	91.10
SD-AR	1.79	1.71	1.73	1.51	0.84	1.20	1.02	0.64	4.26	3.92	4.38	3.55

Table 6
Experimental results of comparisons with the 10-fold cross-validation: accuracy rate (%) on Waveform.

10-fold cross- validation	SVM				KNN				CART			
	SVM	A1 with SVM	A2 with SVM	Our-A with SVM	KNN	A1 with KNN	A2 with KNN	Our-A with KNN	CART	A1 with CART	A2 with CART	Our-A with CART
#1	85.60	85.40	85.80	87.00	81.53	81.53	80.12	81.33	69.80	70.80	69.20	69.60
#2	84.00	84.00	85.00	83.40	78.64	78.64	78.24	79.44	70.20	69.20	70.60	71.00
#3	81.80	82.40	82.60	83.00	82.44	82.63	80.44	78.44	70.00	69.40	68.80	69.20
#4	81.80	82.20	81.60	80.60	75.90	75.90	77.51	76.51	66.60	70.40	66.80	67.20
#5	85.00	85.00	84.60	85.80	78.84	78.64	78.84	76.45	63.80	65.80	63.60	69.00
#6	83.80	83.80	85.40	85.20	79.24	79.24	81.24	74.65	74.20	73.80	73.80	73.00
#7	86.40	86.20	86.00	85.40	79.72	79.72	83.94	79.12	69.60	69.40	70.20	68.60
#8	88.60	88.80	90.20	88.20	81.64	81.64	81.04	79.24	65.20	70.80	66.00	65.60
#9	84.20	84.20	85.00	85.00	82.04	82.04	79.24	78.44	68.80	71.40	69.40	63.80
#10	85.60	85.60	85.60	87.40	79.20	79.20	80.00	76.00	74.20	74.60	73.60	71.00
MAR	84.68	84.76	85.18	85.10	79.92	79.92	80.06	77.96	69.24	70.56	69.20	68.80
SD-AR	2.06	1.93	2.27	2.28	2.01	2.05	1.81	2.00	3.40	2.47	3.18	2.71

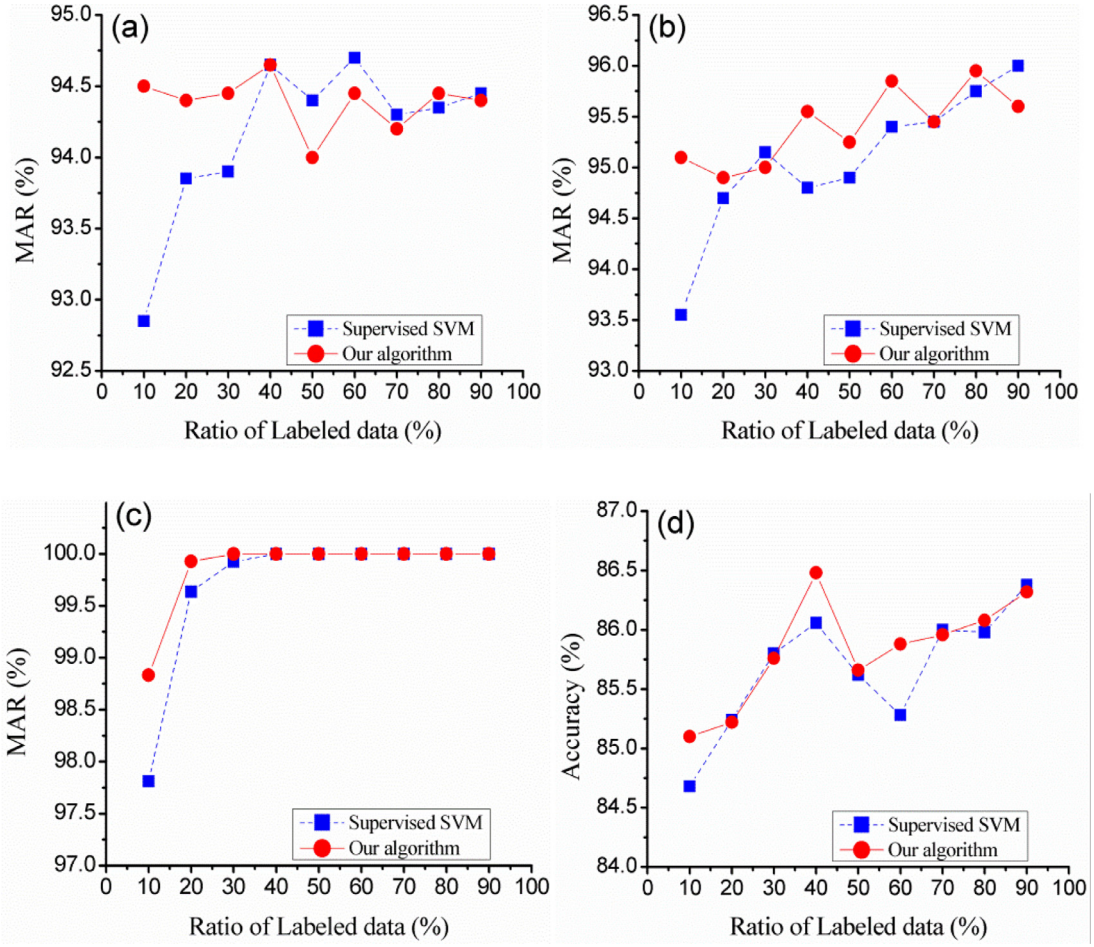

Fig. 5. Test *MAR* of our algorithm and supervised SVM with respect to the ratio of labeled data on different datasets: (a) Gauss50, (b) Gauss50x, (c) Banknote authentication, and (d) Waveform.

We find that the self-labeled SSC algorithms fail to improve the generalization capacity of the supervised algorithm on dataset Waveform, even our algorithm has the worst performances. In order to analyze the reasons, non-classical forms of multidimensional scaling is used to visualize the dissimilarity data to reveal the distribution of the four datasets [29]. As can be seen from Fig. 4, the data with different classes can be distinguished by the distribution charts on datasets of Gauss50, Gauss50x, and banknote authentication. By contrast, the dataset of Waveform has large amounts of strongly overlapping data, which is not suitable for the basis that the similarity between data points is evaluated by the distance measurement. Thus, we cannot distinguish the data with different classes in the overlapping area by the distribution charts. This may be the reason that our algorithm has the worst performances on dataset Waveform.

In summary, we conclude that our algorithm is more effective than semi-supervised Tri-training and semi-supervised FCM to improve the performance of a supervised algorithm on datasets Gauss50, Gauss50x, and banknote authentication. We believe the reason is that our algorithm is more appropriate for the non-spherical distribution data than the other two self-labeled SSC algorithms. Especially, our algorithm with supervised algorithm SVM has the better performance than that with supervised algorithm KNN or CART. We analyze the reason may be that the learning strategy of our algorithm, which iteratively learn a classifier based on the structure of data space discovered by finding the density peaks of data, is similar with the SVM's principle of structural risk minimization. Besides, we also find that our algorithm may lose its efficacy under the situation of dataset with large amounts of strongly overlapping data, like dataset waveform.

4.4. Impact of the ratio of labeled data

We also discuss the behavior of our algorithm with respect to the ratio of labeled data. We designed the experiments to compare our algorithm with supervised algorithm SVM. Fig. 5 shows the results when we increase the initial ratio of labeled data from 10% to 100%. 10-fold cross-validation strategy is also used to determine the final experimental results. As shown in Fig. 5, the *MAR* of the two algorithms increases as the ratio of labeled data increases because more labeled data are used to train the classifier. Generally, our algorithm achieves the higher *MAR* than the supervised algorithm SVM. Moreover, our algorithm also performs less dependence on the initial ratio of labeled data. However, when the initial labeled data can represent the whole data space, our algorithm may have worse performance than the supervised algorithm SVM because unlabeled data actually increase the probability of over fitting. In these cases, unlabeled data will not always help to train a classifier.

4.5. Impact of the cutoff distance d_c

In formula (1), we can see that the computing results depend on the cutoff distance d_c. This section will analyze the performance of our algorithm with the changes of d_c. The cutoff distance d_c is

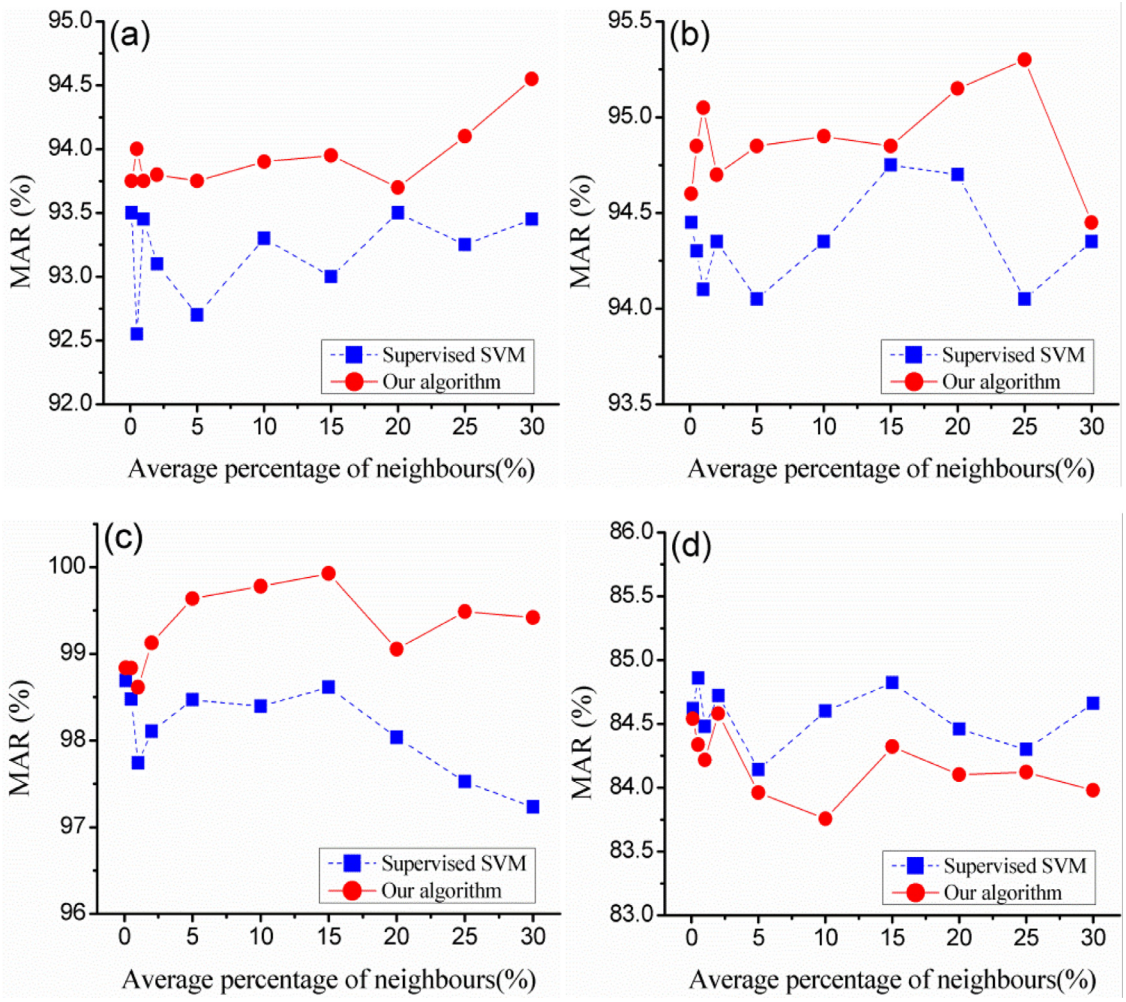

Fig. 6. The *MAR* of our algorithm and supervised SVM with respect to choice of average percentage of neighbors on different datasets: (a) Gauss50, (b) Gauss50x, (c) Iris, and (d) Waveform.

Table 7
The average values of rankings and their corresponding *p*-values computed by Friedman test on Tables 3–6.

Comparison algorithms and the *p*-value	Gauss50			Gauss50x			Banknote authentication			Waveform		
	SVM	KNN	CART	SVM	KNN	CART	SVM	KNN	CART	SVM	KNN	CART
Supervised algorithm *X*	1.35	2.10	2.35	1.60	2.05	2.20	1.85	2.20	2.20	2.20	**2.85**	2.50
A1 with *X*	2.60	2.35	2.35	2.10	2.15	**3.20**	2.35	2.65	**3.35**	2.30	2.80	**3.15**
A2 with *X*	2.65	1.95	2.35	2.95	2.70	2.05	2.35	2.05	1.90	**2.85**	2.75	2.25
Our-A with *X*	**3.40**	**3.60**	**2.95**	**3.35**	**3.10**	2.55	**3.45**	**3.10**	2.55	2.65	1.60	2.10
Corresponding *p*-value	**0.002**	**0.013**	0.647	**0.007**	0.197	0.190	**0.010**	**0.032**	**0.044**	0.612	**0.007**	0.271

set as:

$$D = sort\left(d_{ij}\right), d_c = D(\lfloor N_{T_R} \times P_a \rfloor) \quad (8)$$

where *sort(x)* means that x is sorted in ascending order, N_{TR} is the number of samples of *TR* set, P_a is the average percentage of neighbors in T_R. 10-fold cross-validation strategy and an initial ratio 10% of labeled data are also used in this section. The P_a is increased from 0.1% to 30% and the results are recorded in Fig. 6. As mentioned in [29], the density peaks clustering algorithm is sensitive only to the relative magnitude of ρ_i in different points, implying that, for large data sets, the results of the analysis are robust with respect to the choice of d_c. Fig. 6 shows the mutually consistent results in our experiments, which means that our algorithm is also robust with respect to the P_a in the range of 0.1%–30%. According to Fig. 6, our algorithm shows the relative unstable performance during the initial steps, whilst it predictably shows a robust performance in general in the range of 1%–20%. As a rule of thumb, one can choose d_c so that the average percentage of neighbors P_a is around 1%–2% of the total number of data points in the dataset, which is consistent with the [29].

4.6. Impact of noise

As is well known, one of the weaknesses of SSC algorithm is that they are sensitive to noise in the raw data. For addressing this issue, random noise has been added into the T_R to compare our algorithm with supervised algorithm SVM. Concretely, we increase the percentage of noise from 5% to 30% in the experiments. 10-fold cross-validation strategy and an initial ratio 10% of labeled data are also used on the four datasets. As shown in Fig. 7, the

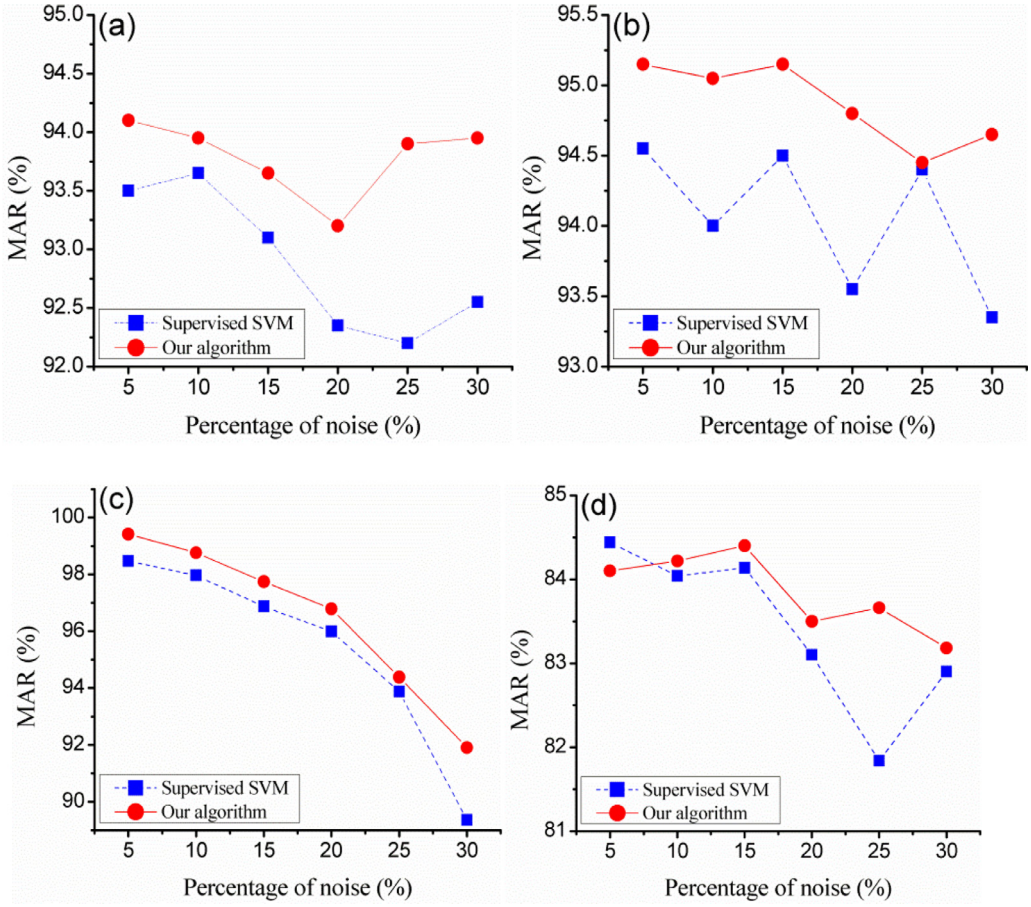

Fig. 7. The *MAR* of our algorithm and supervised SVM with respect to noise on different datasets: (a) Gauss50, (b) Gauss50x, (c) Iris, and (d) Waveform.

MAR of the two algorithms decrease as expected with the increase of percentage of noise. However, we find that the *MAR* of our algorithm decrease less than that of supervised algorithm SVM. For example, the *MAR* of our algorithm decreases from 94.10% to 93.95%, while the *MAR* of supervised algorithm SVM decreases from 93.5% to 92.55%, when the percentage of noise is increased from 5% to 30% on dataset Gauss50. Thus, we conclude that our algorithm is less sensitive to noise than supervised algorithm SVM.

5. Conclusions

In this paper, we introduce a method to discover the spherical or non-spherical distribution structure of data space based on find of density peaks at first. Then, a framework for self-training SSC is proposed, in which the structure of data space is integrated into the self-training process of SSC to help train a better classifier. Especially, we also concluded the algorithm pseudo-code of our proposed framework. Concretely, two representative self-labeled SSC algorithms are chosen to compare with our proposed algorithm under the situations that different supervised algorithms of SVM, KNN, and CART are used as the base classifier. A series of experiments on both artificial and real datasets are run to evaluate their performances. Besides, some other issues, such as impact of the ratio of labeled data, impact of the cutoff distance d_c, and impact of noise, are also analyzed in this paper. According to the experimental results we conclude that our proposed algorithm has better performance than the two chosen self-labeled SSC algorithms, especially when the distribution of data is non-spherical. In addition, we also find that our algorithm with supervised algorithm SVM performs better than that with supervised algorithm KNN or CART.

In the future, we plan to carry out the follow works to further improve the performance of our proposed algorithm. First, we will research a technology to automatically extract threshold value of cutoff distance d_c [37], rather than setting by subjective experience. Second, we will adopt the synthetic examples generation technology to improve the classification performance of our proposed algorithm [38]. Final, we will develop our algorithm to process quality-of-service data [35,36,43] based on optimization algorithm, such as particle swarm optimization [41,42].

Acknowledgment

This work is supported by the National Science and Technology Major Project (2014ZX07104-006), the National Natural Science Foundation of China (NSFC) (No. 61272060 and No. 91646114), and the Hundred Talents Program of CAS (No. Y500091BR1). We would like to express thanks to the anonymous reviewers for their invaluable comments and suggestions.

References

[1] N. Zeng, Z. Wang, H. Zhang, W. Liu, F.E. Alsaadi, Deep belief networks for quantitative analysis of a gold immunochromatographic strip, Cognit. Comput. 8 (4) (2016) 684–692.
[2] N. Zeng, Z. Wang, H Zhang, Inferring nonlinear lateral flow immunoassay state-space models via an unscented Kalman filter, Sci. China. Inform. Sci. 59 (11) (2016) 112204.
[3] N. Zeng, H. Zhang, W. Liu, et al., A switching delayed PSO optimized extreme learning machine for short-term load forecasting, Neurocomputing. 240 (31) (2017) 175–182.
[4] Y. Cao, H. He, H. Huang, LIFT: a new framework of learning from testing data for face recognition, Neurocomputing. 74 (6) (2011) 916–929.
[5] F. Pan, J. Wang, X. Lin, Local margin based semi-supervised discriminant embedding for visual recognition, Neurocomputing. 74 (5) (2011) 812–819.

[6] Z. Qi, Y. Xu, L. Wang, Y. Song, Online multiple instance boosting for object detection, Neurocomputing. 74 (10) (2011) 1769–1775.
[7] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, San Mateo, CA, USA, 2011.
[8] X. Zhu, Semi-Supervised Learning Literature Survey, University of Wisconsin–Madison, 2008 Computer Sciences Technical Report 1530.
[9] F. Schwenker, E. Trentin, Pattern classification and clustering: a review of partially supervised learning approaches, Pattern Recognit. Lett. 37 (2014) 4–14.
[10] I. Triguero, S. García, F. Herrer, Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study, Knowl. Inf. Syst. 42 (2) (2015) 245–284.
[11] G. Wang, F. Wang, T. Chen, D.Y. Yeung, F. Lochovsky, Solution path for manifold regularized semisupervised classification, IEEE Trans. Syst. Man. Cybern. B. 42 (2) (2012) 308–319.
[12] J. Wang, T. Jebara, S.F. Chang, Semi-supervised learning using greedy max-cut, J. Mach. Learn. Res. 14 (1) (2013) 771–800.
[13] A. Fujino, N. Ueda, K. Saito, Semisupervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle, IEEE Trans. Pattern Anal. Mach. Intell. 30 (3) (2008) 424–437.
[14] Y.H Shao, W.J. Chen, N.Y. Deng, Nonparallel hyperplane support vector machine for binary classification problems, Inf. Sci. 263 (2014) 22–35.
[15] K. Chen, S. Wang, Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions, IEEE Trans. Pattern Anal. Mach. Intell. 33 (1) (2011) 129–143.
[16] Q. Wang, P. Yuen, G. Feng, Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions, Pattern Recognit. 46 (9) (2013) 2576–2587.
[17] P.K. Mallapragada, R. Jin, A. Jain, Y. Liu, Semiboost: boosting for semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2000–2014.
[18] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of Annual ACM Conference Computing Learning Theory, New York, NY, USA, 1998, pp. 92–100.
[19] J. Du, C.X. Ling, Z.H. Zhou, When does co-training work in real data? IEEE Trans. Knowl. Data Eng. 23 (5) (2011) 788–799.
[20] Z. Jiang, S.Y. Zhang, J.P. Zeng, A hybrid generative/discriminative method for semi-supervised classification, Knowl. Based Syst. 37 (2013) 137–145.
[21] Z.H. Zhou, M. Li, Semi-supervised learning by disagreement, Knowl. Inf. Syst. 24 (3) (2010) 415–439.
[22] G. Jin, R. Raich, Hinge loss bound approach for surrogate supervision multi-view learning, Pattern Recognit. Lett. 37 (2014) 143–150.
[23] S.L. Sun, A survey of multi-view machine learning, Neural Comput. Appl. 23 (7–8) (2013) 2031–2038.
[24] M. Li, Z.H. Zhou, SETRED: self-training with editing, in: Proceedings of the 9th Pacific-Asia Conference, PAKDD, Hanoi, Vietnam, 2005, pp. 611–621.
[25] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, 1995, pp. 189–196.
[26] E. Riloff, J. Wiebe, T. Wilson, Learning subjective nouns using extraction pattern bootstrapping, in: Proceedings of the Seventh Conference on Natural Language Learning, 2003, pp. 25–32.
[27] M.M. Adankon, M. Cheriet, Help-training for semi-supervised support vector machines, Pattern Recognit. 44 (9) (2011) 2220–2230.
[28] H. Gan, N. Sang, R. Huang, et al., Using clustering analysis to improve semi-supervised classification, Neurocomputing 101 (2013) 290–298.
[29] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 44 (6191) (2014) 1492–1496.
[30] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[31] C. Chang, C. Lin, LIBSVM: A Library for Support Vector Machines, Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html, 2011.
[32] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66.
[33] L.I. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and regression trees (cart), Biometrics. 40 (3) (1984) 358.
[34] C.C. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, Available at http://archive.ics.uci.edu/ml/datasets.html, 2010.
[35] X. Luo, M.C. Zhou, S. Li, Y.N. Xia, Z.H. You, Q.S. Zhu, H. Leung, Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data, IEEE Trans. Cybern. (2017), doi:10.1109/TCYB.2017.2685521.
[36] X. Luo, M. Zhou, Z. Wang, Y. Xia, Q. Zhu, An effective QoS estimating scheme via alternating direction method-based matrix factorization, IEEE Trans. Serv Comput. (2017), doi:10.1109/TSC.2016.2597829.
[37] X.F. Wang, Y.F Xu, Fast clustering using adaptive density peak detection, Stat. Methods Med. Res. 0 (0) (2015) 1–14.
[38] I. Triguero, S. García, F. Herrera, SEG-SSC: a framework based on synthetic examples generation for self-labeled semi-supervised classification, IEEE Trans. Cybern. 45 (4) (2015) 622–634.
[39] Z.-H. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, IEEE. Trans. Knowl. Data Eng. 17 (11) (2005) 1529–1541.
[40] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[41] N. Zeng, Z. Wang, H. Zhang, F.E. Alsaadi, A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay, Cognit. Comput. 8 (2) (2016) 143–152.
[42] N. Zeng, Z. Wang, Y. Li, M. Du, X. Liu, A hybrid EKF and switching PSO algorithm for joint state and parameter estimation of lateral flow immunoassay models, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (2) (2012) 321–329.
[43] X. Luo, M.C. Zhou, Y. Xia, Q. Zhu, A.C. Ammari, A. Alabdulwahab, Generating highly accurate predictions for missing QoS-data via aggregating non-negative latent factor models, IEEE Trans. Neural Netw. Learn. Syst. 27 (3) (2016) 579–592.

Di Wu received the B.S. degree in applied physics from Nanjing University of Science and Technology, Jiangsu, China, and the M.S. degree in optical engineering from the Chongqing University, Chongqing, China in 2009 and 2012, respectively. He is currently working toward the Ph.D. degree of computer science in Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China. His research interests include data mining, machine learning and their applications.

Mingsheng Shang received his Ph.D. degree in computer science from University of Electronic Science and Technology of China (UESTC). He joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China in 2015, and is currently a Professor of computer science and engineering. His research interests include data mining, complex networks, cloud computing and their applications.

Xin Luo received the B.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2011. He joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China in 2016, and is currently a Professor of computer science and engineering. His research interests include big data analysis and intelligent control. He has published 60+ papers (including 10+ IEEE TRANSACTIONS papers) in his related areas.

Ji Xu received the B.E. degree from Beijing Jiaotong University, Beijing, in 2004, and the M.S. degree from Tianjin Normal University, Tianjin, China, in 2008. Now he is a Ph.D. candidate with Southwest Jiaotong University, Chengdu, China. His research interests include data mining, granular computing and software engineering.

Huyong Yan received the B.E degree in computer science and technology from university of Electronic Science and Technology of China, Chengdu, China, and the M.S. degree in petroleum engineering computing technology from Northeast Petroleum University, Daqing, China, in 2007 and 2013, respectively. He is currently working toward the Ph.D. degree in Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China. His research interests include data mining, rough set and water eutrophication assessment.

Weihui Deng received the B.E. degree in information and computing science from Hunan University of Science and technology, Xiangtan, China, in 2012. He is currently working toward the Ph.D. degree in Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Science, Chongqing, China. His research interests include data mining, granular computing and time series analysis.

Guoyin Wang received the B.E. degree in computer software, the M.S. degree in computer software, and the Ph.D. degree in computer organization and architecture from Xi'an Jiaotong University, Xi'an, China, in 1992, 1994, and 1996, respectively. He worked at the University of North Texas, USA, and the University of Regina, Canada, as a visiting scholar during 1998–1999. Since 1996, he has been working at the Chongqing University of Postsand Telecommunications, where he is currently a professor and Ph.D. supervisor, the Director of the Chongqing Key Laboratory of Computational Intelligence, and the Dean of the College of Computer Science and Technology. His research interests include data mining, machine learning, rough set, granular computing, soft computing, cognitive computing, etc. He is the President of International Rough Set Society (IRSS), a Vice-President of the Chinese Association for Artificial Intelligence (CAAI), a council member of the China Computer Federation (CCF), and a senior member of IEEE.