A Simultaneous Learning Framework for Clustering and Classification

Weiling Cai [1,2] *Songcan Chen* [1,3*] *Daoqiang Zhang* [1]

[1] (*Department of Computer Science & Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, P. R. China*)

[2] (*Department of Mathematics and Computer Science, Nanjing Normal University, Nanjing 210097, P. R. China*)

[3] (*State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, P. R. China*)

Abstract—Traditional pattern recognition generally involves two tasks: unsupervised clustering and supervised classification. When class information is available, fusing the advantages of both clustering learning and classification learning into a *single* framework is an important problem worthy of study. To date, most algorithms generally treat clustering learning and classification learning in a *sequential* or two-step manner, *i.e.*, first execute clustering learning to explore structures in data, and then perform classification learning on top of the obtained structural information. However, such sequential algorithms can not always guarantee the simultaneous optimality for both clustering and classification learning. In fact, the clustering learning in these algorithms just aids the subsequent classification learning and does not benefit from the latter. To overcome this problem, a simultaneous learning framework for clustering and classification (SCC) is presented in this paper. SCC aims to achieve three goals: (1) acquiring the robust classification and clustering simultaneously; (2) designing an effective and transparent classification mechanism; (3) revealing the underlying relationship between clusters and classes. To this end, with the Bayesian theory and the *cluster posterior probabilities of classes*, we define a *single* objective function to which the clustering process is directly embedded. By optimizing this objective function, the effective and robust clustering and classification results are achieved *simultaneously*. Experimental results on both synthetic and real-life datasets show that SCC achieves promising classification and clustering results at one time.

Keywords—Structure in Data; Bayesian Theory; Clustering Learning; Classification Learning; Simultaneous Classification and Clustering Learning

* Corresponding author: Tel: +86-25-84896481-12106, Fax: +86-25-84498069. Email: s.chen@nuaa.edu.cn (S. C. Chen) caiwl@nuaa.edu.cn (W. L. Cai)

1. Introduction

Traditional pattern recognition involves two tasks: unsupervised clustering and supervised classification [1, 2]. In unsupervised clustering, samples without class labels are grouped into meaningful clusters. These clusters can be utilized to describe the underlying structure in data, which is helpful for better understanding of data. In supervised classification, samples with class labels are used to build the classification mechanism, through which class labels can be provided for new samples.

When class information is available, most traditional classifiers are designed in a *direct* way by employing supervised information to determine their decision functions. Such classifiers usually provide only class labels for new samples, but rarely care about the revelation of data distribution. For example, multi-layer perceptron (MLP) [3] and support vector machines (SVM) [4, 5] successfully utilize the class information of samples to achieve high classification accuracies; however, they emphasize more the classification of the data than the revelation of the data distribution, thus fail to interpret the obtained classification results well.

In contrast to these classifiers, another type of classifiers is designed in an *indirect* way by incorporating structural information into their classification schemes. Since clustering analysis is appropriate for exploring the data distribution [1, 2], these classifiers usually first perform clustering to uncover the underlying structure in data, and then design classification rules based on the obtained structural information. In this way, these classifiers fuse the advantages of both clustering learning and classification learning together to some extent. On the other hand, clustering methods can be roughly categorized into unsupervised ones and supervised ones, depending on whether using class labels or not. Thus, as shown in Fig. 1, the classifier design based on the clustering methods can also be categorized into two types: unsupervised-clustering plus classifier-design and supervised-clustering plus classifier-design.

Radial Basis Function neural network (RBFNN) [6] is a classical algorithm belonging to the first category, i.e., unsupervised-clustering plus classifier-design. To determine the parameters of the hidden layer in RBFNN, training samples are clustered in an unsupervised way by using c-means or fuzzy c-means (FCM) [7]. Then, the connection weights between the hidden and output layers are optimized by minimizing the mean squared error (MSE) criterion between the target and actual outputs. Here, clustering makes RBFNN yield good generalization [3], but its function is just to help determine the parameters of the neural network, rather than explore the underlying structure of the input space. In fact, RBFNN can not really inherit the merits of both clustering learning and classification learning as shown below.

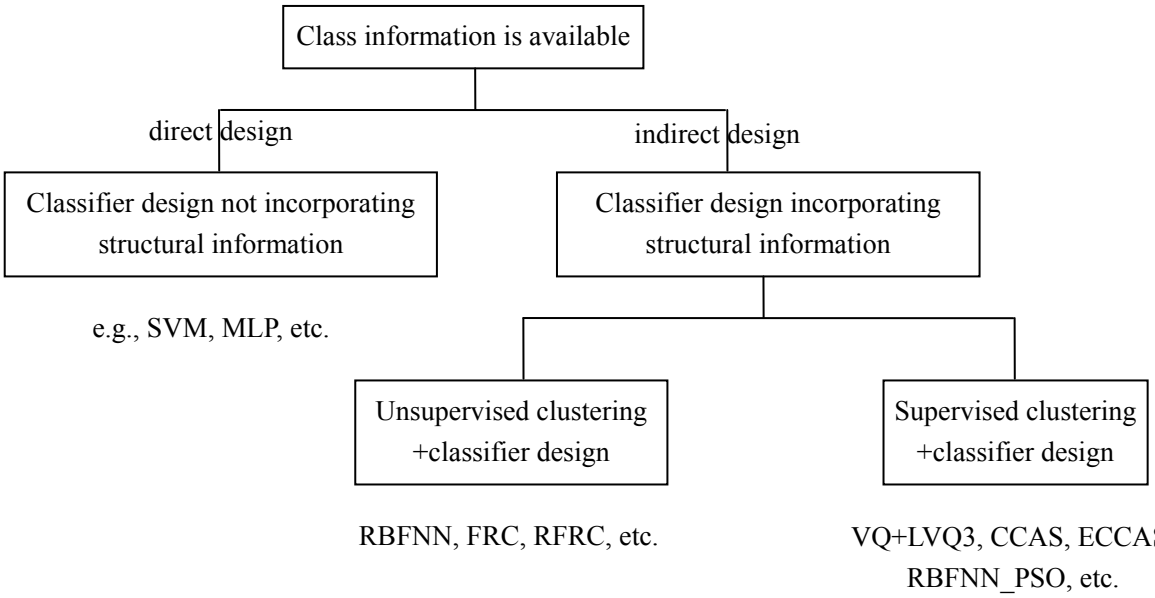


Fig. 1 A taxonomy of the classifier design when the class information is available

Recently, some fuzzy relation based methods are proposed to bridge clustering and classification [8-9], which also belong to the first category. Setnes et al. [8] proposed FRC to represent a transparent alternative to conventional black-box techniques such as neural networks. To enhance FRC's robustness, in one of our previous works, we developed RFRC [9] by replacing FCM and hard class labels with Kernelized FCM (KFCM) [10, 11] and soft labels, respectively. The training of both algorithms includes two steps. First, unsupervised clustering is performed on training samples to discover the natural structure in data. Then, a relation matrix **R** between the obtained clusters and given class labels is established to reflect the logical relationship between clusters and classes. Here this matrix **R** plays a role of a connection weight matrix as in RBFNN. However, such relationship in both FRC and RFRC is directly constructed by the logical composite operator rather than by optimizing some defined criterion function. As a result, the clustering and classification results cannot be simultaneously optimal. In addition, the entries in the relation matrix **R** lack the statistical characteristic, and thus fail to indicate the relative reliability of the obtained relationship. Moreover, it is difficult to optimize these entries by defining an objective function due to the indifferentiability of the composite operators.

On the other hand, following the second design line, the classifier design process can be described as supervised-clustering plus classifier-design, i.e., using supervised clustering to generate the proper prototypes, and then adopting the k Neighbor-Nearest (kNN) or weighted kNN classification rule to classify new samples on top of the prototypes. Pedrycz [12] proposed a supervised clustering method by

directly incorporating the given class labels into the clustering objective function, which aims to make a trade-off between the compactness of the structures and the accuracy of class information. This method needs to predetermine the logical relationship between clusters and classes. However, if there is no prior knowledge available, the correspondences between all the clusters and classes can only be exhaustively determined by enumerating all the permutation and combination, which is a NP hard problem and thus also intractable. Kim and Oommen [13] proposed an algorithm called VQ+LVQ3 which utilizes both the positions and class labels of the cluster centers to classify new samples, and thus does not need to predetermine the relationship between clusters and classes. Similar to VQ+LVQ3, a supervised clustering and classification algorithm named CCAS [14, 15] and its extended version ECCAS [16] also fall into such a two-step framework of supervised clustering plus classifier design. Since VQ+LVQ3 and CCAS (or ECCAS) adopt the 1NN and the weighted kNN classifiers in their classifier design phase, respectively, they actually do not need to experience any training, in other words, both VQ+LVQ3 and CCAS (or ECCAS) do not have a true design phase.

It is worth noting that all above algorithms have a common point: *sequentially* optimizing the clustering and classification objective functions respectively. That is, the clustering learning obtains a description of the underlying data distribution, and then the classification learning uses the obtained information to train the classification rules. In these algorithms, although the clustering learning and classification learning *separately* optimize their own criteria, such kind of sequential learning manner can not always guarantee simultaneous optimality for both clustering and classification learning. In fact, the clustering learning here just aids the classification learning and does not benefit from the classification learning.

To compensate for this shortcoming, a simultaneous learning framework for clustering and classification (SCC) is presented in this paper. In its implementation, the Bayesian theory and the *cluster posterior probabilities of classes* is employed to create a bridge between clustering learning and classification learning. Based on this, a *single* objective function to which the clustering process is directly embedded is designed to evaluate both clustering and classification performance. By optimizing it, the robust and effective classification and clustering learning are *simultaneously* achieved. It is worth mentioning that under the guidance of supervision information, the clustering process can avoid the unfavorable influence of the outliers, and thus the classification learning based on unbiased clustering result can also resist the effects of outliers. Moreover, SCC provides the statistical relationship between clusters and classes, so that we gain some meaningful insights to make SCC prone to be transparent. For example, we can know that

whether the formed clusters are pure or not, whether the class of the dataset is composed of single-group or multi-groups and so on. Due to the generality of SCC, different distance metrics can be adopted to result in different algorithms. By using the Euclidian distance, SCC1 is proposed which is only suitable for the dataset with spherical distribution. To make SCC more appropriate for the non-spherical distribution, SCC2 is developed by adopting a kernel-based metric [4, 17]. In summary, SCC achieves the three goals at one time: (1) robustly clustering the data with guidance of supervision information; (2) designing an effective and transparent classification algorithm; (3) adaptively revealing an underlying relationship between clusters and classes. The comparative experiments on both synthetic and real-life datasets show that SCC effectively and simultaneously achieve all above goals within a single framework.

The rest of this paper is organized as follows: In Section 2, Robust Fuzzy Relational Classifier is reviewed. In section 3, the simultaneous learning framework for clustering and classification is described. The experimental results on 2 synthetic datasets and the 20 real-life benchmark datasets are presented in section 4. Finally, the conclusions are given in section 5.

2. Robust fuzzy relational classifier

In RFRC, a fuzzy relation matrix **R** is built to connect the unsupervised clustering and supervised classification. However, this **R** is constructed by the composite operator, which leads to several disadvantages: (1) the optimization of the **R** is difficult due to the indifferentiability and complexity of the composite operators as shown in formulas (4-6) below; (2) the **R** entries lack a statistical characterization, and thus fail to indicate the reliability of the obtained relationship between clusters and classes; (3) when the training dataset contains inconsistent class information, the **R** entries approach to 0, and thus fail to reflect any discriminant information.

2.1 Training of the classifier

The training of the classifier consists of two steps. In the first step, the previously-proposed kernelized FCM (KFCM) is applied to reveal the natural structures in the given dataset. In the second step, a relation matrix **R** is established from the obtained fuzzy partition and the soft class labels to formulate the relationship between clusters and classes. In what follows, we first give a brief description for KFCM.

The theoretical basis of KFCM is the kernel trick, which aims at converting the nonlinear problem in the original low dimensional input space into a linear one in the higher dimensional feature space [18]. By

using an implicit nonlinear map ϕ, the given sample in the input space is mapped into a rather higher dimensional feature space F:

$$\phi : \mathbf{z} \to \phi(\mathbf{z}) \in F . \tag{1}$$

To evaluate the intra-cluster compactness in the feature space, the objective function of KFCM is described as follows

$$J_{KFCM}(\mathbf{U},\mathbf{V}) = \sum_{j=1}^{K}\sum_{i=1}^{N} u_{ji}^{m} \left\| \phi(\mathbf{x}_i) - \phi(\mathbf{v}_j) \right\|^2 , \tag{2}$$

where N is the total number of the training samples and K the number of clusters; let $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2,\ldots, \mathbf{x}_N\}$ and $\mathbf{V}=\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K\}$ denote the training set and cluster centers (or prototypes), respectively, where $\mathbf{x}_i \in \mathrm{R}^d$ and $\mathbf{v}_j \in \mathrm{R}^d$; and the fuzzy matrix $\mathbf{U}=(u_{ji})_{K\times N}$ makes up of the fuzzy memberships of the each training sample $\mathbf{x}_i$ to each cluster $\mathbf{v}_j$. By definition, each sample $\mathbf{x}_i$ satisfies the constraint $\sum_{j=1}^{K} u_{ji} = 1$. The parameter m ($1 \le m < \infty$) is a weighting exponent on each fuzzy membership that determines the amount of fuzziness of the resulting classification. In the following experiment, the value of m is set to 2. It is worth emphasizing that the distance metric induced by RBF kernel is robust [19] in terms of Huber's robust statistics [20], so that the RBF kernel is adopted to guarantee the robustness of KFCM.

Then, a fuzzy relation matrix **R** is established from the obtained cluster membership matrix **U** and the soft class labels. This **R** can be represented by a $K\times L$ matrix

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1L} \\ r_{21} & r_{22} & \ldots & r_{2L} \\ \ldots & \ldots & \ldots & \ldots \\ r_{K1} & r_{K2} & \ldots & r_{KL} \end{bmatrix} \tag{3}$$

where r_{jl} represents the fuzzy relationship between the jth cluster and the lth class and L is the number of classes. By using a fuzzy conjunction operator, the partial relation $\mathbf{R}_i$ with respect to the $\mathbf{x}_i$ can be aggregated into the **R**:

$$\mathbf{R} = \bigcap_{i=1}^{N} \mathbf{R}_i \tag{4}$$

where the entries of **R** is obtained respectively according to

$$r_{jl} = \min_{i=1,2,\cdots,N} [(r_{jl})_i] . \tag{5}$$

To compute the $(r_{jl})_i$ in the formula (5), the φ-composition operator [21] is adopted

$$(r_{jl})_i = \min(1, 1 - u_{ji} + y_{li}), \quad l = 1, 2, \cdots, L, \quad j = 1, 2, \cdots, K \tag{6}$$

where y_{li} is the class membership of the ith sample to the lth class. In FRC, y_{li} is the hard class label and its value takes from {0, 1}. Here in RFRC, y_{li} is the soft class label and its value becomes a fuzzy value between 0 and 1. Compared to the hard class labels, the soft ones are able to characterize the membership degrees of the training samples relative to the different classes, and thus more precisely reflect class information. The effectiveness of adopting such a labeling is carefully proved in [9].

2.2 Classification of new samples

The classification of a new sample $\mathbf{x}$ proceeds in three steps. First, the cluster membership $\hat{\mathbf{u}}_x = [\hat{u}_{1x}, \hat{u}_{2x}, \ldots, \hat{u}_{jx}, \ldots, \hat{u}_{Kx}]$ is computed by measuring the distances between the $\mathbf{x}$ and the cluster centers in the feature space

$$\hat{u}_{jx} = \frac{\left(\left\| \phi(\mathbf{x}) - \phi(\mathbf{v}_j) \right\|^2 \right)^{-1/(m-1)}}{\sum_{j=1}^{c} \left(\left\| \phi(\mathbf{x}) - \phi(\mathbf{v}_j) \right\|^2 \right)^{-1/(m-1)}}. \tag{7}$$

Then, using the fuzzy relational composition, the class membership $\hat{\mathbf{y}}_x = [\hat{y}_{1x}, \hat{y}_{2x}, \ldots, \hat{y}_{lx}, \ldots, \hat{y}_{Lx}]$ is obtained by fuzzy relational composition

$$\hat{\mathbf{y}}_x = \hat{\mathbf{u}}_x \circ_T \mathbf{R}, \tag{8}$$

where $\circ_T$ is the sup-t composition operator [22]. Each component in the vector $\hat{\mathbf{y}}_x$ is obtained from

$$\hat{y}_{lx} = \max_{1 \le j \le K} [\max(\hat{u}_{jx} + r_{jl} - 1, 0)], \quad l = 1, 2, \cdots, L. \tag{9}$$

Finally, the class membership $\hat{\mathbf{y}}_x$ is defuzzified using the maximum operator to obtain a crisp decision for classification

$$\hat{\omega}_x = \arg\max_{1 \le l \le L} \hat{y}_{lx} \tag{10}$$

where $\hat{\omega}_x$ is the final class label.

3. Simultaneous learning framework for clustering and classification

3.1 Description of the proposed algorithm

In SCC, a classification mechanism depending only on the clustering centers is developed at first. Then,

based on this mechanism, a single objective function is designed to evaluate both the classification and clustering ability. Finally, an evolutionary technique called modified particle swarm optimizer (PSOm) [27] is adopted to optimize this objective function.

3.1.1 Classification mechanism

In a classification problem, if the posterior probabilities $p(\omega_l|\mathbf{x}_i)$ for each class is modeled, the output class label for $\mathbf{x}_i$ can be determined by

$$f\left(\mathbf{x}_i\right) = \arg\max_{1\le l\le L} p\left(\omega_l \mid \mathbf{x}_i\right). \tag{11}$$

In order to incorporate the cluster information into $p(\omega_l|\mathbf{x}_i)$, we resort to the formed clusters $\{c_j\}$ to reformulate $p(\omega_l|\mathbf{x}_i)$ through the total probability theorem

$$\begin{aligned} p\left(\omega_l \mid \mathbf{x}_i\right) &= \sum_{j=1}^{K} p\left(\omega_l, c_j \mid \mathbf{x}_i\right) \\ &= \sum_{j=1}^{K} p\left(c_j \mid \mathbf{x}_i\right) p\left(\omega_l \mid c_j, \mathbf{x}_i\right), \\ &= \sum_{j=1}^{K} p\left(c_j \mid \mathbf{x}_i\right) p\left(\omega_l \mid c_j\right) \end{aligned} \tag{12}$$

where ω_l denotes the lth class, c_j represents the jth cluster, $p(c_j|\mathbf{x}_i)$ is the posterior probabilities of the presence of corresponding samples in the input space and $p(\omega_l|c_j)$ denotes the cluster posterior probabilities of class membership. Notice that $p(\omega_l|c_j, \mathbf{x}_i)$ has no relationship with $\mathbf{x}_i$, and thus can be simplified as $p(\omega_l|c_j)$.

Thanks to the generality of (12), $p(c_j|\mathbf{x}_i)$ can be computed in different forms according to different clustering methods. When using Gaussian Mixture model (GMM) [23] as a clustering model, $p(c_j|\mathbf{x}_i)$ is computed by

$$\begin{aligned} p\left(c_j \mid \mathbf{x}_i\right) &= \frac{p\left(\mathbf{x}_i \mid c_j\right) p\left(c_j\right)}{p\left(\mathbf{x}_i\right)} \\ &= \frac{1}{\sqrt{(2\pi)\left|\mathbf{\Sigma}_j\right|} p\left(\mathbf{x}_i\right)} \exp\left(-\frac{\left(\mathbf{x}_i - \mathbf{u}_j\right)^T \mathbf{\Sigma}_j^{-1} \left(\mathbf{x}_i - \mathbf{u}_j\right)}{2} \right) p\left(c_j\right) \end{aligned} \tag{13}$$

where $\mathbf{u}_j$ and $\mathbf{\Sigma}_j$ are two parameters of the jth Gaussian model. When using the fuzzy clustering algorithms such as FCM and KFCM, $p(c_j|\mathbf{x}_i)$ is calculated by

$$p\left(c_j \mid \mathbf{x}_i\right)=\frac{dist\left(\mathbf{x}_i, \mathbf{v}_j\right)^{-1}}{\sum_{r=1}^{K} dist\left(\mathbf{x}_i, \mathbf{v}_r\right)^{-1}} \tag{14}$$

In fact, the final $\{\mathbf{u}_j, \mathbf{\Sigma}_j\}$ in (13) and $\{\mathbf{v}_j\}$ in (14) can be obtained respectively by optimizing their own objective function in GMM or fuzzy clustering algorithms. However, in this paper what we concern is the general representation form of $p(c_j|\mathbf{x}_i)$ from clustering algorithms. From (13) and (14), we can observe that its representation depends only on the cluster parameters such as centers. Consequently, without loss of generality, we just adopt (14) to compute $p(c_j|\mathbf{x}_i)$.

In (12), the posterior probabilities of class membership $p(\omega_l|c_j)$ is computed through Bayesian theorem

$$p\left(\omega_l \mid c_j\right)=\frac{p\left(\omega_l, c_j\right)}{p\left(c_j\right)} \tag{15}$$

where $p(c_j)$ is the prior probability and $p(\omega_l, c_j)$ is the joint distribution. Here $p(c_j)$ is calculated by the proportion of the samples in the jth clusters, i.e., $Num(\mathbf{x}\in c_j)/N$; similarly, $p(\omega_l, c_j)$ is computed in terms of the proportion of the samples in the jth cluster and meanwhile in the lth class, denoted by Num ($\mathbf{x}\in\omega_l$ and $\mathbf{x}\in c_j$)/N. Therefore, $p(\omega_l|c_j)$ is transformed into

$$p\left(\omega_l \mid c_j\right)=\frac{Num\left(\mathbf{x} \in \omega_l \ \ and \ \mathbf{x} \in c_j\right)}{Num\left(\mathbf{x} \in c_j\right)}. \tag{16}$$

For each cluster c_j, the corresponding posterior probabilities of class membership satisfies that

$$\sum_{l=1}^{L} p\left(\omega_l \mid c_j\right)=1. \tag{17}$$

By examining (16), we can give such an intuitive interpretation that when $p(\omega_l|c_j)$ is large (small), the proportion of samples in cluster c_j from the class l is large (small). It is obvious that $p(\omega_l|c_j)$ reveals the statistical relationship between the formed clusters and the given classes. All $p(\omega_l|c_j)$s constitute a $K\times L$ matrix denoted by $\mathbf{P}$:

$$\mathbf{P}=\begin{bmatrix} p\left(\omega_1 \mid c_1\right) & p\left(\omega_2 \mid c_1\right) & \ldots & p\left(\omega_L \mid c_1\right) \\ p\left(\omega_1 \mid c_2\right) & p\left(\omega_2 \mid c_2\right) & \ldots & p\left(\omega_L \mid c_2\right) \\ \ldots & \ldots & \ldots & \ldots \\ p\left(\omega_1 \mid c_K\right) & p\left(\omega_2 \mid c_K\right) & \ldots & p\left(\omega_L \mid c_K\right) \end{bmatrix}. \tag{18}$$

The detailed analysis about this matrix is given in the subsection 3.2.2.

Note that the above classification mechanism is *only* relevant to the cluster centers $\{\mathbf{v}_j\}$. For a given training dataset with the class labels, $p(\omega_l|c_j)$ is just dependent on the clustering partition of the training

samples. Here the partition of the training samples is disjoint, that is, each sample only belongs to one cluster. Such a partition can be obtained by assigning each sample to the nearest clustering centers. Therefore, $p(\omega_l|c_j)$ in turn relies only on the clustering centers. Moreover, according to (14), $p(c_j|\mathbf{x}_i)$s are just dependent on the clustering centers as well. Thus, the posterior probabilities $p(\omega_l|x_i)$s are only determined by the clustering centers, which is a crucial starting point to realize the SCC algorithm.

3.1.2 Objective function

Based on the above formal description for the classification mechanism, a single objective function is designed to evaluate not only classification ability but also clustering ability. To this end, this objective function consists of two terms: misclassification rate and clustering impurity. Given the training samples $\{\mathbf{x}_i, y_i\}$ where $\mathbf{x}_i \in \mathrm{R}^d$ and $y_i \in \{1, 2, \cdots, L\}$, the objective function is formulated as

$$J(\{\mathbf{v}_j\}) = \sum_{i=1}^{N} \delta\left(f\left(\mathbf{x}_i\right), y_i\right) \Big/ N + \beta\, q\left(\mathbf{X}\right), \tag{19}$$

where δ is a loss function whose value is 0 when $f(\mathbf{x}_i)=y_i$, 1 otherwise. $q(\mathbf{X})$ denotes the clustering impurity

$$\begin{aligned} q\left(\mathbf{X}\right) &= 1 - \sum_{j=1}^{K} \max_{l=1,2,\ldots,L} p\left(\omega_l, c_j\right) \\ &= 1 - \sum_{j=1}^{K} \max_{l=1,2,\ldots,L} p\left(\omega_l \mid c_j\right) \times p\left(c_j\right) \\ &= 1 - \sum_{j=1}^{K} \frac{\max_{l=1,2,\ldots,L} p\left(\omega_l \mid c_j\right) \times Num(x \in c_j)}{N} \end{aligned} . \tag{20}$$

A lower value of $q(\mathbf{X})$ implies a better clustering and vice versa. The regularization parameter β in (19) gives a trade-off between the classification accuracy and the clustering purity, and its value is restricted to {0.01, 0.1, 1} throughout this paper. Since the values of the two terms in (19) all rely on the clustering partition on the training samples, the value of $J(\{\mathbf{v}_j\})$ is naturally dependent on *just* the set of the clustering centers $\{\mathbf{v}_j\}$. By optimizing the $\{\mathbf{v}_j\}$ in (19), the clustering impurity and the classification error is minimized simultaneously. Furthermore, it is observed from (19) that the clustering process is directly embedded to this objective function. However, different from a naïve unsupervised clustering process, this clustering process is executed with the guidance of the classification information. Due to this very characteristic, the clustering result in SCC can avoid the influence of the outliers to great extent. Consequently, SCC can robustly cluster the data in the supervised manner and this conclusion will be

experimentally demonstrated by the experiment in the subsection 4.1.

In the objective function of SCC, different distance metrics can be adopted to result in different algorithms. For example, by using the Euclidian distance and kernel-based distance, SCC1 and SCC2 are derived from SCC, respectively. In SCC1, the clustering region formed by Euclidian distance is represented as

$$\begin{aligned}\text{Region}_i &= \{\mathbf{x} \mid \|\mathbf{x}-\mathbf{v}_i\|^2 < \|\mathbf{x}-\mathbf{v}_j\|^2, i \neq j\} \\ &= \{\mathbf{x} \mid (\mathbf{v}_i-\mathbf{v}_j)^T \left(\mathbf{x}-\frac{\mathbf{v}_i-\mathbf{v}_j}{2}\right) > 0, i \neq j\}\end{aligned} . \tag{21}$$

Here the boundary between the regions is a hyper-plane which can only be induced by hyper-spherical clusters. Therefore, SCC1 is suitable to cluster and classify the dataset with spherical structures. To make SCC more appropriate for the dataset with non-spherical distribution, SCC2 is developed by using a kernel-induced distance metric [9, 10, 11]. A *kernel* is a function K that for all $\mathbf{x}$, $\mathbf{z}$ from the original input space satisfies

$$K(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \tag{22}$$

where $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ denotes the inner product operation. Through the kernel substitution, a novel distance is obtained

$$\begin{aligned}dist(\mathbf{x}_i, \mathbf{v}_j) &= \|\phi(\mathbf{x}_i) - \phi(\mathbf{v}_j)\|^2 \\ &= (\phi(\mathbf{x}_i) - \phi(\mathbf{v}_j))^T (\phi(\mathbf{x}_i) - \phi(\mathbf{v}_j)) \\ &= K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{v}_j, \mathbf{v}_j) - 2K(\mathbf{x}_i, \mathbf{v}_j)\end{aligned} , \tag{23}$$

in this way, a new class of non-Euclidean distance measures in original input space are induced. $K(\mathbf{x}, \mathbf{y})$ here is taken as the radial basis function (RBF) kernel due to its robustness [11]

$$K(\mathbf{x},\mathbf{y}) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right). \tag{24}$$

Therefore, the distance can be written as

$$dist(\mathbf{x}_i, \mathbf{v}_j) = 2 - 2\exp\left(\frac{-\|\mathbf{x}_i - \mathbf{v}_j\|^2}{\sigma^2}\right) \tag{25}$$

where σ is the kernel parameter and significantly affects the clustering result. In order to simplify the selection of this kernel parameter, we define the parameter σ in terms of [24]

$$\sigma^2 = \frac{\sum_{i=1}^{N} \left\| \mathbf{x}_i - \overline{\mathbf{x}} \right\|^2}{\lambda} \tag{26}$$

where λ is a scale factor and $\overline{\mathbf{x}} = \sum_{i=1}^{N} \mathbf{x}_i \Big/ N$. To get an appropriate value of σ, we indirectly determine it by seeking an appropriate scale factor λ in {0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15} according to the trial-and-error approach [24]. In SCC2, the clustering regions formed by the kernel-induced distance metric can be represented as below

$$\text{Region}_i = \{\mathbf{x} \mid \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \frac{\left\| \mathbf{x} - \mathbf{v}_i \right\|^{2n}}{\sigma^{2n}} < \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \frac{\left\| \mathbf{x} - \mathbf{v}_j \right\|^{2n}}{\sigma^{2n}}, i \neq j\} . \tag{27}$$

Here we have used

$$1 - K(\mathbf{x}, \mathbf{v}_i) = 1 - \exp\left(\frac{-\left\| \mathbf{x} - \mathbf{v}_i \right\|^2}{\sigma^2} \right) = 1 - \sum_{n=0}^{\infty} \frac{\left(-\left\| \mathbf{x} - \mathbf{v}_i \right\|^2 / \sigma^2 \right)^n}{n!} = \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \frac{\left\| \mathbf{x} - \mathbf{v}_i \right\|^{2n}}{\sigma^{2n}} . \tag{28}$$

From the above analysis, it is observed that the kernel trick can make SCC2 more likely adapt to non-spherical shape of distributions in data, which also accords with the conclusion obtained in [10, 11].

3.1.3 Optimization of objection function

Genetic Algorithms (GA) and Particles Swarm Optimization (PSO) are two prevailing population based optimization algorithms that have been proven to be successful in a wide variety of tasks. PSO can achieve a comparable performance but requires less computation time compared to GA [25]. Therefore, in this paper, we adopt a modified Particle Swarm Optimization (PSOm) [26, 27] to optimize the clustering centers in the objective function. PSOm is an evolutionary technique through individual improvement plus population cooperation and competition. Compared to the standard PSO [26], PSOm introduces an inertia weight w to balance the global search and local search [27]. This w is a positive fixed constant or even a decreasing function of time. A large w facilitates a global search while a small w facilitates a local search. It has been reported that PSOm has good performance in solving real-valued optimization problems.

In PSOm, each individual of the population is called a 'particle', which, in fact, represents a solution to a problem. Here a particle $\mathbf{x}_i$=[x_{i1}, x_{i2}, …x_{id},…, x_{iD}] in SCC is a vector composed of all the clustering centers and its dimension is D=d×K. Each particle has its own best position $\mathbf{p}_i$=[p_{i1}, p_{i2}, …, p_{id}, …, p_{iD}]. The index of the best particle among all the particles in the population is denoted by the g. Each particle

'flies' around in the multi-dimensional research space with a velocity $\mathbf{vel}_i$=[vel_{i1}, vel_{i2}, …vel_{id}, …, vel_{iD}]. This velocity is updated by the particle's own experience and the experience of the whole swarm

$$vel_{id}(t+1) = w(t) \times vel_{id}(t) + w_1 \times r_1 \times \left(p_{id}(t) - x_{id}(t) \right) + w_2 \times r_2 \times \left(p_{gd}(t) - x_{id}(t) \right), \quad (29)$$

where t is the current iteration number, w_1 and w_2 are acceleration coefficients set to 2, r_1 and r_2 are two independent random numbers uniformly distributed in the range of [0, 1]. In (29), the first term is the previous velocity of the particle, the second term represents the private thinking of the particle itself and the third term represents the collaboration among the particles. To obtain the good optimization result, the inertia factor w in (29) is defined to a linearly decreasing function of iterations:

$$w(t) = 1.4 - 0.4 \times t / I \quad (30)$$

where I is the maximum iteration number. As the w linearly decreases from a relatively large value 1.4 to a small value 0.4, PSOm tends to have more global search ability at the beginning of the iterations, and then have more local search ability near the end of the iterations. The experimental result demonstrates the effectiveness of this setting [27]. The position of each particle at each generation is updated by

$$x_{id}(t+1) = x_{id}(t) + vel_{id}(t+1). \quad (31)$$

The whole process of PSOm is summarized as follows:

Step 1: Set the number P of particles to 1000, the maximum number I of iterations to 500, the current iteration number t to 1 and the target value ε of objective function to 0;
initialize the particles with random positions and velocities.

Step 2: Evaluate the objective values of all particles according to (19), set $\mathbf{p}_i$ of each particle and its objective value equal to its current position and objective value, and set $\mathbf{p}_g$ and its objective value equal to the position and objective value of the best initial particle.

Step 3: Update the velocity and position of every particle by (29) and (31).

Step 4: Update w by (30) and increase the iteration number t.

Step 5: Evaluate the objective values of all particles in terms of (19).

Step 6: For each particle, compare its current objective value with the objective value of its $\mathbf{p}_i$. If the current value is smaller, then update $\mathbf{p}_i$ and its objective value with the current position and objective value.

Step 7: Determine the best particle of the current population with the best objective value. If the objective value is smaller than the objective value of $\mathbf{p}_g$, then update $\mathbf{p}_g$ and its objective value with the position and objective value of the current best particle.

Step 8: If the objective function of $\mathbf{p}_g$ is less than ε or t is larger than I, then output $\mathbf{p}_g$ and its objective value; otherwise go back to Step 3.

Fig. 2 Procedure of PSOm

In PSOm, there are three factors influencing its complexity: the iteration number I, the particle number P and the fitness function [28]. I and P are the user-specified parameters and in our experiment set to 1000 and 500, respectively. The fitness function is appointed the formula (19) and its complexity is $O(N\times L\times K)$ where N is the number of samples, K the number of clusters and L the number of classes. Consequently, the whole time complexity of SCC is $O(P\times I\times N\times L\times K)$.

3.2 Analyses of the proposed algorithm

When the simultaneous clustering and classification learning is achieved, we can further (1) compute the clustering and classification result for new samples; (2) mine some underlying information from the relation matrix **P** to help understand the structure of given data and the relationship between the structure and their classes.

3.2.1 Clustering and classification results for new samples

Generally speaking, for a new sample **x**, the clustering methods can give its cluster label to determine which natural sub-structure the **x** should belong to, but fail to provide any class result. On the other hand, the directly designed classifiers such as SVM and MLP can determine *only* its class label, but fail to determine which sub-class (i.e., clusters) the **x** should be categorized to. In contrast, our proposed framework SCC can not only provide the clustering result, but also give the classification result. Concretely, the posterior probability $p(c_j|\mathbf{x})$ for the **x** can be computed according to (14), thus its cluster label can be determined by $\arg\max_{1\le j\le K} p\left(c_j \mid \mathbf{x}\right)$; using the obtained $p(c_j|\mathbf{x})$ and the relation matrix **P**, its posterior probability $p(\omega_l|\mathbf{x})$ can be obtained by (12) and then its output class label is yielded by $\arg\max_{1\le l\le L} p\left(\omega_l \mid \mathbf{x}\right)$.

In SCC, for a new sample **x**, if $p(c_i|\mathbf{x})>p(c_j|\mathbf{x})$ for $i, j=1, 2, \ldots, K$ and $i\neq j$, then **x** is categorized to the ith cluster

$$\begin{aligned} & p\left(c_i \mid \mathbf{x}\right)-p\left(c_j \mid \mathbf{x}\right)>0 \\ \Rightarrow\ & dist\left(\mathbf{x}, \mathbf{v}_j\right)-dist\left(\mathbf{x}, \mathbf{v}_i\right)>0 \end{aligned} . \tag{32}$$

From (32), we can observe that (1) the cluster label of the **x** is determined by the distances to all the clustering centers; (2) the **x** will be finally categorized to the cluster corresponding to the nearest cluster center. According to Bayesian decision rule, if $p(\omega_i|\mathbf{x})>p(\omega_j|\mathbf{x})$ where $i, j=1, 2, \ldots, L$ and $i\neq j$, then **x** is assigned to the ith class. By using (12) and (14), we rewritten this inequality as

$$\begin{aligned} & p\left(\omega_i \mid \mathbf{x}\right)-p\left(\omega_j \mid \mathbf{x}\right)>0 \\ \Rightarrow\ & \sum_{k=1}^{K} p\left(c_k \mid \mathbf{x}\right)\left[p\left(\omega_i \mid c_k\right)-p\left(\omega_j \mid c_k\right)\right]>0 \\ \Rightarrow\ & \sum_{k=1}^{K} \frac{dist\left(\mathbf{x}, \mathbf{v}_k\right)^{-1}}{\sum_{r=1}^{K} dist\left(\mathbf{x}, \mathbf{v}_r\right)^{-1}}\left[p\left(\omega_i \mid c_k\right)-p\left(\omega_j \mid c_k\right)\right]>0 \end{aligned} . \tag{33}$$

For a given new sample **x**, $\sum_{r=1}^{K} dist\left(\mathbf{x}, \mathbf{v}_r\right)^{-1}$ is a fixed positive number, therefore the following inequality is resulted

$$\sum_{k=1}^{K} \frac{\left[p\left(\omega_i \mid c_k\right)-p\left(\omega_j \mid c_k\right)\right]}{dist\left(\mathbf{x}, \mathbf{v}_k\right)}>0 . \tag{34}$$

So we can conclude that the final classification decision for the **x** is dependent on the relation matrix **P** and its distance to all the clustering centers.

It is worth mentioning that the obtained cluster (or class) posterior probabilities imply the membership of the new sample to the clusters (or classes), thus indicating the reliability of the clustering (or classification) result.

3.2.2 Effect of relation matrix **P**

In SCC, a relation matrix **P** reveals the statistical relationship between clusters and classes. By analyzing the distribution characteristics of the elements in **P**, we capture some underlying structural information to help understand further both the structure of given data and the relationship between the structure and their classes.

For a given $K \times L$ relation matrix **P** where K is the number of the clusters and L that of the classes, its jth *row*-elements $[p(\omega_1|c_j), p(\omega_2|c_j), \ldots, p(\omega_L|c_j)]$ satisfying the constraint (17) can reflect the relationship between the jth cluster and all the classes, while its lth *column*-elements $[p(\omega_l|c_1), p(\omega_l|c_2), \ldots, p(\omega_l|c_K)]$ uncover the relationship between the lth class and all the clusters. From the row elements of **P**, let us gain some distribution information about the formed cluster. For the jth row corresponding to the jth cluster, if there is one and only one row element with the value of 1 and the others with 0, the cluster is pure and the samples falling into the cluster consistently belong to the same class; if the multiple non-zero elements exist in this row, the corresponding cluster is composed of the samples from multiple classes. At the same time, we can also capture some structural knowledge from the column-elements of **P**. Only one non-zero column-element implies that corresponding class contains only one cluster; while the multiple non-zero column elements implies that the samples in the corresponding class are scattered over multiple clusters. Consequently, the **P** plays an important role in the discovery and formulation of the structural knowledge in given dataset and thus makes SCC prone to be transparent.

To be clear, let us give an example. Given the 5×3 relation matrix **P**=[1, 0, 0; 0.94, 0.06, 0; 0, 1, 0; 0.09, 0.91, 0; 0, 0, 1] (in the Matlab format) obtained in the subsection 4.2, it represents the relationship between five clusters and three classes. For cluster c_1, c_3 and c_5 corresponding to the first, third and fifth rows of **P**, respectively, the 1s in these rows indicate that the three clusters are pure and belong to class ω_1, ω_2, and ω_3, respectively; for cluster c_2, the multiple non-zero values in the second row imply that this cluster is impure, and most samples in this cluster belong to class ω_1 and few samples belong to class ω_2; the similar analysis can be made for cluster c_4. At the same time, we can also make a similar analysis on the column-elements of **P**. From its first column [1; 0.94; 0; 0.09; 0], we find that class ω_1 consists of the three clusters,

specifically, almost all the samples in this class are scattered into clusters c_1 and c_2, and very few samples belong to cluster c_4; for class ω_2, a similar analysis can be given; for class ω_3, it corresponds to only one cluster, i.e., cluster c_5.

The **P** in SCC can reflect the statistical characteristics of the relationship between the classes and clusters. Therefore, we can easily judge the reliability of the obtained relationship from **P**. Taking the above **P** as an example, the first row [1, 0, 0] corresponding to the cluster c_1 implies a more reliable relationship than the second row [0.94, 0.06, 0]. However, since the relation matrix **R** in RFRC does not have the statistical characteristics, it fails to reflect any reliability of the relationship. For example, from **R**=[0, 0, 0.8; 0.1, 0.03, 0; 0.83, 0, 0; 0, 0.87, 0; 0.09, 0.78, 0.09] obtained by RFRC on the same dataset, we cannot draw a similar conclusion that the fourth row [0, 0.87, 0] can form a more reliable relationship than the fifth row [0.09, 0.78, 0.09].

4. Experimental results

In the following experiment, we choose the above-mentioned algorithms RBFNN, RFRC and VQ+LVQ3 as the competitors to demonstrate the effectiveness of SCC1 and SCC2. To make a fair comparison, we also especially design RBFNN_PSO (a variant of the RBFNN) as a competitor. In RBFNN_PSO, PSOm is also utilized to adjust the clustering centers with purpose to optimize the MSE criterion between the target and actual output.

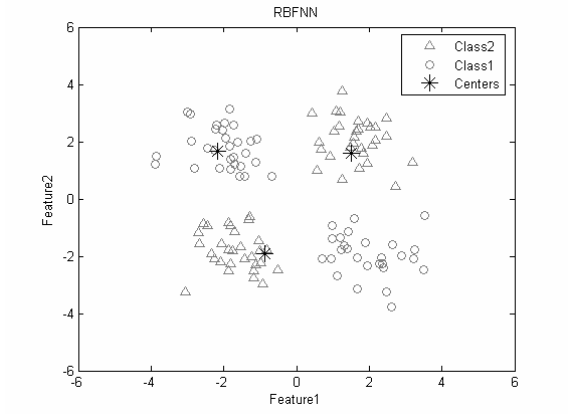
4.1 Performance test: robustness of clustering learning and classification learning

This experiment aims to evaluate both the clustering robustness and classification robustness among the above six algorithms when the training dataset contains outliers. A two-dimensional synthetic dataset1 for this experiment is generated with the statistics listed in Table 2. This dataset has two classes and four groups. The training dataset and test dataset follow the same distribution. In addition, three outliers (100, 100), (-100, -40) and (30, 200) are added into the training dataset to examine the robustness.

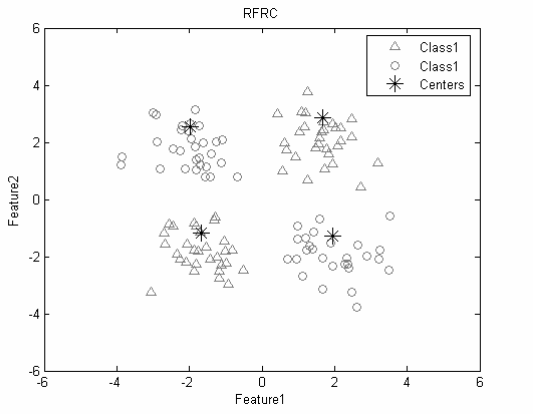
Table 2 Synthetic dataset1 with two Classes in four Groups

Group	Class label	Group center	Variance
Gaussian Distribution 1	ω_1	(-2, 2)	(0.5, 0.5)
Gaussian Distribution 2	ω_1	(2, -2)	(0.5, 0.5)
Gaussian Distribution 3	ω_2	(-2, -2)	(0.5, 0.5)
Gaussian Distribution 4	ω_2	(2, 2)	(0.5, 0.5)

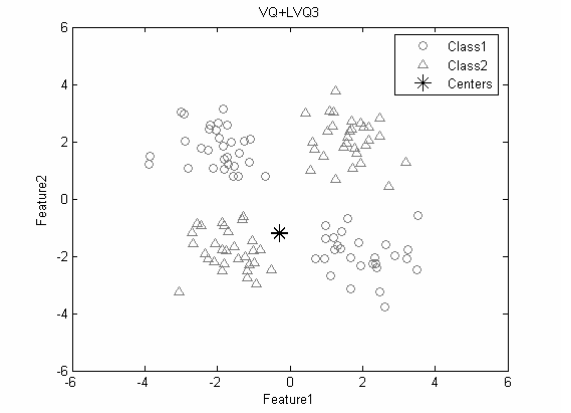
To show whether the above algorithms are influenced by the outliers, we compare the corresponding clustering results and classification results. The clustering centers obtained by the six algorithms on the training dataset are shown in Fig. 3. We can see from this figure that the clustering centers respectively obtained in RBFNN, VQ+LVQ3 and RBFNN_PSO fail to reflect the underlying groups in the training dataset; in contrast, the centers in RFRC, SCC1 and SCC2 relatively reflect the data distribution correctly.



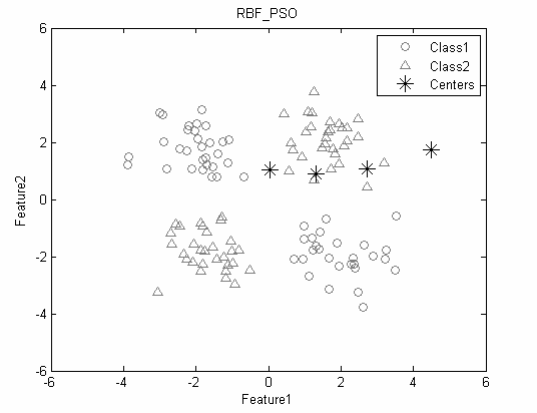
(a)



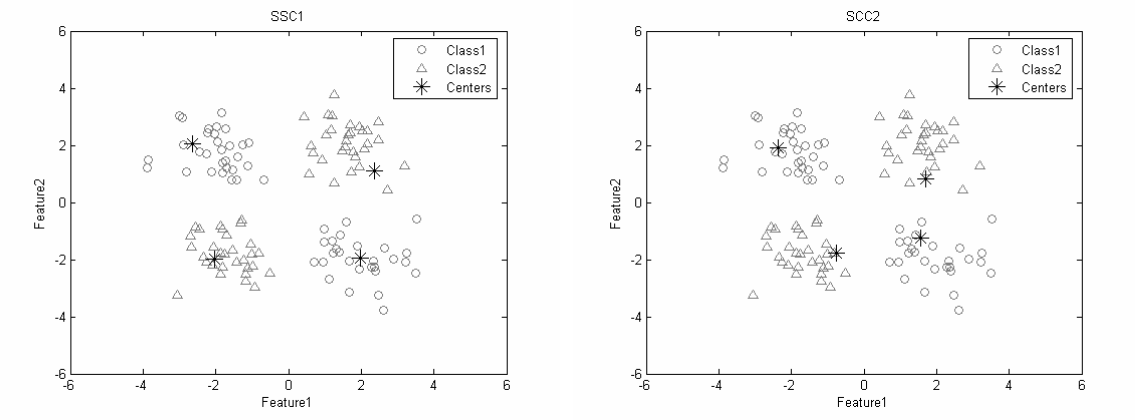
(b)



(c)



(d)



(e) (f)

Fig. 3 Comparison of the clustering centers '*' yielded by the six algorithms on the synthetic dataset1

Furthermore, Table 3 lists the clustering centers, the relation matrices **P**s (or connection weights) and the classification accuracies of the six algorithms, from which we can make the following analyses:

(1) In RBFNN, the resulted center $\mathbf{v}_4$ is heavily influenced by the outliers due to the non-robustness of FCM, thus leading to the low accuracy of 25.0%. In addition, the connection weights between the hidden layer and output layer fail to reflect any relationship between the clusters and classes.

(2) In RFRC, the resulted clustering centers are close to the real centers listed in Table 2, which can attribute to the robustness of KFCM. Such robustness makes the subsequently-formed relation matrix **R** logically correct, and hence RFRC achieves high accuracy of 98.3%. However, since the values of **R** are not from optimizing a criterion function, the classification accuracy is still lower than those of SCC1 and SCC2. Moreover, the values of the **R** lack the statistical property as well, and thus fail to exhibit the reliability of the obtained relationship. For example, from its second row [0.54, 0.01], we cannot judge whether 0.54 can represent reliably a relationship between cluster c_2 and class ω_1.

(3) In VQ+LVQ3, the clustering centers $\mathbf{v}_2$ and $\mathbf{v}_3$ are almost located in the same position, and $\mathbf{v}_1$ and $\mathbf{v}_4$ are far from the normal data distribution. Such result indicates that the formed centers are heavily influenced by the outliers and thus naturally fail to reveal the inherent structure of the dataset. As a result, the classifier based on the clustering results fails to correctly classify this dataset, and hence just achieves accuracy of 50% on the test dataset.

(4) In RBFNN_PSO, the classification accuracy achieves 98.3% which is much better than that of 25% in RBFNN. The underlying reason is that RBFNN_PSO utilizes PSOm to adjust the clustering centers so that it achieves the smaller value of the MSE criterion than RBFNN. Therefore, we can draw a

conclusion that RBFNN_PSO emphasizes the classification more than RBFNN. However, its clustering result still cannot capture the real structure of data. This phenomenon indicates that the clustering method in RBFNN_PSO is just employed as an aid to determine the parameters of the neural network rather than a method to explore the underlying structure of the input space.

(5) In SCC1 and SCC2, since the class information is utilized to guide the clustering process, the clustering result is relatively close to the real clustering centers listed in Table 2. This result implies that SCC1 and SCC2 can relatively resist the unfavorable effect of outliers effectively in the training dataset. Moreover, the relation matrices Ps in SCC1 and SCC2 indeed reveal the relationship between clusters and classes. In SCC1, the 1s in the first and second rows of **P** indicate that clusters c_1 and c_2 are pure and all belong to class ω_1; the third row [0.03, 0.97] and the fourth row [0.06, 0.94] mean that most of the samples falling into clusters c_3 and c_4 belong to class ω_2, and very few samples belong to class ω_1. According to the rules presented in subsection 3.2.2, we can also make a similar analysis about the **P** obtained by SCC2. Due to the robustness of both clustering result and so-generated relationship matrix **P**, both SCC1 and SCC2 achieve the highest classification accuracies of 99.2%.

Table 3 Clustering centers, relation matrix (weights) and accuracy of six algorithms, respectively

	RBFNN	RFRC	VQ+LVQ3	RBFNN_PSO	SCC1	SCC2
Cluster centers	$\mathbf{v}_1$=(-0.88 -1.93) $\mathbf{v}_2$=(1.52 1.59) $\mathbf{v}_3$=(-2.16 1.66) $\mathbf{v}_4$=(38.33 187.08)	(-1.97 2.54) (1.94 -1.26) (-1.67 -1.19) (1.67 2.85)	(-10.16 -18.87) (-0.28 -1.17) (-0.29 -1.21) (13.55 51.64)	(4.49 1.75) (1.30 0.89) (2.73 1.06) (0.04 1.04)	(-2.64 2.05) (1.98 -1.93) (-2.03 -1.99) (2.37 1.09)	(-2.35 1.91) (1.55 -1.24) (-0.76 -1.77) (1.70 0.82)
Relation matrix or weights	$\begin{bmatrix} -0.00 & 1.22 \\ 1.98 & -1.52 \\ -1.01 & 1.46 \\ -0.45 & 0.55 \end{bmatrix}$	$\begin{bmatrix} 0.80 & 0.00 \\ 0.54 & 0.01 \\ 0.00 & 0.64 \\ 0.00 & 0.83 \end{bmatrix}$	———	$10^4\times\begin{bmatrix} -1.98 & 1.98 \\ -8.13 & 8.12 \\ 7.07 & -7.07 \\ 3.03 & -3.03 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 \\ 1.00 & 0 \\ 0.03 & 0.97 \\ 0.06 & 0.94 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 \\ 1.00 & 0 \\ 0 & 1.00 \\ 0.03 & 0.97 \end{bmatrix}$
Accuracy	25.0%	98.3%	50%	98.3%	99.2%	99.2%

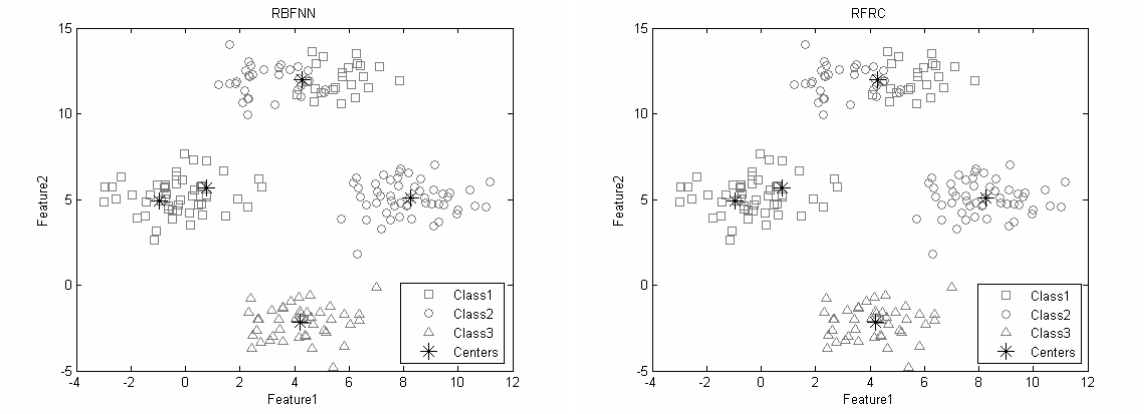
4.2 Performance test: effectiveness of clustering learning

To further examine the effectiveness of clustering result yielded by the above six algorithms, we design a two-dimensional synthetic dataset2 in terms of the statistics listed in Table 4.

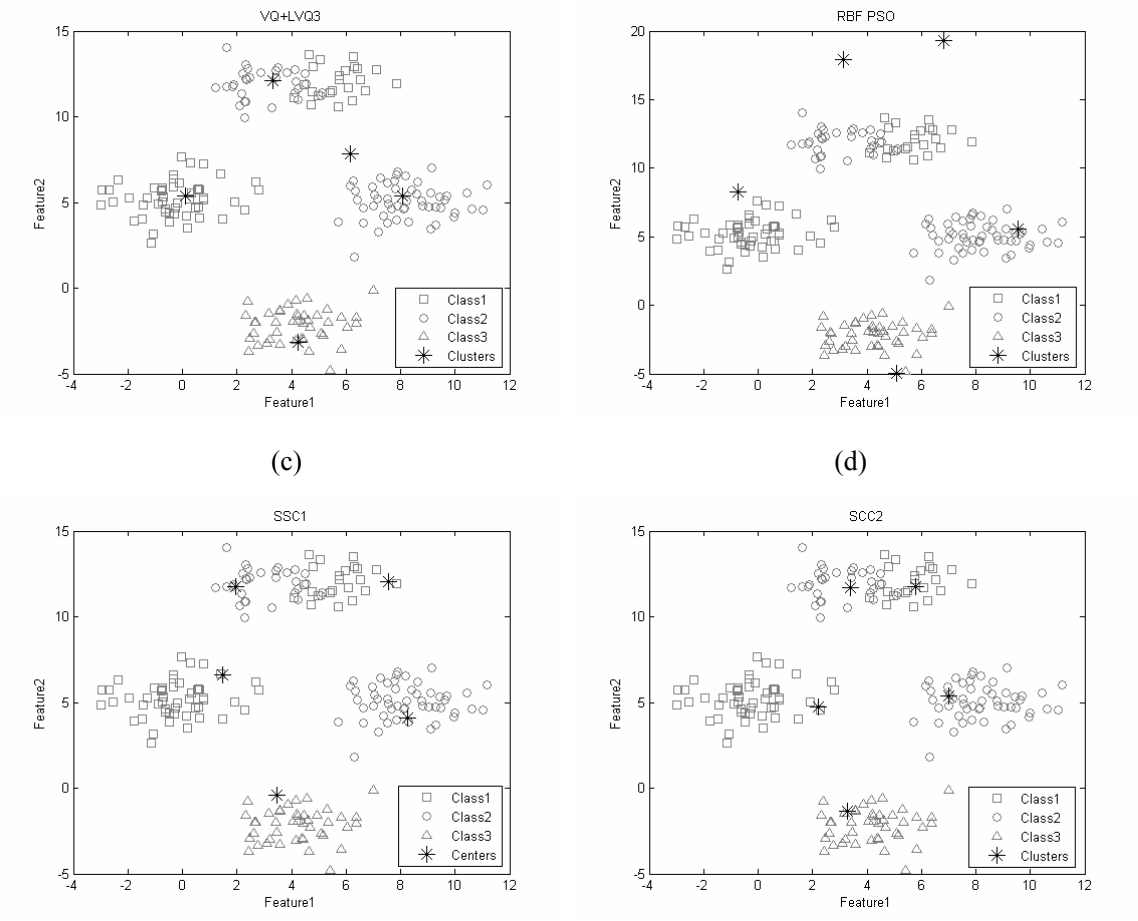
Table 4 Synthetic dataset2 with three classes in five Groups

Group	Class label	Group center	Variance
Gaussian Distribution 1	ω_1	(6, 12)	(1, 0.5)
Gaussian Distribution 2	ω_1	(0, 5)	(2, 1)
Gaussian Distribution 3	ω_2	(3, 12)	(2, 1)
Gaussian Distribution 4	ω_2	(8, 5)	(1, 0.5)
Gaussian Distribution 5	ω_3	(4, -2)	(2, 1)

To make a comparison among the six algorithms, we illustrate the obtained clustering centers on Fig. 4. It can be seen from these figures that in RBFNN and RFRC, Distributions 1 and 3 are merged into a big group. In VQ+LVQ3, the clustering centers $\mathbf{v}_2$, $\mathbf{v}_3$, $\mathbf{v}_4$ and $\mathbf{v}_5$ are relatively close to the real centers, but clustering center $\mathbf{v}_1$ is deviated from the original centers. In RBFNN_PSO, the clustering centers $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ are far away from the normal training samples. Therefore, these four algorithms obviously fail to discover the inherent structure of the dataset. In contrast, the clustering centers obtained by SCC1 and SCC2 reflect the structure in data correctly.



(a) (b)



(c) (d)

(e) (f)

Fig. 4 Comparison of the clustering centers '*' obtained by the six algorithms on the synthetic dataset2

Furthermore, from the clustering result and classification results shown in Table 5, we can make the following analyses to further understand the above the above algorithms:

(1) In RBFNN, since FCM is an unsupervised clustering method, Distributions 1 and 3 are merged into one single group characterized by the center $\mathbf{v}_1$=(4.28, 12.00). In fact, the samples respectively belonging to Distributions 1 and 3 are from different classes, and naturally should be categorized to two different groups. Therefore, these formed centers are almost incorrect and lead to the low accuracy of 83.5%.

(2) In RFRC, the clustering result is also almost incorrect still mainly due to the unsupervision of KFCM. Moreover, the first row [0.03 0.10 0.00] of relation matrix **R** indicate that the relationship between the obtained cluster c_1 and all the classes is unreliable. Due to the incorrect clustering centers and unreliable relationship, RFRC cannot make the correct classification decisions for the new samples falling into cluster c_1, and thus just achieves the accuracy of 79.5%.

(3) In VQ+LVQ3, since the clustering center $\mathbf{v}_1$ is not located in the right place, the classification based on

improper clustering centers just acquires the relatively low classification accuracy of 86.5%.

(4) In RBFNN_PSO, the accuracy of 97.5% is higher than that of 83.5% in RBFNN, this is because RBFNN_PSO utilizes the supervised information to adjust clustering centers in order to obtain the better classification result than RBFNN. However, the clustering centers in RBFNN_PSO still cannot uncover the structure in data, indicating that RBFNN_PSO fails to perform a valid supervised clustering in the input space. This result again demonstrates that RBFNN_PSO emphasizes the classification more than the clustering. In conclusion, although RBFNN_PSO achieves good classification performance, it lacks the ability of exploring the underlying structure in data.

(5) In SCC1 and SCC2, the obtained clustering centers are relatively close to the original clustering centers listed in Table 4 and the formed clustering regions represent an underlying data distribution of this dataset. Such effective clustering result can attribute to the utilization of the class information to guide the clustering process. The obtained matrices **P**s in SCC1 and SCC2 both indicate that the samples falling into clusters c_1 and c_2 are very likely to belong to class ω_1, the samples in clusters c_3 and c_4 are very likely to belong to class ω_2 and the samples in cluster c_5 are from class ω_3. Such relationship between clusters and classes accords with the real relationship listed in Table 4. Based on the real clustering centers and correct relationship matrix **P**, SCC1 and SCC2 achieve the good accuracies of 98.5%, respectively.

Table 5 Clustering centers, relation matrix (weights) and accuracy of six algorithms, respectively

	RBFNN	RFRC	VQ+LVQ3	RBFNN_PSO	SCC1	SCC2
Cluster centers	$\mathbf{v}_1$=(4.27 12.00) $\mathbf{v}_2$=(-0.95 4.93) $\mathbf{v}_3$=(0.78 5.68) $\mathbf{v}_4$=(8.27 5.11) $\mathbf{v}_5$=(4.20 -2.16)	(4.28 12.00) (-0.95 4.93) (0.77 5.67) (8.26 5.11) (4.19 -2.16)	(6.15 7.82) (0.11 5.39) (3.31 12.10) (8.07 5.38) (4.25 -3.19)	(3.12 17.95) (-0.73 8.27) (6.82 19.33) (9.56 5.55) (5.08 -4.97)	(7.57 12.05) (1.47 6.61) (1.95 11.75) (8.25 4.06) (3.47 -0.39)	(5.78 11.78) (2.21 4.75) (3.40 11.71) (7.01 5.40) (3.29 -1.34)
Relation matrix	$\begin{bmatrix} 1.33 & -4.26 & -0.63 \\ 0.11 & 0.71 & 0.36 \\ 0.87 & 0.35 & -0.42 \\ -1.50 & 4.86 & 0.15 \\ -0.30 & -0.31 & 1.33 \end{bmatrix}$	$\begin{bmatrix} 0.03 & 0.10 & 0.00 \\ 0.87 & 0.00 & 0.00 \\ 0.78 & 0.09 & 0.09 \\ 0.00 & 0.83 & 0.00 \\ 0.00 & 0.00 & 0.80 \end{bmatrix}$	———	$\begin{bmatrix} 1.31 & -0.56 & -0.24 \\ 6.41 & -7.06 & 0.31 \\ -0.73 & 1.74 & -0.18 \\ -7.64 & 8.82 & -0.03 \\ -0.31 & -0.12 & 1.24 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.85 & 0.15 & 0 \\ 0 & 1.00 & 0 \\ 0.10 & 0.90 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.94 & 0.06 & 0 \\ 0 & 1.00 & 0 \\ 0.09 & 0.91 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$
Accuracy	83.5%	79.5%	86.5%	97.5%	98.5%	98.5%

4.3 Performance Test: Effectiveness of Classification Learning

To further investigate the effectiveness of SCC on real-life datasets, we use 20 datasets cited from the

UCI Machine Learning Repository [29] which is a repository of databases, domain theories and data generators collected by the machine learning community for the empirical analysis of machine learning algorithms.

In this experiment, the comparison is made among RFRC, VQ+LVQ3, RBFNN, RBFNN_PSO, SVM, SCC1 and SCC2. In these algorithms except SVM, the cluster number K is sought in the range from the number of classes up to $c_{\max}$. Here the parameter $c_{\max}$ is set to $\sqrt{N}$ in terms of Bezdek's suggestion [30] where N is the number of the training samples. In RFRC, RBFNN, RBFNN_PSO, SVM and SCC2, the RBF kernel is adopted and its scale factor λ is determined by searching in {0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15}. In SCC1 and SCC2, the parameter β is selected from {0.01, 0.1, 1}. In SVM, the regularization parameter C is determined from $\{2^{-1}, 2^{0}, 2^{3}, 2^{5}, 2^{7}, 2^{9}\}$. Due to the multiple parameters existing in these algorithms, the Discrete Grid Search [31] which is a minimization procedure based on exhaustive search in a limited range is adopted to acquire the optimal values with trial-and-error approach [24] on the training dataset.

In all of our experiments, each dataset is randomly partitioned into two halves: one half is used for training and the other for testing. This process runs repeatedly and independently 10 times, and their averaged accuracy and the corresponding standard deviation are reported in Table 6.

Table 6 Comparison of classification accuracies on benchmark datasets

Dataset (#samples×#dim×#class)	RFRC	VQ+LVQ3	RBFNN	RBFNN_PSO	SVM	SCC1	SCC2
WBCD (683×9×2)	97.0 ± 0.6	96.8 ± 0.6	96.8 ± 0.5	96.9 ± 0.9	96.9 ± 0.5	96.8 ± 0.6	97.0 ± 0.4
Water (116×38×2)	97.9 ± 1.3	98.4 ± 1.2	98.3 ± 1.0	98.4 ± 1.4	98.5 ± 0.8	98.3 ± 1.5	98.4 ± 1.2
Thyroid (215×5×3)	91.8 ± 2.0	92.7 ± 2.2	95.3 ± 1.0	95.5 ± 1.7	95.2 ± 1.5	**96.3 ± 1.5**	**96.4 ± 1.5**
Lung_cancer (32×56×3)	40.6 ± 11.3	42.5 ± 10.8	43.8 ± 15.8	43.8 ± 9.5	41.9 ± 8.4	**48.3 ± 13.3**	**48.3 ± 14.2**
Pid (768×8×2)	69.6 ± 2.8	72.1 ± 2.0	74.6 ± 2.5	75.1 ± 2.4	76.4 ± 1.7	74.4 ± 2.3	**77.0 ± 0.3**
Soybean_small (47×35×4)	**99.1 ± 1.7**	96.1 ± 10.4	98.1 ± 1.7	98.3 ± 2.2	98.3 ± 3.5	96.5 ± 3.3	**99.6 ± 1.3**
WDBC (569×30×2)	92.0 ± 1.6	96.4 ± 0.9	95.0 ± 1.2	96.4 ± 1.0	**97.2 ± 0.7**	96.6 ± 0.9	**96.9 ± 0.7**
Ionosphere (351×34×2)	90.9 ± 1.7	85.4 ± 1.8	88.4 ± 1.6	89.0 ± 1.6	**93.0 ± 0.9**	92.1 ± 1.5	**93.2 ± 1.4**
Waveform (5000×21×3)	83.0 ± 0.5	85.1 ± 0.6	**86.5 ± 0.9**	**86.6 ± 0.7**	**86.2 ± 0.4**	82.9 ± 2.4	**86.3 ± 0.6**
Balance_scale (625×4×3)	84.7 ± 1.5	86.0 ± 1.8	**90.5 ± 1.0**	**90.8 ± 1.1**	**90.5 ± 1.0**	89.4 ± 1.6	**90.6 ± 1.3**
Heart_disease (270×13×2)	80.9 ± 2.2	81.4 ± 1.8	82.5 ± 2.3	**83.0 ± 3.2**	**83.3 ± 2.2**	82.7 ± 1.9	**83.0 ± 2.1**
Pima_Indian_diabetes(768×8×2)	70.7 ± 3.2	72.6 ± 2.0	74.2 ± 2.3	**76.2 ± 1.6**	**76.3 ± 2.0**	75.2 ± 1.6	**76.0 ± 1.4**
Glass (214×9×6)	63.8 ± 3.8	63.2 ± 3.6	65.0 ± 3.8	65.4 ± 3.5	**68.5 ± 3.5**	65.1 ± 3.6	64.9 ± 2.5
Sonar (208×60×2)	77.5 ± 3.9	73.9 ± 2.8	80.2 ± 3.0	80.5 ± 4.3	**85.4 ± 3.3**	81.7 ± 4.5	80.8 ± 5.1
Wine (178×13×3)	96.0 ± 1.7	96.5 ± 1.5	97.3 ± 1.1	**98.2 ± 1.3**	**98.4 ± 1.1**	96.9 ± 1.5	97.1 ± 1.8
Spambase (4601×57×2)	85.1 ± 1.1	88.5 ± 0.7	80.7 ± 1.0	**90.8 ± 0.5**	89.2 ± 0.6	78.5 ± 7.7	88.1 ± 1.3

Ecoli (336×7×8)	81.8 ± 3.3	78.8 ± 3.0	85.2 ± 2.7	**86.2 ± 2.7**	85.0 ± 1.7	82.9 ± 3.7	83.7 ± 1.8
Lenses (24×4×3)	71.7 ± 7.6	74.2 ± 11.5	75.8 ± 14.6	**80.8 ± 9.9**	75.1 ± 10.4	77.5 ± 9.9	77.5 ± 3.7
Iris (150×4×3)	95.3 ± 1.1	94.7 ± 1.9	**96.4 ± 1.6**	**96.5 ± 1.5**	95.9 ± 15	94.9 ± 1.0	95.2 ± 1.4
Bupa (345×6×2)	61.0 ± 2.4	62.1 ± 3.7	**70.8 ± 3.6**	**70.9 ± 4.3**	66.7 ± 7.5	64.2 ± 3.0	67.5 ± 5.8

First, we compare the classification results yielded by SCC1 and SCC2. It can be seen from Table 6 that the accuracies of SCC2 are respectively better than those of SCC1 except for the dataset *Sonar*. Such a performance promotion of SCC2 attributes to the adoption of the kernel-induced distance metric, which makes SCC2 more appropriate for the datasets with non-spherical distributions.

Second, we make the comparison among SCC1, SCC2, RFRC and VQ+LVQ3. SCC1 achieves the comparable or better performance than RFRC (VQ+LVQ3) on 17 (18) datasets except for *Soybean_small, Spambase* and *Iris* (*waveform* and *Spambase*). SCC2 overall achieves better performance than VQ+LVQ3 and RFRC on all the datasets. The relatively good classification performances of SCC1 and SCC2 come from the effective classification learning mechanism of the proposed framework SCC.

Third, we compare RBFNN, SCC1 and SCC2. SCC1 obtains comparable or better performance on 12 datasets and worse performance on 8 datasets than RBFNN, while SCC2 performs comparably or better than RBFNN on 17 datasets. In conclusion, the classification ability of RBFNN is comparable to that of SCC1, but worse than that of SCC2.

Fourth, let us compare RBFNN_PSO, SCC1 and SCC2. RBFNN_PSO performs consistently better than RBFNN on all the datasets, indicating that RBFNN_PSO is more capable than RBFNN. Compared to RBFNN_PSO, SCC1 yields worse performance on 11 datasets. The possible reason is that the Euclidian distance metric makes SCC1 unsuitable for the non-spherical distribution, thus heavily limits its performance. It is also observed that compared to RBFNN_PSO, SCC2 produces better classification performance on 5 datasets, comparable performance on 9 datasets, but worse performance on the other 6 datasets. It is worth pointing out that the comparison here is in fact not quite favorable for our algorithm SCC2. Because in fact, RBFNN_PSO just aims to achieve good classification regardless of revelation of data distribution; in contrast, SCC2 aims to make a balance between the classification performance and the clustering performance. However, we still need to point out that even so, on 14 out of all the 20 datasets, SCC2 still achieves better than or comparable classification performance to RBFNN_PSO. Moreover, one disadvantage of RBFNN_PSO is that it usually sacrifices the structure of clustering for the good performance in classification, thus fails to retain a valid result in the clustering phase. This phenomenon

has been empirically demonstrated by the experimental results in the subsection 4.1 and 4.2. In contrast, SCC concerns clustering performance as well as classification performance, which is also visually proved in the same experiments.

Finally, to give a baseline reference, we make a comparison against the state-of-the-art classifier SVM. Due to not using the kernel trick, SCC1 works worse than SVM on most of the datasets. In contrast, SCC2 gains better accuracies than SVM on 5 datasets, comparable accuracies on 13 datasets and worse accuracies only on 2 datasets *Sonar* and *Glass*, indicating that SCC2 is highly competitive in terms of classification accuracy. However, SCC-type algorithms possesses the following advantages compared to SVM: (1) both the effective and robust classification result and the clustering result can be obtained simultaneously; (2) from the obtained relation matrix **P,** we can gain some insight into the structure of given data and the relation between the structure and their classes; (3) the class posterior probabilities computed in this framework can reflect the confidence of the classification decision, which is important for reliable and interpretable classification.

5. Conclusions

In this paper, a new simultaneous learning framework for clustering and classification (SCC) is presented to fuse the advantages of classification learning and clustering learning into single framework. In SCC, the Bayesian theory is employed to model the *cluster posterior probabilities of classes* which represent the correspondences between clusters and classes. By using such posterior probabilities, the objective function depending only on the clustering centers is designed to evaluate not only the classification ability but also the clustering ability. By *only* optimizing the clustering centers in the objective function, both the classification learning and clustering learning can be realized simultaneously. Here we use an evolutionary technique called modified particle swarm optimization (PSOm) to find the optimal clustering centers. The experimental results on 2 synthetic datasets and 20 real-life datasets demonstrate that SCC can: (1) robustly cluster the data with guidance of supervision information to form relatively pure clusters; (2) adaptively reveal an underlying relationship between clusters and classes; (3) design an effective and transparent classification mechanism.

In this paper, we also show that the framework of SCC can easily be extended by using different clustering methods. As an example, under the guidance of GMM clustering method, we can adopt the formula (13) instead of (14) and then optimize the corresponding parameters of Gaussian model

embedding in the objective function of SCC. As a result, a novel algorithm which can simultaneously realize the clustering learning and classification learning can be derived from SCC.

It is worth mentioning that extending SCC to the semi-supervised case is not so straightforward because when the training data is partially labeled, the relation matrix P can not be directly derived and calculated by the formula (16). One of future works is to develop SCC in the semi-supervised scenario. Moreover, the adaptive determination for the number K of the prototypes and the kernel parameter also deserve a further study.

Acknowledgement:

The authors would like to thank the anonymous referees for their helpful comments and suggestions to improve the presentation of this paper. We thank Natural Science Foundation of Jiangsu Province under Grant No. BK2006521, National Science Foundation of China under Grant No. 60505004, 60603029, 60773061 and 60873176, respectively.

References

[1] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-22 (1) (2000) 4-37.

[2] A.K. Jain, M.N. Murty, and P. J. Flynn, Data clustering: a review. ACM Computing Surveys, 31(3) (1999) 264-323.

[3] S. Haykin, Neural Networks: A comprehensive foundation, New Jersey: Prentice Hall, 1999.

[4] V.N. Vapnik, The nature of statistical learning theory. New York: Springer, 1999.

[5] C. Burges, A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2 (1998) 121-167.

[6] M.T. Musavi, W. Ahmed, K.H. Chan, K.B. Faris, D.M. Hummels, On the training of radial basis function classifiers, Neural Network, 5 (4) (1992) 595-603.

[7] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum, 1981.

[8] M. Setnes, R. Babuška, Fuzzy Relational Classifier Trained by Fuzzy Clustering, IEEE Trans. SMC Part B, 29 (1999) 619-625.

[9] W.L. Cai, S.C. Chen, D.Q. Zhang, Robust fuzzy relational classifier incorporating the soft class labels,

Pattern Recognition Letters, 28 (2007) 2250-2263.

[10] D.Q. Zhang, S.C. Chen, A novel kernelized fuzzy c-means algorithm with application in medical image segmentation, Artificial Intelligence in Medicine, 32 (1) (2004) 37-50.

[11] S.C. Chen and D.Q. Zhang, Robust Image Segmentation Using FCM With Spatial Constraints Based on New Kernel-Induced Distance Measure, IEEE Trans. SMC Part B, 34 (4) (2004) 1907-1916.

[12] W. Pedrycz, G. Vukovich, Fuzzy clustering with supervision, Pattern Recognition, 37 (2004) 1229-1349.

[13] S.W. Kim., B.J. Oommen., Enhancing prototype reduction schemes with LVQ3-type algorithms, Pattern Recognition, 36 (2003) 1083-1093.

[14] N. Ye and X. Li, A supervised, incremental learning algorithm for classification problems, Comput. Ind. Eng. J., 43 (4) (2002) 677-692.

[15] X. Li and N. Ye, Grid- and dummy-cluster-based learning of normal and intrusive clusters for computer intrusion detection, Qual. Reliab. Eng. Int., 18 (3) (2002) 231-242.

[16] X. Li and N. Ye, A supervised clustering and classification algorithm for mining data with mixed variables, IEEE Trans. SMC Part A, 36 (2) (2006) 396-406.

[17] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, A. Smola, Input space vs. feature space in kernel-based methods, IEEE Trans. Neural Networks, 10 (5) (1999) 1000-1017.

[18] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities in pattern recognition, IEEE Trans. Electronic Computers, 14 (1965) 326-334.

[19] R. J. Hathaway, J. C. Bezdek, Generalized fuzzy c-means clustering strategies using L_p norm distance, IEEE Trans. Fuzzy Syst., 8 (2000) 576-572.

[20] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[21] W. Pedrycz, Reasoning by analogy in fuzzy controllers, fuzzy control systems, Kandel and Langholz, Eds. Boca Ratom, FL: CRC, pp: 55-74, 1994.

[22] G.J. Klir, B. Youan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[23] G. McLachlan, Finite Mixture Models. New York: Wiley, 2000.

[24] S. Abe, Training of Support Vector Machines with Mahalanobis Kernels, International Conference on Artificial Networks, Lecture Notes in Computer Science, 3697 (2005) 571-576.

[25] Y. Rahmat-Samii, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in Engineering

Eelectromagnetics, 17th International Conference on Applied Electromagnetics and Communications, pp: 1-5, 2003.

[26] J. Kennedy, R.C. Eberhart, Particle Swarm Optimization, IEEE International Conference on Neural Networks, pp: 1942-1948, 1995.

[27] Y. Shi, R.C. Eberhart, A modified Particle swarm optimizer, IEEE International Conference on Evolutionary Computation, 1998.

[28] M. Zubair, M. Choudhry, A. Malik, I. Qureshi, Particle Swarm Optimization Assisted Multiuser Detection along with Radial Basis Function, IEICE Transactions on Communications, E90-B (7) (2007) 1861-1863.

[29] C. Blake, E. Keogh, C.J. Merz,: UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA, 1998.

[30] J.C. Bezdek, Pattern Recognition in handbook of Fuzzy computation. IOP Publishing Ltd., Boston, Ny, (Chapter 6), 1998.

[31] Á. B. Jiménez, J. L. Lázaro and J. R. Dorronsoro, Finding Optimal Model Parameters by Discrete Grid Search, Advances in Soft Computing, 44 (2008) 120-127.