A Multi-objective Simultaneous Learning Framework for Clustering and Classification

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

[1] (*Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, PR China*)

[2] (*Computer School, Nanjing Normal University, Nanjing 210097, PR China*)

Abstract: Traditional pattern recognition involves two tasks: clustering learning and classification learning. Clustering result can enhance the generalization ability of classification learning, while the class information can improve the accuracy of clustering learning. Hence, both learning methods can complement each other. To fuse the advantages of both learning methods together, many existing algorithms have been developed in a *sequential fusing way by* first optimizing the clustering criterion and then the classification criterion associated with the obtained clustering results. However, such kind of algorithms naturally fails to achieve the simultaneous optimality for two criteria, and thus have to sacrifice either the clustering performance or the classification performance. To overcome that problem, in this paper, we present a multi-objective simultaneous learning framework (named MSCC) for both clustering and classification learning. MSCC utilizes multiple objective functions to formulate the clustering and classification problems, respectively, and more importantly it employs the Bayesian theory to make these functions all *only* dependent on a set of the same parameters, i.e., clustering centers which play a role of the bridge connecting the clustering and classification learning. By simultaneously optimizing the clustering centers embedded in these functions, not only the effective clustering performance but also the promising classification performance can be simultaneously attained. Furthermore, from the multiple Pareto-optimality solutions obtained in MSCC, we can get an interesting observation that there is complementarity to great extent between clustering and classification learning processes. Empirical results on both synthetic and real data sets demonstrate the effectiveness and potential of MSCC.

Key words: Pattern recognition; Clustering learning; Classification learning; Bayesian theory; Multi-objective optimization.

[*] Corresponding author: Tel: +86-25-84896481-12221, Fax: +86-25-84892400. Email: s.chen@nuaa.edu.cn (S. C. Chen) caiwl@nuaa.edu.cn (W. L. Cai)

1. Introduction

Traditional pattern recognition involves two tasks [1]: clustering learning and classification learning. In the case of clustering learning, the problem is to group the given samples into meaningful clusters based on similarity [2]. The formed clusters are appropriate for the exploration of the underlying structure in data and the better understanding for the nature of the data. In the case of classification learning, the problem is to construct the discriminant function for distinguishing the samples with different class labels [3]. The discriminant function can provide class labels for the newly encountered samples.

It has been proven that the clustering results or structures in data can help enhance the generalization ability of classification learning [4], and thus exploiting as much prior knowledge (including structure in data) as possible about given problem to boost the generalization performance of a classifier is consistent with the famous No Free Lunch (NFL) theorem [3]. Our experimental results (refer to Section 4 for more details) also give a positive validation on the above assertion. On the other hand, the class information can also help improve performance of clustering learning. E.g., by utilizing the class information to guide the clustering process, some supervised clustering [5-7] or semi-supervised clustering algorithms [8-10] have been developed. The corresponding empirical results all demonstrated that the class information can significantly improve the effectiveness of the clustering results. Hence, we have reason to believe that the clustering and classification learning can complement to each other.

Generally, clustering and classification learnings are usually formulated by different models or criteria, hence it is relatively difficult to cast both into a single framework. To fuse the advantages of both learners together, many existing algorithms [11-20] handle the clustering learning and classification learning in a *sequential* or *independent* manner. As illustrated in Fig. 1, these algorithms firstly utilize the clustering criterion to optimize the clustering process so that the structures in data can be explicitly revealed. Then, based on the obtained clustering result, these algorithms optimize the classification criterion associated with the obtained structural information to give the class label for new samples. Such kind of algorithms *sequentially* optimizes the clustering criterion and the classification criterion, and thus fails to achieve the simultaneous optimality for such two criteria. Recently, we have gone a small step ahead in this research and proposed a simultaneous learning algorithm for clustering and classification (named SCC) [21]. In SCC, the classification criterion and clustering criterion are combined to a ***single*** objective function by a trade-off parameter, whose goal is to compromise the classification and the clustering performances, but its value in optimizing the objective is generally hard to be optimally chosen except for an exhaustive search in some range, which is a heavier learning burden. In fact, the all above mentioned algorithms usually have to sacrifice the clustering performance for the classification performance, or vice versa. As a result, it is not easy for them to achieve an effective clustering and classification performance at one time.

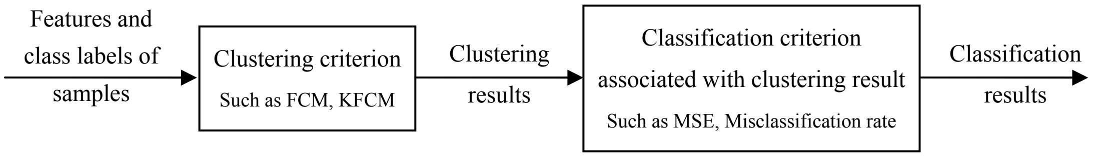

Features and class labels of samples
Clustering criterion
Such as FCM, KFCM
Clustering results
Classification criterion associated with clustering result
Such as MSE, Misclassification rate
Classification results

Fig. 1 Sequential optimization for the clustering and classification criteria

To overcome this defect, in this paper, we present a multi-objective simultaneous learning framework (named MSCC) for both clustering and classification learning. As shown in Fig. 2, we utilize the multiple functions to formulate the clustering and classification problems to realize the joint learning in MSCC. More importantly, we employ the Bayesian theory to bridge a connection between them and make all these functions *only* dependent on the same set of the parameters, i.e., the clustering centers. In all of our experiments, we just utilize the following two objective functions, i.e., the misclassification rate and the intra-cluster compactness in the feature space to evaluate the classification and clustering performances, respectively. Since the clustering and classification learnings seek different goals, thus generally speaking, the objective function established just for classification focuses on more classifier' generalization and less discovering inherent structures in data; conversely, the objective function established just for the clustering learning concerns more discovering structures in data and less classification performance. Consequently, the result obtained by optimizing the classification objective function alone is usually more likely inconsistent with that obtained by optimizing the clustering objective function alone. However, this does not imply that the two objectives can neither form a compromise nor be more prone to consistent for their performance improvement. This is our starting point of using multi-objective optimization technique to achieve simultaneous optimality for both. To this end, concretely, we adopt the multi-objective particle swarm optimization (MOPSO) [22] to simultaneously optimize the clustering centers embedded in these two functions, as a result, by such optimization, we can intuitively obtain a consistent result between the clustering and classification. In the corresponding experiment, an interesting observation is that those clustering centers which yield relatively low values of the objectives jointly for both clustering compactness and classification error rate on the training dataset can empirically result in the best clustering or classification result on the corresponding test data. This phenomenon again demonstrates the consistency or complementarity between the clustering and classification learnings, that is, the optimization of clustering criterion is beneficial to classification, or vice versa. The subsequent more experimental results on both synthetic and real-life datasets all demonstrate also the effectiveness and potential of MSCC.

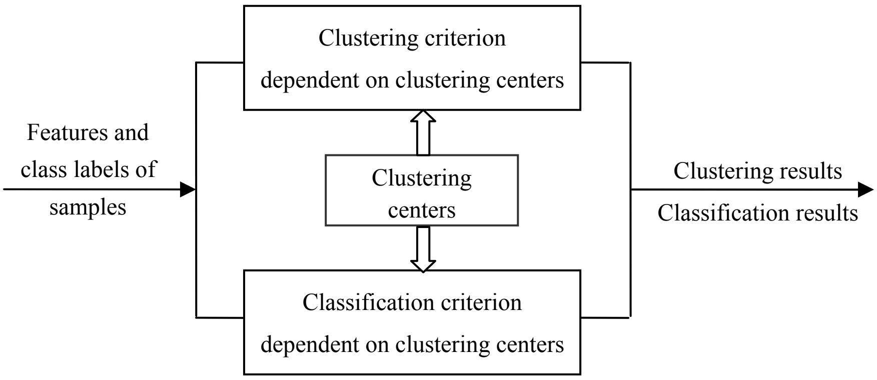


Fig. 2 Simultaneous optimization for the clustering and classification criteria

The outline of the rest of the paper is as follows. In section 2, we discuss the related work. In section 3, we present the main ideas of the MSCC algorithm. The experimental results are provided in section 4. We conclude in Section 5.

2. Related work

There have been several recent related works to inherit the merits of both clustering and classification learning. We will review the main works as follows.

Radial Basis Function neural network (RBFNN) [12, 13], as shown in Fig. 3, is a feed-forward multi-layer network. It usually consists of three layers: input layer, hidden layer and output layer. Each basis function Φ_k corresponds to a hidden unit and w_{kl} represents the weight from the kth basis function or hidden unit to the lth output units. In the training phase, RBFNN first executes unsupervised clustering process to determine the parameters of the basis function Φ_k under the guidance of fuzzy c-means (FCM) clustering criterion [13]. Next, it uses the mean squared error (MSE) classification criterion between the target and actual outputs to optimize the connection weights w_{kl} between the hidden and output layers. In

RBFNN, the clustering method can ensure the good classification generalization. However, such clustering method is just an aid in determining the parameters of the neural network, rather than a method to reveal the inherent structure in data. In fact, RBFNN can not really inherit the advantages of both clustering learning and classification learning in a *single* algorithm. In addition, another defect of RBFNN is that the connecting weights w_{kl} conceal the learned knowledge, which leads to the poor transparency and interpretability for knowledge (representation).

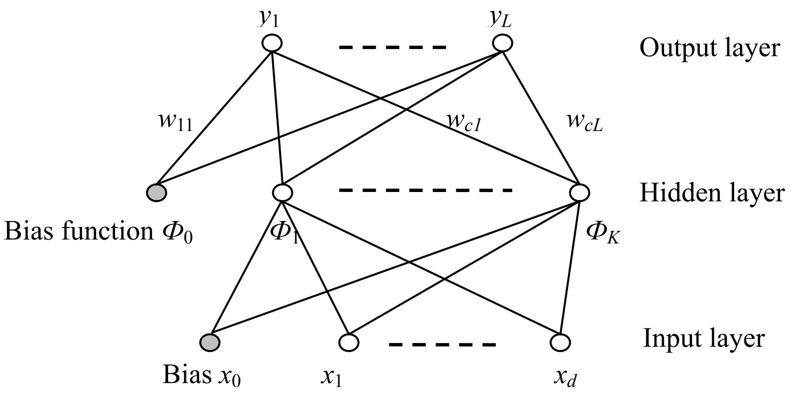


Fig. 3 Architecture of RBFNN

Setnes et al. proposed **Fuzzy Relational Classifier (FRC)** [14] to provide a transparent alternative to the black-box techniques such as neural networks. Its training process also involves two main steps which are illustrated in Fig. 4. In the first step, it adopts the FCM clustering criterion to discover the natural structure in data. In the second step, by using the obtained fuzzy partition and the given hard class labels (i.e., the samples from the same class share a common class label), it computes a relation matrix **R** under the implicit classification criterion to reflect the relationship between clusters and classes.

Lately, in our previous work, we have presented **Robust FRC (RFRC)** [20] with the aim of enhancing the robustness of FRC. According to the two-step training way of FRC, its robustness is improved from the following two sources: first, use the robust Kernelized FCM (KFCM) [23] to replace FCM; second, employ the soft class label motivated by the fuzzy *k*-nearest-neighbor [24] to replace the hard class label. This way, with incorporation of both KFCM and the soft class labels, RFRC makes the constructed relation matrix **R** more really reflect the relationship between the classes and clusters for the subsequent classification, and thus significantly boosts the robustness and accuracy of FRC.

FRC and RFRC fuse the merits of clustering and classification learning to some extent, but such *sequential* optimization can not be guaranteed to obtain satisfactory clustering and classification results simultaneously. In addition, the entries in the relation matrix **R** lack the statistical meaning, thus it is difficult to judge whether the obtained relationship is really reliable.

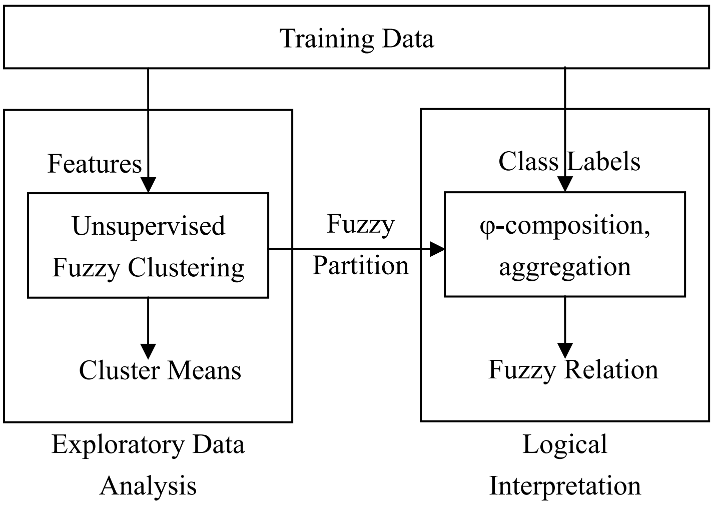


Fig. 4 Training process of FRC and RFRC

Likewise, Kim and Oommen [18] proposed an algorithm called VQ+LVQ3. It first utilizes Learning Vector Quantization (LVQ) to optimize both the positions and class labels of the cluster centers, and then applies 1NN classifier to perform classification on the top of the obtained centers. Actually, LVQ3 is a supervised clustering in which the class information is used to guide clustering. Similar to VQ+LVQ3, a supervised clustering and classification algorithm named CCAS [11, 25] and its extended version ECCAS [26] also fall into such a two-step framework. Since both 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) have no a true design phase. Their common idea is to seek a set of good prototypes as class representatives for subsequent classification using the 1NN classifier.

To sum up, all above methods first optimize the clustering criterion, and then the classification criterion associated with the clustering result, i.e. they adopt a two-step learning paradigm which fails to realize the simultaneous optimization for both criteria. This may limit the strength of both clustering and classification.

3. The proposed method

To obtain the satisfactory clustering and classification result and inspired by our previous work [21], we present a multi-objective simultaneous learning framework (named MSCC) for both clustering and classification learning. In its implement, we first employ the Bayesian theory to bridge the connection between both and make all their objectives *only* dependent on the same set of the cluster centers as the parameters to be optimized. Next, we utilize the multi-objective framework to formulate the clustering and classification problems. Finally, we adopt MOPSO to simultaneously optimize the clustering centers embedded in these functions.

3.1 Clustering mechanism and classification mechanism

To realize the simultaneous clustering and classification in MSCC, one key is to make the clustering and classification results all *only* dependent on the same parameters.

In the clustering learning, by using the fuzzy c-means clustering as reference, the clustering membership u_{ik} of the training sample $\mathbf{x}_i$ to the kth cluster can be computed

$$u_{ik} = \frac{dist\left(\mathbf{x}_i, \mathbf{v}_k\right)^{-1}}{\sum_{r=1}^{K} dist\left(\mathbf{x}_i, \mathbf{v}_r\right)^{-1}} \tag{1}$$

where *dist* represents the distance between the samples and the centers. When the clustering centers are determined, the clustering mechanism can be established.

Next, we will employ the Bayesian theory to design a classification mechanism only relying on $\{\mathbf{v}_k\}$. In the classification learning, when the posterior probabilities $p(\omega_l|\mathbf{x}_i)$ can be modeled, the output class label $f(\mathbf{x}_i)$ can be determined

$$f\left(\mathbf{x}_i\right) = \underset{1\le l\le L}{\arg\max}\, p\left(\omega_l \mid \mathbf{x}_i\right) \tag{2}$$

To introduce the cluster information into $p(\omega_l|\mathbf{x}_i)$, we resort to the formed clusters $\{c_k\}$ to reformulate $p(\omega_l|\mathbf{x}_i)$ through the total probability theorem as

$$\begin{aligned} p\left(\omega_l \mid \mathbf{x}_i\right) &= \sum_{k=1}^{K} p\left(\omega_l, c_k \mid \mathbf{x}_i\right) \\ &= \sum_{k=1}^{K} p\left(c_k \mid \mathbf{x}_i\right) p\left(\omega_l \mid c_k, \mathbf{x}_i\right) \\ &= \sum_{k=1}^{K} p\left(c_k \mid \mathbf{x}_i\right) p\left(\omega_l \mid c_k\right) \end{aligned} \tag{3}$$

where ω_l denotes the lth class, c_k represents the kth cluster, $p(c_k|\mathbf{x}_i)$ represents the posterior probabilities of the presence of corresponding samples and $p(\omega_l|c_k)$ denotes the cluster posterior probabilities of class membership. Notice that $p(\omega_l|c_k, \mathbf{x}_i)$ has no relationship with $\mathbf{x}_i$, and thus can be simplified as $p(\omega_l|c_k)$. According to the intuitive meaning of $p(c_k|\mathbf{x}_i)$, it can also be computed by Eq. (1). Now $p(\omega_l|c_k)$ can be computed through Bayesian theorem:

$$p\left(\omega_l \mid c_k\right) = \frac{p\left(\omega_l, c_k\right)}{p\left(c_k\right)} \tag{4}$$

where $p(c_k)$ is the prior probability and can be calculated by the proportion of the samples in the kth clusters, i.e., $Num(\mathrm{x} \in c_k)/N$; $p(\omega_l, c_k)$ is the joint distribution and similarly, can be computed in terms of the proportion of the samples in the kth cluster and meanwhile in the lth class, i.e., $Num(\mathbf{x} \in \omega_l$ and $\mathbf{x} \in c_k)/N$. Therefore, $p(\omega_l|c_k)$ can be rewritten as

$$p\left(\omega_l \mid c_k\right) = \frac{Num\left(\mathbf{x} \in \omega_l \ \ and \ \mathbf{x} \in c_k\right)}{Num\left(\mathbf{x} \in c_k\right)} \tag{5}$$

For each cluster c_k, the constraint $\sum_{l=1}^{L} p\left(\omega_l \mid c_k\right) = 1$ should be satisfied where L is the class number. Eq. (5) indicates that when $p(\omega_l|c_k)$ is large (small), the proportion of samples in cluster c_k from the class l is large (small). Now all the $p(\omega_l|c_k)$ can constitute a $K \times L$ matrix denoted by $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} p\left(\omega_1 \mid c_1\right) & p\left(\omega_2 \mid c_1\right) & \dots & p\left(\omega_L \mid c_1\right) \\ p\left(\omega_1 \mid c_2\right) & p\left(\omega_2 \mid c_2\right) & \dots & p\left(\omega_L \mid c_2\right) \\ \dots & \dots & \dots & \dots \\ p\left(\omega_1 \mid c_K\right) & p\left(\omega_2 \mid c_K\right) & \dots & p\left(\omega_L \mid c_K\right) \end{bmatrix} \tag{6}$$

It is obvious that such a relation matrix $\mathbf{P}$ can reveal the statistical relationship between the formed clusters and the given classes.

For a given training dataset with class labels, the clustering result described by u_{ik} or $p(c_k|\mathbf{x}_i)$ is *only* relevant to the clustering centers. On the other hand, the classification result yielded by $p(\omega_l|\mathbf{x}_i)$s also relies on the clustering centers. The underlying reason is that the matrix $\mathbf{P}$ is dependent on the clustering partition and its value is determined by assigning each sample to the nearest clustering centers. In summary, by using the Bayesian theory, the proposed clustering and classification mechanism are all *only* determined by the cluster centers.

3.2 Multi-objective functions for clustering and classification

Based on the above description of clustering and classification mechanism, the multi-objective clustering and classification learning can be formulated by

$$\min J(\{\mathbf{v}_k\}) = [J_1(\{\mathbf{v}_k\}),\ \ldots, J_m(\{\mathbf{v}_k\}),\ \ldots, J_M(\{\mathbf{v}_k\})] \tag{7}$$

where M is the number of objective functions and $J_m(\{\mathbf{v}_k\})$ is the mth objective function depending only on the clustering centers. Note that among the multiple objective functions, there is at least one objective function evaluating the clustering (classification) performance.

First, based on the intra-class compactness and inter-class separability, different clustering objective function can be designed. Here we just introduce three clustering criteria：

（1）Xie-Bi index [27] which is presented by :

$$J_m(\{\mathbf{v}_k\}) = \frac{\sum_{k=1}^{K}\sum_{i=1}^{N} u_{ik}^2 \left\|\mathbf{x}_i - \mathbf{v}_k\right\|^2}{N \times \min_{j\neq i} \|\mathbf{v}_i - \mathbf{v}_j\|} \tag{8}$$

(2) v_{sv} index [28] which is proposed by Kim：

$$\begin{aligned} v_u &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{|c_k|}\sum_{\mathbf{x}_j\in c_k}\left\|\mathbf{x}_j - \mathbf{v}_k\right\|^2\right) \\ v_o &= \frac{K}{\min_{j\neq i}\left\|\mathbf{v}_i - \mathbf{v}_j\right\|^2} \\ J_m(\{\mathbf{v}_k\}) &= v_u + v_o \end{aligned} \tag{9}$$

where c_k is the set of the samples falling into the cluster k, $|\, c_k \,|$ is the number of samples in c_k.

(3) In order to introduce the kernel trick to the clustering objective function, we design the intra-cluster compactness in the feature space：

$$\begin{aligned} J_m(\{\mathbf{v}_k\}) &= \sum_{k=1}^{K}\sum_{i=1}^{N} u_{ik}^m(\mathbf{v}_k)\left\|\phi(\mathbf{x}_i) - \phi(\mathbf{v}_k)\right\|^2 \\ &= \sum_{k=1}^{K}\sum_{i=1}^{N} u_{ik}^m(\mathbf{v}_k)\left(\phi(\mathbf{x}_i) - \phi(\mathbf{v}_j)\right)^T\left(\phi(\mathbf{x}_i) - \phi(\mathbf{v}_j)\right) \\ &= \sum_{k=1}^{K}\sum_{i=1}^{N} u_{ik}^m(\mathbf{v}_k)\left(\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_i) - 2\phi(\mathbf{v}_j)^T\phi(\mathbf{x}_i) + \phi(\mathbf{v}_j)^T\phi(\mathbf{v}_j)\right) \end{aligned} \tag{10}$$

where Φ is an implicit nonlinear map from the input space to a higher dimensional feature space. By using the kernel to substitute the inner product in (10), the Eq. (10) can be rewritten:

$$J_m(\{\mathbf{v}_k\}) = \sum_{k=1}^{K}\sum_{i=1}^{N} u_{ik}^m(\mathbf{v}_k)\left(K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{v}_k, \mathbf{v}_k) - 2K(\mathbf{x}_i, \mathbf{v}_k)\right) \tag{11}$$

When RBF kernel is adopted, $J_m(\{\mathbf{v}_k\})$ can be simplified as:

$$J_m(\{\mathbf{v}_k\}) = \sum_{k=1}^{K}\sum_{i=1}^{N} u_{ik}^{m}(\mathbf{v}_k)\left(2-2K(\mathbf{x}_i,\mathbf{v}_k)\right) \tag{12}$$

where $u_{ik}(\mathbf{v}_k)$ is the membership of x_i to the cluster k. Note that $u_{ik}(\mathbf{v}_k)$ is the function of the cluster center $\mathbf{v}_k$ and determined by the distance between the samples and centers in the feature space. The final objective function can be written as:

$$J_m(\{\mathbf{v}_k\}) = 2\sum_{k=1}^{K}\sum_{i=1}^{N}\left(\frac{\left(1-K(\mathbf{x}_i,\mathbf{v}_k)\right)^{-1/(m-1)}}{\sum_{j=1}^{K}\left(1-K(\mathbf{x}_i,\mathbf{v}_j)\right)^{-1/(m-1)}}\right)^{m}\left(1-K(\mathbf{x}_i,\mathbf{v}_k)\right) \tag{13}$$

Second, based on the classification mechanism designed in the subsection 3.1, the different classification objective functions can be designed. Here we just list the two classification criteria:

(1) Minimization of the misclassification rate:

$$J_m(\{\mathbf{v}_k\}) = \sum_{i=1}^{N}\delta\left(f(\mathbf{x}_i), y_i\right)\Big/N \tag{14}$$

where y_i is the class label of $\mathbf{x}_i$ and $y_i \in \{1, 2, \ldots, L\}$.

(2) Minimization of squared error between the target outputs and actual outputs:

$$J_m(\{\mathbf{v}_k\}) = \sum_{i=1}^{N}\sum_{j=1}^{L}\left(p(\omega_j \mid \mathbf{x}_i) - y_{il}\right)^2 \tag{15}$$

where $p(\omega_l|\mathbf{x}_i)$ is the class *posterior probabilities of* $\mathbf{x}_i$ and y_{il} is the membership of $\mathbf{x}_i$ to the lth class. Here y_i is represented by one-of-c coding. For example, if there are 4 classes in the given dataset and the sample $\mathbf{x}_i$ belongs to the third class, then its class label y_i is encoded by [0, 0, 1, 0].

In this paper, without loss of generality, we just adopt the two functions to formulate the clustering and classification problems:

$$\min J(\{\mathbf{v}_k\}) = [J_1(\{\mathbf{v}_k\}), J_2(\{\mathbf{v}_k\})] \tag{16}$$

where $J_1(\{\mathbf{v}_k\})$is the misclassification rate and $J_2(\{\mathbf{v}_k\})$ measures the compactness in the feature space. Eq. (16) aims to *simultaneously* minimize the classification criterion $J_1(\{\mathbf{v}_k\})$ and the clustering criteria $J_2(\{\mathbf{v}_k\})$. No matter what clustering or classification criterion is selected from the above criteria, the values of $J(\{\mathbf{v}_k\})$ all only depend on a set of the cluster centers. By just optimizing the centers embedded in $J(\{\mathbf{v}_k\})$, the clustering and classification criteria can be optimized at the same time.

3.3 Optimization of Multi-objective functions

To describe the concept of optimality in the multi-objective functions, we will introduce a few definitions [22] involving in multi-objective optimization.

Definition 1: (Dominance) For a given multi-objective problem min $J(\mathbf{x})=[J_1(\mathbf{x}), J_2(\mathbf{x}), \ldots, J_M(\mathbf{x})]$, the solution $\mathbf{x}^1$ dominates $\mathbf{x}^2$ or the solution $\mathbf{x}^2$ is inferior to $\mathbf{x}^1$ (denoted as $\mathbf{x}^1 \prec \mathbf{x}^2$) if the following two conditions are held: (1) $\forall\, i\in[1,2, \ldots, M], J_i(\mathbf{x}^1)\leq J_i(\mathbf{x}^2)$; (2) $\exists\; i\in[1,2, \ldots, M], J_i(\mathbf{x}^1)< J_i(\mathbf{x}^2)$.

Definition 2: (Pareto Optimality) The solution $\mathbf{x}^0$ is **Pareto optimal** if there exists no solution $\mathbf{x}^1$ such that $\mathbf{x}^1 \prec \mathbf{x}^0$**.**

Definition 3: (Pareto Optimal Set) The Pareto optimal set is defined as $Ps = \{\mathbf{x}^0 \mid \bar{\exists}\mathbf{x}^1 \prec \mathbf{x}^0\}$.

Definition 4: (Pareto Front) For a given Pareto optimal set Ps, the Pareto front is defined as

$$P_F = \{J(\mathbf{x}) = (J_1(\mathbf{x}), J_2(\mathbf{x}), \ldots, J_M(\mathbf{x})) \mid \mathbf{x} \in P_s\}.$$

To explain the above concepts clearly, we give Fig. 5 under the condition of two objective functions. The white 'o' denotes a dominated solution and the dark 'o' represents a Pareto-optimal solution which is also termed *non-dominated* solution. According to definition 1, the solution $\mathbf{x}^0$ dominates $\mathbf{x}^1$ and $\mathbf{x}^2$; the solution denoted by the dark 'o' is Pareto-optimal in terms of definition 2; all the dark 'o's constitute the Pareto optimal set in terms of definition 3; according to definition 4, Pareto Front is composed of the objective values of all the dark 'o's.

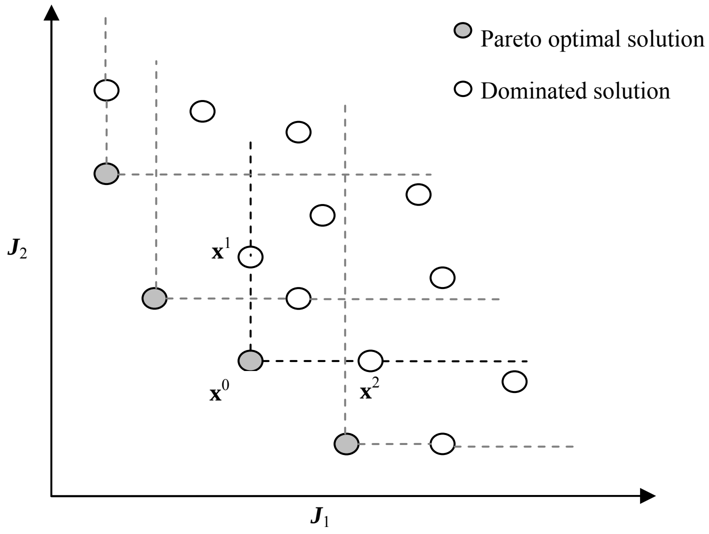


Fig. 5 The Pareto front of a set of solutions in a two objective space.

Next, we employ the above concepts to briefly discuss the existing optimization methods for multi-objective problems. Classical methods suggest converting the multi-objective optimization problem to a single-objective optimization problem by objective weighting. By introducing a weight parameter *β*, the optimization for the multiple objective functions in Eq. (16) can be transformed to:

$$\min J(\{\mathbf{v}_k\}) = J_1(\{\mathbf{v}_k\}) + \beta\, J_2(\{\mathbf{v}_k\})] \qquad (17)$$

By optimizing Eq. (17) instead of Eq. (16), a single Pareto-optimal solution (i.e., clustering centers) that makes a balance between the clustering performance and classification performance can be obtained. However, this single point solution is usually sensitive to the weight *β* [29], as a result, in order to get a solution as optimal as possible, multiple sets of different weights have to be used and leading to the same problem being solved many times.

In recent years, a number of multi-objective evolutionary algorithms (MOEA) [22, 29, 30] have been suggested such as Non-dominated sorting genetic algorithm (NSGA) [29] and Pareto Archive Evolutionary Strategy (PAES) [30]. The primary reason for this is their ability to find multiple Pareto-optimal solutions rather than a single solution in one single simulation run. Some researchers suggested that multi-objective search and optimization might be a problem area where EA's do better than other blind search strategies [29, 30]. In 2004, Coello [22] et al. proposed a multi-objective Particle Swarm Optimization named MOPSO and proved its good performance and the high speed of convergence. MOPSO is an evolutionary technique through individual improvement plus population cooperation and competition. Many works [31, 32] have been done and shown that PSO-type methods are prevailing population-based optimization algorithms and successful in a wide variety of learning tasks such as attribute selection in a bioinformatics data set, time series prediction and Face classification problems. MOPSO utilizes an external repository to keep a historical record of the non-dominated solutions found along the search process. In its implement, MOPSO employs this repository to guide the flight of the current particles and store the non-dominated solutions.

In this paper, we adopt the simplified version of MOPSO to solve the multi-objective optimization of MSCC. By using the MOPSO, the multiple sets of Pareto-optimal clustering centers can be acquired in the two objective spaces. Since the clustering and classification learning methods can complement each other, the corresponding two criteria can also have the complementarity to some extent, as a result, those Pareto-optimal clustering centers which attain relatively low values jointly for both the clustering compactness and classification error rate on the training data can consistently achieve the best clustering or classification result on the corresponding test data (later given in experiments).

In the MOPSO, each individual of the population is called a ‘particle’, which, in fact, represents a solution to a problem. Here a particle $\mathbf{x}_i=[x_{i1}, x_{i2}, \ldots x_{id}, \ldots, x_{iD}]$ in MSCC is a vector composed of all the clustering centers and its dimension is $D=d\times K$. Each particle ‘flies’ around in the multi-dimensional research space with a velocity $\mathbf{vel}_i=[vel_{i1}, vel_{i2}, \ldots vel_{id}, \ldots, vel_{iD}]$. This velocity is updated by the experience of particle itself and repository

$$vel_{id}^{t+1} = w\times vel_{id}^{t} + r_1 \times \left(pbset_{id}^{t} - x_{id}^{t}\right) + r_2 \times \left(Repository_d(h) - x_{id}^{t}\right) \qquad (18)$$

where t is the current iteration number, w is inertia weight and set to 0.4, r_1 and r_2 are two independent random numbers uniformly distributed in the range of [0, 1]. $\mathbf{pbest}_i=[pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD}]$ represents the best position that the ith particle has had. $\mathbf{Repository(}h\mathbf{)}=[Repository_{h1}, Repository_{h2}, \ldots, Repository_{hD}]$ is a value randomly taken from the repository and h is the selected index. The position of each particle at each generation is updated by

$$x_{id}(t+1) = x_{id}(t) + vel_{id}(t+1) \qquad (19)$$

The whole process of using the simplified version MOPSO can be summarized as follows:

```
MSCC Learning Algorithm
Step1: Set the number $P$ of particles to 500, the maximum number $I$ of iterations to 100 and the current
       iteration number $t$ to 1; Initialize the particles with random positions and velocities.
Step2: Evaluate the two objective values of all particles according to (13) and (14) and set $\mathbf{pbest}_i$ of each
       particle equal to its current position.
Step3: Store the positions of the particles that represent non-dominated solutions in the Repository.
Step4: While $I>t$
(a) Compute the speed of each particle by (18).
(b) Compute the new position $x_i$ of each particle according to (19).
(c) Evaluate the two objective values of particles in terms of (13) and (14).
(d) Find all the currently non-dominated locations (the non-dominated solutions found at each iteration):
    For m=1: $P$
      Non_dominated_flag=1;
      For n=1: $P$
        If $x_m$ is dominated by $x_n$
```

```
                Non_dominated_flag=0;
            End;
        End;
        If Non_dominated_flag ==1
            $x_{\mathrm{m}}$ is the currently non-dominated location.
        End;
    End;
(e) Insert all the currently non-dominated locations into **Repository**;
     Eliminate any dominated locations from the **Repository.**
(f) If the current position $\mathbf{x}_i$(t)of the particle dominates $\mathbf{pbest}_i$
                $\mathbf{pbest}_i$=$\mathbf{x}_i$(t);
     else if the $\mathbf{pbest}_i$ dominates $\mathbf{x}_i$(t),
                $\mathbf{pbest}_i$ is kept;
     else if neither of them is dominated by the other
                 $\mathbf{pbest}_i$ is updated or kept randomly
     End;
(g) Update $t$=$t$+1.
```

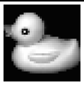
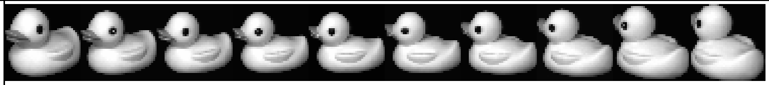
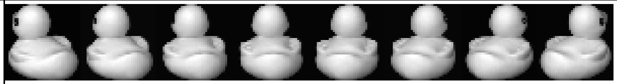
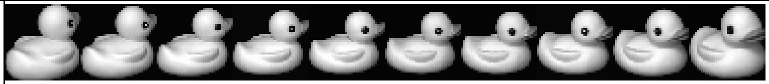
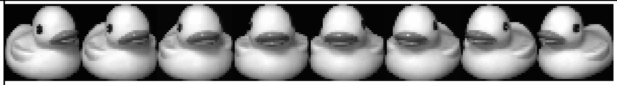
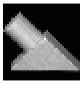
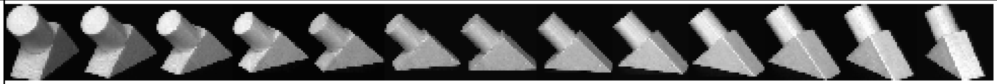
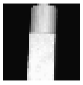
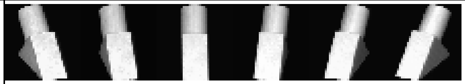
3.4 Time complexity Analysis of MSCC

The time complexity of MSCC is O(I×P×max(K×d, P×M, N×K×L)) where I is the maximum iteration number, P is the particle number, M is the objective function number, K is the cluster number, d is the data dimension, N is the sample number and L is the class number. In our experiment, I, P and M are the user-specified parameters and set to the constant values 500, 100 and 2, respectively. Moreover, K, d, N and L are the variable parameters dependent on the chosen dataset. It is worth pointing out that the larger the cluster number (or the data dimension, the sample number, the class number) is, the more the computational time.

3.5 A toy illustration for MSCC benefit

Here we give a toy illustration on the dataset COIL [33] to explain why simultaneous classification and clustering learnings can give more than just either classification leraning or clustering learning. COIL is available at http://www.cs.columbia.edu/CAVE. The full COIL dataset consists of images of 100 objects where the images of the objects were taken at pose intervals of 5^{o}, i.e., 72 poses per object. In this paper, we have used a part of the COIL database by involving only the first 2 objects, with 144 images in total. The training set consists of 36 images (one for every 10^{o}) for each object, and the test set consists of the remaining 36 images for each object [34]. For such dataset, classification algorithms only pay attention to the class information of objects; clustering algorithms only care similarity among objects. In contrast, our algorithm utilizes both the class information and structural information to not only classify the objects to different classes, but also discover the objects with similar poses. As shown in Table1, the objects grouped to the same clusters have very similar poses, which indicates that the structure hidden in data is discovered. Moreover, the relation matrix in Table 2 means that the objects falling into clusters c_1, c_2, c_3 and c_4 belong to class ω_1 and similarly, the objects in clusters c_5, c_6, c_7 and c_8 belong to class ω_2. Due to the correct clustering and so-generated relationship matrix **P**, MSCC achieves the classification accuracy of 100%. From this example, we can see that MSCC discovers both the structures hidden in data and the relationship between the structures and their classes, which makes COIL dataset prone to be transparent and interpretable. However, SVM has difficulty to great extent to *simultaneously* achieve the two aspects.

Table 1 Clustering results of the test samples on the COIL dataset

Cluster *i*	Cluster center (pose angle)	Samples belonging to the *i*th cluster
1	 0	 320 330 340 350 0 10 20 30 40 50
2	 90	 60 70 80 90 100 110 120 130
3	 180	 140 150 160 170 180 190 200 210 220 230
4	 270	 240 250 260 270 280 290 300 310
5	 0	 300 310 320 330 340 350 0 10 20 30 40 50 60
6	 90	 70 80 90 100 110 120

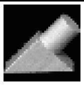
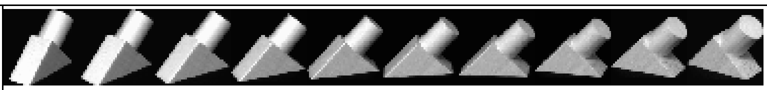
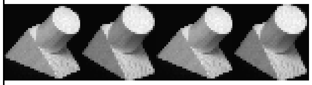
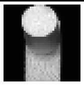
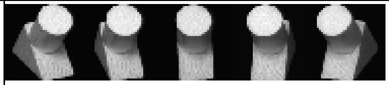
7	 180	 130 140 150 160 170 180 190 200 210 220  210 220 230 240
8	 270	 250 260 270 280 290

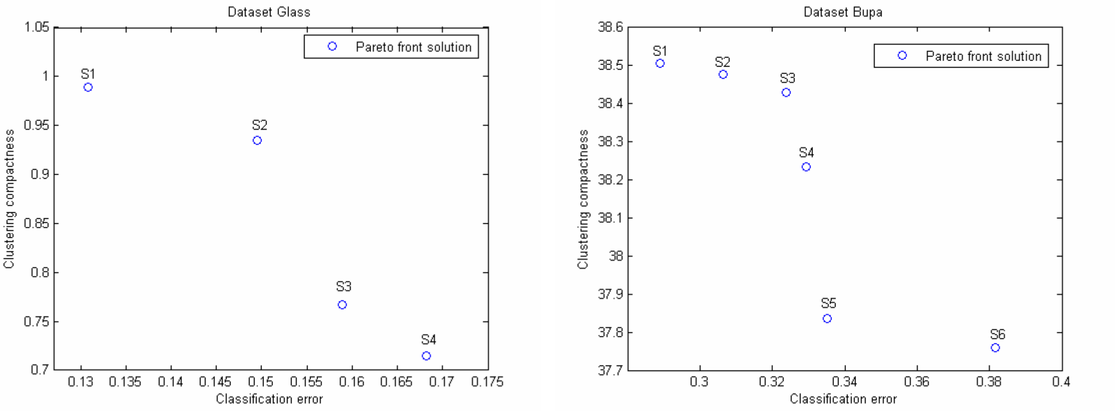
Table 2 Classification results and the parameters on COIL dataset

	MSCC
Relation matrix **P**	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T$
Accuracy	100%

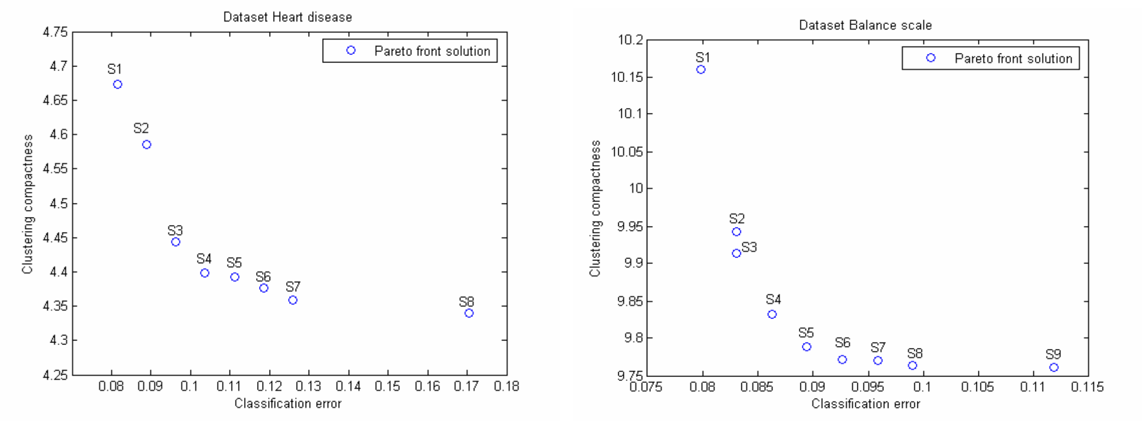
4. Experimental results

4.1 Pareto optimal solution

To investigate the property of the Pareto-optimal solutions, we give the Pareto-optimal front on the datasets *Glass*, *Bupa*, *Heart_disease* and *Balance_scale* respectively in Fig. 6. It can be observed that MSCC acquires 4, 6, 8 and 9 Pareto optimal solutions on these four datasets, respectively. This result implies that a *solution-inconsistency* can in general occur in this multiple objective problem [22].



(a) Glass (b) Bupa



(c) Heart_disease

(d) Balance_scale

Fig. 6 Pareto fronts obtained on the datasets *Glass*, *Bupa*, *Heart_disease* and *Balance_scale*, respectively

Furthermore, Tables 3, 4, 5 and 6 list the J_1 values and J_2 values of the Pareto-optimal solutions on both the training data and test data of *Glass*, *Bupa*, *Heart_disease* and *Balance_scale*, respectively. From these tables, we can find that on the test datasets, S2, S2, S7 and S2 in each table obtain the best classification performance, and while S3, S3, S5 and S8 achieve the best clustering performance. From this result, we can have an **interesting observation that the best performance of clustering or classification on the test datasets corresponds to those solutions which achieve relatively low values of the objectives of both clustering compactness and classification error rate on the training datasets**. This conclusion empirically demonstrates the consistency or complementarity between the clustering and classification learnings. In other words, the pursuit for good clustering compactness is beneficial to classification learning, while the pursuit for high classification accuracy is helpful for the clustering compactness.

Table 3 Misclassified rate and clustering compactness on the training and test dataset of *Glass*

Pareto optimal solution	Training misclassified rate J_1	Training clustering compactness J_2	Test misclassified rate J_1	Test clustering compactness J_2
S1	0.1308	0.9895	0.3551	4.2538
S2	0.1495	0.9345	**0.2897**	4.1861
S3	0.1589	0.7668	0.3178	**4.0213**
S4	0.1682	0.7146	0.4206	4.0632

Table 4 Misclassified rate and clustering compactness on the training and test dataset of *Bupa*

Pareto optimal solution	Training misclassified rate J_1	Training clustering compactness J_2	Test misclassified rate J_1	Test clustering compactness J_2
S1	0.2890	38.5036	0.3488	43.6930
S2	0.3064	38.4753	**0.3372**	44.1799
S3	0.3237	38.4271	0.3605	**43.6214**
S4	0.3295	38.2328	0.3779	46.1935
S5	0.3353	37.8356	0.3779	47.3340
S6	0.3815	37.7587	0.3605	46.9800

Table 5 Misclassified rate and clustering compactness on the training and test dataset of *Heart_disease*

Pareto optimal solution	Training misclassified rate J_1	Training clustering compactness J_2	Test misclassified rate J_1	Test clustering compactness J_2
S1	0.0815	4.6729	0.2074	6.0206
S2	0.0889	4.5855	0.1926	6.1621
S3	0.0963	4.4441	0.2148	5.8755
S4	0.1037	4.3977	0.2296	5.6910
S5	0.1111	4.3925	0.2148	**5.6571**
S6	0.1185	4.3762	0.2074	5.7172
S7	0.1259	4.3589	**0.1852**	5.9531
S8	0.1704	4.3402	0.2593	5.9170

Table 6 Misclassified rate and clustering compactness on the training and test dataset of *Balance_scale*

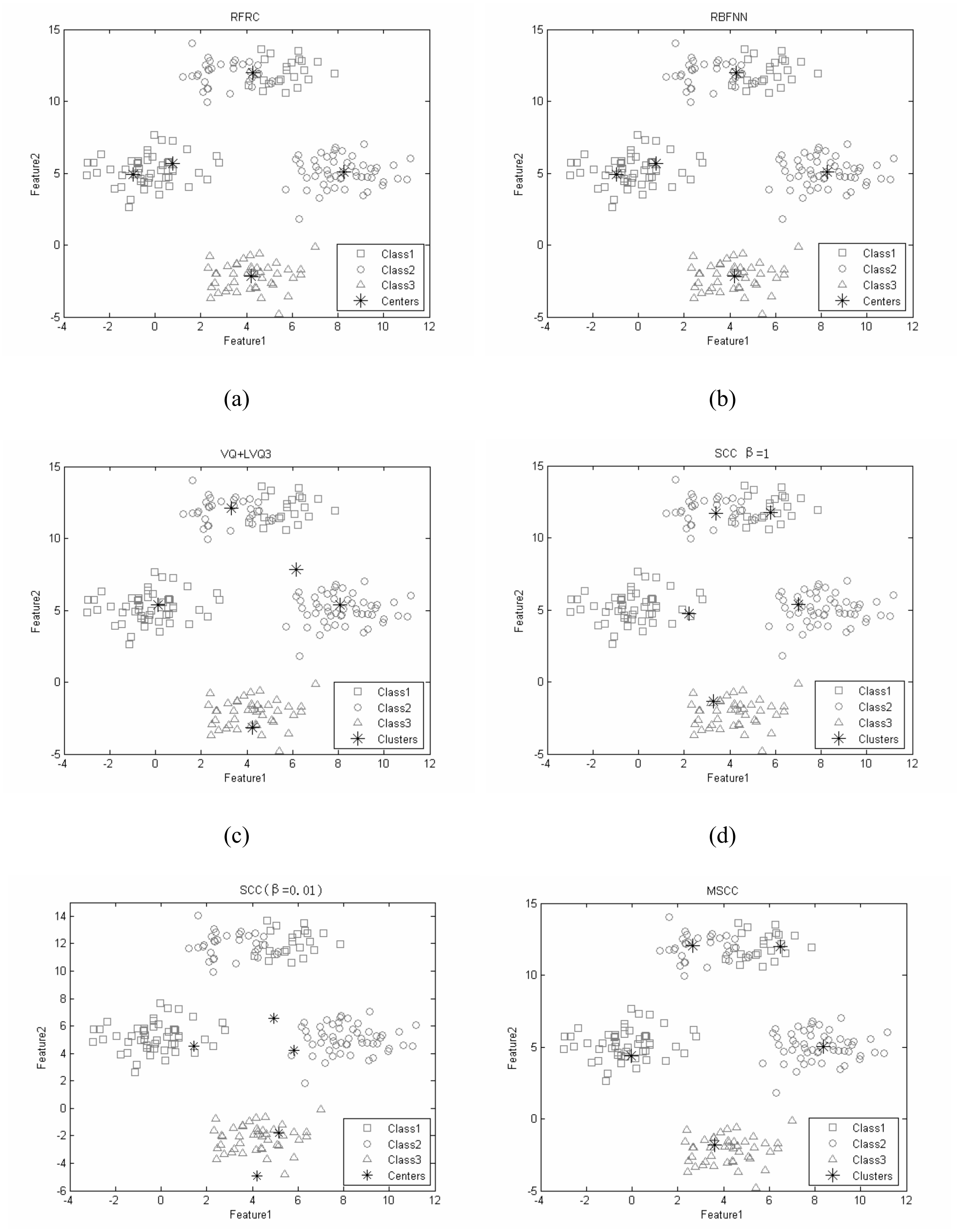
Pareto optimal solution	Training misclassified rate J_1	Training clustering compactness J_2	Test misclassified rate J_1	Test clustering compactness J_2
S1	0.0799	10.1600	0.0994	10.4577
S2	0.0830	9.9428	**0.0962**	10.2207
S3	0.0832	9.9139	0.0994	10.1893
S4	0.0863	9.8318	0.1058	10.0685
S5	0.0895	9.7891	0.1058	10.0603
S6	0.0927	9.7713	0.1250	10.0442
S7	0.0958	9.7702	0.1186	10.0104
S8	0.0990	9.7640	0.1186	**10.0047**
S9	0.1118	9.7616	0.1346	10.0708

4.2 Synthetic dataset

We apply RBFNN, RFRC, VQ+LVQ3, SCC and MSCC on a synthetic dataset in Table 7 to compare their both classification and clustering ability. Here the number K of cluster centers is set to 5 and the scale factor λ of the RBF kernel is 1. To evaluate their clustering effectiveness, we list the obtained clustering centers in Fig. 7. It can be seen from this figure that in RBFNN and RFRC, the samples localized in the upper part of each panel are characterized by one clustering center, but in fact these samples come from different classes (i.e., Class 1 and 2) and hence should be categorized into different clusters in terms of their class labels. In VQ+LVQ3, there exists a clustering center deviated from the distribution of the given samples, thus failing to precisely describe the data distribution. In SCC, when a proper value is selected for β, the correct clustering result is obtained as show in Fig. 7(d); however, when an improper value is selected, the obtained clustering result is unable to uncover the structure in data as shown in Fig. 7(e). So in order to get a solution as optimal as possible, multiple sets of different weights have to be used and thus leading to that the same problem has to be solved many times. In contrast, MSCC removes the weight parameter β and obtains the correct clustering centers located in the proper places, and thus reflects the inherent structure in this data relatively correctly.

Table 7 Synthetic dataset 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)



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 7 Cluster centers obtained by RFRC, RBFNN, VQ+LVQ3, SCC and MSCC, respectively

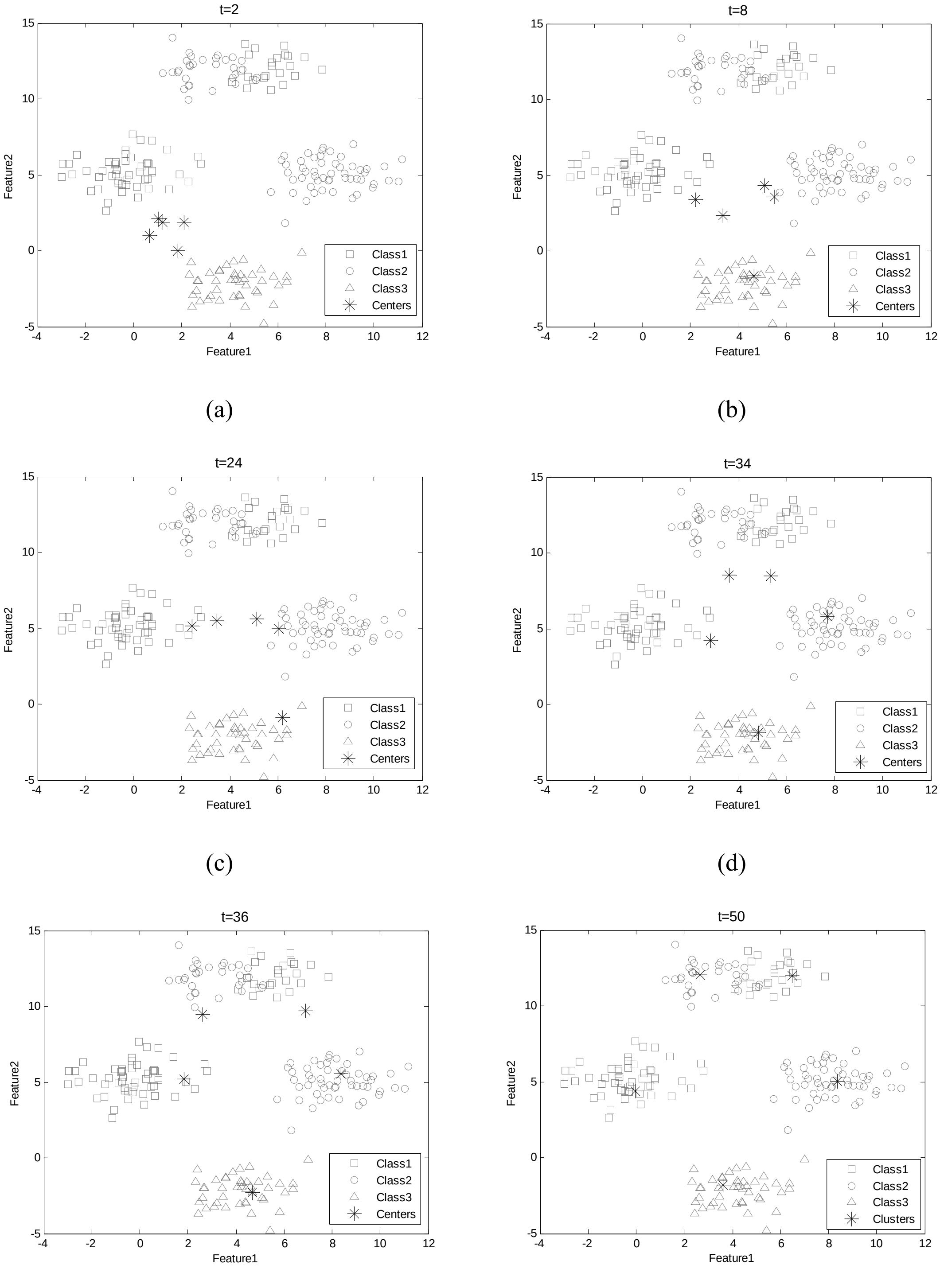
To further compare the classification effectiveness, we present the relation matrices (connecting weights) and classification accuracy in Table 8. From this table, we can make the following analyses that (1) the connecting weights in RBFNN are yielded by optimizing the MSE criterion between target and actual outputs, as a result, their values do not have any intuitive meaning; the relation matrix in RFRC are obtained by the composite operators, thus lacks the statistical meaning; in VQ+LVQ3, its relation matrix are determined by the class labels of clustering centers, such hard values can not quantatively reflect the fuzzy belonging degree between clusters and classes; in SCC, the *larger* the β value is, the more attention the objective function pays on the classification problem; the *smaller* the β value is, the more attention the objective function pays on the clustering problem, as a result, a proper value should be selected for β so that a balance can be made between the classification and the clustering performances, and thus the correct result can be obtained as shown in Table 8 (β=1); in MSCC, the relation matrix can not only reveals the underlying logical relationship in data but also the quite precise statistical relationship between the formed clusters and given classes. (2) due to the wrong clustering centers and imprecise relation matrix (connecting weights), RBFNN, RFRC and VQ+LVQ3 fail to achieve the satisfying classification performance; SCC can achieve the high classification accuracy of 98.5%, but an exhaustive search for the weight parameter β has to be executed in some range, which is a heavier burden; in contrast, MSCC achieves the highest classification accuracy of 99.0%, indicating that its classification mechanism works better the other algorithms, such good performance can attribute to its correct clustering centers and real relation matrix.

From this initial empirical evaluation, it can be concluded that MSCC can achieve the effective clustering and classification performance at one time. The underlying reason is that it optimizes the clustering and classification criterion simultaneously, thus does not need to sacrifice the clustering performance for the classification performance, or vice versa.

Table 8 Parameter comparison among RFRC, RBFNN, VQ+LVQ3 and MSCC

Parameters	RBFNN	RFRC	VQ+LVQ3	SCC (β=0.01)	SCC (β=1)	MSCC
Relation matrix	$\begin{bmatrix} 1.33 & -4.26 & -0.63 \\ 0.11 & 0.71 & 0.36 \\ 0.87 & 0.35 & -0.42 \\ -1.50 & 4.86 & 0.15 \\ -0.30 & -0.31 & 1.33 \end{bmatrix}$	$\begin{bmatrix} 0.03 & 0.10 & 0.00 \\ 0.87 & 0.00 & 0.00 \\ 0.78 & 0.09 & 0.09 \\ 0.00 & 0.83 & 0.00 \\ 0.00 & 0.00 & 0.80 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.69 & 0.31 & 0 \\ 0 & 1.00 & 0 \\ 0 & 0 & 1.00 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.94 & 0.06 & 0 \\ 0 & 1.00 & 0 \\ 0.09 & 0.91 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.94 & 0.06 & 0 \\ 0 & 1.00 & 0 \\ 0.09 & 0.91 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$
Accuracy	83.5%	79.5%	86.5%	86.5%	98.5%	99.0%

To make the iterative process of MSCC clearer, we give the intermediate results of the clustering centers in Fig.8 below, and their corresponding relation matrix and classification accuracy in Table 9. From Fig.8, it can be seen that as the iteration step t increases from 2 to 50, the obtained clustering centers tend to gradually exhibit the real structure hidden in data. Moreover, from Table 9, it can be observed that during the iterative process, the resulted relation matrix **P** tends to gradually discover the correct relationship between the structures and classes, and the corresponding classification accuracy increases from 86.7% to 99.0%.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 8 Iterative process of MSCC

Table 9 Parameters of MSCC at different iteration step

Iteration step	t=2	t=8	t=24	t=34	t=36	t=50
Relation Matrix **P**	$\begin{bmatrix} 0.70 & 0.30 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.30 & 0.70 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.68 & 0.32 & 0 \\ 0 & 0 & 0 \\ 0 & 1.00 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.91 & 0.09 & 0 \\ 0.30 & 0.70 & 0 \\ 0 & 1.00 & 0 \\ 0.02 & 0 & 0.98 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.85 & 0.15 & 0 \\ 0 & 1.00 & 0 \\ 0.13 & 0.87 & 0 \\ 0.02 & 0 & 0.98 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.91 & 0.09 & 0 \\ 0 & 1.00 & 0 \\ 0.09 & 0.91 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 & 0 \\ 0.94 & 0.06 & 0 \\ 0 & 1.00 & 0 \\ 0.09 & 0.91 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$
Accuracy	86.7%	88.4%	88.6%	96.5%	98.0%	99.0%

4.3 Real-life dataset

We evaluate the classification capability of MSCC on real-life datasets. We select 20 datasets from the UCI Machine Learning Repository [35] which is a repository of databases for the empirical analysis of machine learning algorithms. The classification performance comparison is made among RFRC, VQ+LVQ3, RBFNN, SVM, Clustering based SVM (named CBSVM)[1], SCC and MSCC. 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 [36] where N is the number of the training samples. In RFRC, RBFNN, SVM, SCC and MSCC, the RBF kernel is adopted and its scale factor λ is determined by searching in {0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15}. In SVM, the regularization parameter C is determined from $\{2^{-1}, 2^{0}, 2^{3}, 2^{5}, 2^{7}, 2^{9}\}$. In SCC, the weight parameter β is selected from {0.01, 0.1, 1}. In MSCC, since the multiple Pareto-optimal solutions can be obtained, the final solution is determined by the trial-and-error approach [37] associated with the classification accuracy. Due to the multiple parameters existing in these algorithms, the Discrete Grid Search [38] based on exhaustive search in a limited range is adopted to acquire the optimal values for these parameters. In what follows, we list the number K of the cluster centers and the scale factor λ used in the experiments in Table 10.

[1] CBSVM is our purposely-designed classifier for more extensive comparison. CBSVM adopts the same architecture as RBFNN, but chooses a different loss function. Specifically, like RBFNN, CBSVM first also uses an unsupervised k-means to obtain the cluster centers as parameters of a set of Gaussian functions to establish a mapping from the input to the space formed by a set of the functions, but unlike RBFNN, CBSVM adopts the SVM (loss) criterion rather than the least square error criterion in training in the above mapped space. CBSVM also falls into the two-step framework which optimizes a clustering criterion first, and then the classification criterion associated with the clustering result, but it fails to realize the simultaneous optimization for such two learnings.

Table 10 The number of the clustering centers and the scale factor of RBF kernel used in algorithms

Dataset	RFRC		VQ+LVQ3	RBFNN		SVM	CBSVM		SCC		MSCC	
(#samples×#dim×#class)	*K*	*λ*	*K*	*K*	*λ*	*λ*	*K*	*λ*	*K*	*λ*	*K*	*λ*
WBCD (683×9×2)	6	0.01	20	4	1	1	4	1	4	1	2	0.1
Water (116×38×2)	6	1	4	4	0.001	1	2	1	2	1	2	0.01
Thyroid (215×5×3)	10	0.1	20	26	1	0.1	16	10	14	1	10	0.001
Lung_cancer (32×56×3)	16	0.01	12	14	0.01	0.01	6	0.1	4	0.01	6	0.01
Pid (768×8×2)	60	0.001	80	22	0.1	1	30	1	20	1	10	0.1
Soybean_small (47×35×4)	24	0.01	24	20	0.1	1	12	0.1	8	1	4	1
WDBC (569×30×2)	6	0.01	60	18	0.01	1	4	1	2	0.1	2	0.001
Waveform (5000×21×3)	100	1	100	100	1	1	100	1	100	0.01	100	0.01
Balance_scale (625×4×3)	26	0.01	16	18	0.01	1	22	1	10	0.1	16	0.01
Heart_disease (270×13×2)	60	1	70	16	0.001	0.01	30	0.1	12	0.01	34	0.01
Pima_Indian_diabetes(768×8×2)	30	0.001	28	10	1	1	30	0.1	10	0.01	25	0.01
Glass (214×9×6)	30	0.1	30	20	1	0.1	50	10	20	1	20	0.01
Sonar (208×60×2)	20	0.01	92	20	1	1	28	0.1	18	0.01	18	0.01
Wine (178×13×3)	14	1	14	6	0.001	1	4	0.1	6	0.001	6	0.1
Ecoli (336×7×8)	26	0.001	50	12	1	1	30	1	14	0.001	24	0.01
Lenses (24×4×3)	5	0.1	5	5	0.01	1	4	1	5	0.01	4	0.01
Iris (150×4×3)	9	0.1	12	12	1	1	12	1	12	0.001	12	0.001
Bupa (345×6×2)	30	0.01	30	22	0.1	0.1	26	0.1	10	0.001	28	0.1
Image segmentation (2310×19×7)	100	1	100	100	1	1	100	1	100	1	100	1
Spambase (4601×57×2)	68	1	68	68	1	0.1	60	0.1	18	0.001	18	0.001

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 for 10 times, and only their averaged accuracies and the corresponding standard deviations are reported in Table 11.

First, we compare the classification results yielded respectively by SCC and MSCC. It can be seen from the table that on all of the datasets, the accuracies of MSCC are respectively better than those of SCC. Especially, on the datasets *Lung_cancer*、*Lenses*、*Sonar* and *Glass*, MSCC achieves significant promotion of 9.8%、9.2%、4.8% and 4%, respectively. Such a promotion of MSCC can attribute to effectiveness of the multi-objective form and diversity of the multiple Pareto-optimal solutions. In comparison with SCC, MSCC has two advantages: (1) by utilizing the multi-objective functions to describe the clustering and classification problems, respectively, MSCC can remove the weighting parameter in SCC, and thus the computational burden for choosing this parameter can be exempted; (2) by extending the single solution to multiple solutions, MSCC can improve the effectiveness of SCC. Moreover, it is worth pointing out that in SCC, its maximum iteration number *I* and its particle number *P* are respectively set to 500 and 1000; while in MSCC, they just are 100 and 500 and much less than those in SCC, which is naturally favorable for reduction of the learning.

Second, we make the comparison among MSCC, RFRC, VQ+LVQ3 and RBFNN. Compared to RFRC and VQ+LVQ3, MSCC achieves better performance on all of the datasets. Compared to RBFNN, it yields better performance on the 17 datasets, comparable performance on the 2 datasets and worse performance on the 1 dataset. The excellent classification performance of MSCC comes from its effective learning mechanism.

Finally, to give a baseline reference, we make comparison against state-of-the-art classifier SVM and our purposely-designed algorithm CBSVM. It is worth pointing out that CBSVM is superior to SVM in classification ability mainly due to the incorporation of the clustering information into CBSVM, which states that combing clustering and SVM (like the algorithms introduced in Section2) should also be effective to some degree and thus deserves a further exploration. More importantly, we can observe that compared to SVM, MSCC gains higher performances on the 12 datasets, and further compared to CBSVM, MSCC possesses higher accuracy on the 12 datasets, comparable accuracy on the other 4 datasets, all of which indicate that MSCC is highly competitive with the state-of-the-art classifiers in classification accuracy. In addition, MSCC still possesses the following advantages: (1) both the effective classification result and clustering result can be simultaneously obtained; (2) the class posterior probabilities computed in this framework can reflect the confidence of the classification decision, which is important for reliable and interpretable classification.

Table 11 Classification accuracy comparison on real-life datasets

Dataset (#samples×#dim×#class)	RFRC	VQ+LVQ3	RBFNN	SVM	**CBSVM**	SCC	MSCC
WBCD (683×9×2)	97.0 ± 0.6	96.8 ± 0.6	96.8 ± 0.5	96.9 ± 0.5	96.9 ± 0.6	97.0 ± 0.4	**97.6 ± 0.6**
Water (116×38×2)	97.9 ± 1.3	98.4 ± 1.2	98.3 ± 1.0	98.5 ± 0.8	98.3 ± 1.1	98.4 ± 1.2	**99.7 ± 0.7**
Thyroid (215×5×3)	91.8 ± 2.0	92.7 ± 2.2	95.3 ± 1.0	95.2 ± 1.5	95.3 ± 1.2	96.4 ± 1.5	**96.4 ± 1.6**
Lung_cancer (32×56×3)	40.6 ± 11.3	42.5 ± 10.8	43.8 ± 15.8	41.9 ± 8.4	41.9 ± 6.9	48.3 ± 14.2	**58.1 ± 4.9**
Pid (768×8×2)	69.6 ± 2.8	72.1 ± 2.0	74.6 ± 2.5	76.4 ± 1.7	76.5 ± 1.4	76.6 ± 0.3	**77.0 ± 2.5**
Soybean_small (47×35×4)	99.1 ± 1.7	96.1 ± 10.4	98.1 ± 1.7	98.3 ± 3.5	98.3 ± 2.9	99.6 ± 1.3	**100 ± 0.0**
WDBC (569×30×2)	92.0 ± 1.6	96.4 ± 0.9	95.0 ± 1.2	**97.2 ± 0.7**	**97.4 ± 0.8**	96.8 ± 0.7	**97.3 ± 0.7**

Waveform (5000×21×3)	83.0 ± 0.5	85.1 ± 0.6	**86.5 ± 0.9**	86.2 ± 0.4	86.0 ± 0.3	86.2 ± 0.6	**86.5 ± 0.3**
Balance_scale (625×4×3)	84.7 ± 1.5	86.0 ± 1.8	90.5 ± 1.0	90.5 ± 1.0	**93.3 ± 1.2**	90.6 ± 1.3	90.8 ± 1.2
Heart_disease (270×13×2)	80.9 ± 2.2	81.4 ± 1.8	82.5 ± 2.3	83.3 ± 2.2	83.1 ± 2.1	83.0 ± 2.1	**84.2 ± 1.8**
Pima_Indian_diabetes(768×8×2)	70.7 ± 3.2	72.6 ± 2.0	74.2 ± 2.3	76.3 ± 2.0	**77.0 ± 1.4**	76.0 ± 1.4	76.5 ± 1.1
Glass (214×9×6)	63.8 ± 3.8	63.2 ± 3.6	65.0 ± 3.8	**68.5 ± 3.5**	67.2 ± 3.0	64.9 ± 2.5	**68.9 ± 2.5**
Sonar (208×60×2)	77.5 ± 3.9	73.9 ± 2.8	80.2 ± 3.0	**85.4 ± 3.3**	**85.4 ± 4.1**	80.8 ± 5.1	**85.6 ± 4.1**
Wine (178×13×3)	96.0 ± 1.7	96.5 ± 1.5	97.3 ± 1.1	**98.4 ± 1.1**	**98.1 ± 1.0**	97.1 ± 1.8	**98.3 ± 1.3**
Ecoli (336×7×8)	81.8 ± 3.3	78.8 ± 3.0	**85.2 ± 2.7**	**85.0 ± 1.7**	**85.1 ± 1.8**	83.7 ± 1.8	**85.0 ± 2.4**
Lenses (24×4×3)	71.7 ± 7.6	74.2 ± 11.5	75.8 ± 14.6	75.1 ± 10.4	76.2 ± 10.4	77.5 ± 3.7	**86.7 ± 11.9**
Iris (150×4×3)	95.3 ± 1.1	94.7 ± 1.9	96.4 ± 1.6	95.9 ± 15	95.6 ± 1.7	95.2 ± 1.4	**97.1 ± 1.7**
Bupa (345×6×2)	61.0 ± 2.4	62.1 ± 3.7	**70.8 ± 3.6**	66.7 ± 7.5	69.9 ± 3.0	67.5 ± 5.8	68.2 ± 5.7
Image segmentation (2310×19×7)	91.1 ± 1.6	90.5 ± 1.1	95.1 ± 0.5	91.0 ± 0.4	90.8 ± 0.6	91.5 ± 1.0	**92.2 ± 0.6**
Spambase (4601×57×2)	85.1 ± 1.1	88.5 ± 0.7	80.7 ± 1.0	89.2 ± 0.6	**90.4 ± 0.5**	88.1 ± 1.3	89.9 ± 0.8

5. Conclusion

To fuse the strengths of classification learning and clustering learning, many existing algorithms such as RBFNN, RFRC, VQ+LVQ3, CCAS and ECCAS *sequentially* and separately optimize the clustering criterion and the classification criterion. Such a two-step optimization process limit the effectiveness of both clustering and classification learning to great extent. Different from these algorithms, in this paper, a multi-objective simultaneous learning framework named MSCC is presented for simultaneous clustering and classification learning. MSCC adopts the simultaneous optimization process for the clustering and classification learning, and thus does not need to sacrifice the clustering (classification) performance for the classification (clustering) performance. From the experimental results, it can be observed that (1) MSCC can acquire both the promising clustering results and classification results at one time; (2) the Pareto-optimal solutions obtained in MSCC again demonstrate the complementarity between clustering and classification learnings.

In our MSCC, its clustering mechanism is designed by adopting the fuzzy c-means clustering as a reference. However, many other clustering algorithms can also be adopted. For example, when Gaussian finite mixture (GMM) [39] is adopted, the multi-objective functions dependent on both the clustering centers and covariance can be designed. By optimizing both the clustering centers and covariance in these multi-objective functions, the clustering and classification results can also be yielded. Furthermore, since the multiple-optimal solutions yielded by MSCC have diversity, our another work is to employ the diversity to develop an ensemble method [40] to further improve the performance of MSCC.

It is worth mentioning that MSCC is a supervised learning algorithm but extending it to the semi-supervised case is not so straightforward because when the training dataset has unlabeled data, the relation matrix P can not be directly established directly by the formula (5). Undoubtedly, one of future works is to develop a semi-supervised MSCC via different path.

Acknowledgement:

We would like to thank the anonymous referees for their constructive comments and suggestions to greatly improve the presentation of this paper. We thank Dr Thorsten Joachims for the accessible SVM-light software. We also thank Natural Science Foundation of Jiangsu Province under Grant No. BK2008381, National Science Foundation of China under Grant No. 60603029, 60773061 and 60873176, respectively for partial supports.

Reference

[1] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4-37, 2000.

[2] S. E. Schaeffer, "Graph Clustering," *Computer Science Review*, vol. 1, pp. 27-64, 2007.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*: Wiley, 2000.

[4] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to Statistical Learning Theory," *http://www.kyb.mpg.de/~bousquet*.

[5] S. Papadimitriou, S. Mavroudi, L. Vladutu, and A. Bezerianos, "Ischemia Detection with a Self-organizing Map Supplemented by Supervised Learning," *IEEE Trans. Neural Networks*, vol. 12, pp. 503-515, 2001.

[6] W. Pedrycz and G. Vukovich, "Fuzzy Clustering with Supervision," *Pattern Recognition*, vol. 37, pp. 1229–1349, 2004.

[7] H. Timm, "Fuzzy Cluster Analysis of Classified Data," presented at The 9th IFSA World Congress, 2001.

[8] S. Basu, M. Bilenko, and R. Money, "A Probabilistic Framework for Semi-supervised Clustering," presented at In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004.

[9] B. Kulis, S. Basu, I. Dillon, and R. J. Mooney, "Semi-Supervised Grpah Clustering: A Kernel Approach," presented at International Conference on Machine Learning, 2005.

[10] X. J. Zhu, "Semi-supervised learning literature survey," *Technical Report 1530, Computer Sciences, University of Wisconsin-Madison*, 2005.

[11] N. Ye and X. Li, "A Supervised, Incremental Learning Algorithm for Classification Problems," *Comput. Ind. Eng. J.*, vol. 43, pp. 677-692, 2002.

[12] Z. R. Yang, "A Novel Radial Basis Function Neural Network for Discriminant Analysis," *IEEE Trans. Neural Networks*, vol. 17, pp. 604-612, 2006.

[13] I. Maglogiannis, H. Sarimveis, C. T. Kiranoudis, A. A. Chatziioannou, N. Oikonomou, and V. Aidinis,

"Radial Basis Function Neural Networks Classification for the Recognition of Idiopathic Pulmonary Fibrosis in Microscopic Images," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, pp. 42- 54, 2008.

[14] M. Setnes and R. Babuška, "Fuzzy Relational Classifier Trained by Fuzzy Clustering," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 29 pp. 619-625, 1999.

[15] L. Ramirez, N. G. Durdle, D. L. Hill, and V. J. Raso, "Prototypes Stability Analysis in the Design of Fuzzy Classifiers to Assess the Severity of Scoliosis," presented at IEEE Canadian Conference on Electrical and Computer Engineering, 2003.

[16] C. E. Pedreira, "Learning Vector Quantization with Training Data Selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 157-162, 2006.

[17] N. B. Karayiannis and M. M. Randolph-Gips, "Soft Learning Vector Quantization and Clustering Algorithms based on Non-Euclidean Norms: Multinorm Algorithms," *IEEE Trans. Neural Networks*, vol. 14, pp. 89-102, 2003.

[18] S. W. Kim and B. J. Oommen, "Enhancing Prototype Reduction Schemes with LVQ3-type Algorithms," *Pattern Recognition*, vol. 36 pp. 1083-1093, 2003.

[19] W. L. Cai, S. C. Chen, and D. Q. Zhang, "Enhanced Fuzzy Relational Classifier with Representative Training Samples," presented at 2007 International Conference on Wavelet Analysis and Pattern Recognition, 2007.

[20] W. L. Cai, S. C. Chen, and D. Q. Zhang, "Robust Fuzzy Relational Classifier Incorporating the Soft Class Labels," *Pattern Recognition Letters*, vol. 28, pp. 2250-2263, 2007.

[21] W. L. Cai, S. C. Chen, and D. Q. Zhang, "A Simultaneous Learning Framework for Clustering and Classification," *Pattern Recognition, accepted, available at http://dx.doi.org/10.1016/zj.patcog*, 2009.

[22] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling Multiple Objectives with Particle Swarm Optimization " *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 256-279, 2004

[23] D. Q. Zhang and S. C. Chen, "A Novel Kernelized Fuzzy c-means Algorithm with Application in Medical Image Segmentation," *Artificial Intelligence in Medicine*, vol. 32, pp. 37-50, 2004.

[24] J. M. Keller, M. R. Gray, and J. A. Givens, "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, pp. 580-585, 1985.

[25] X. Li and N. Ye, "Grid and Dummy Cluster based Learning of Normal and Intrusive Clusters for Computer Intrusion Detection," *Qual. Reliab. Eng. Int.*, vol. 18 pp. 231-242, 2002.

[26] X. Li and N. Ye, "A Supervised Clustering and Classification Algorithm for Mining Data with Mixed Variables," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 36, pp. 396-406, 2006.

[27] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity Index for Crisp and Fuzzy Clusters " *Pattern Recognition*, vol. 37, pp. 487-501 2004.

[28] D. J. Kim, Y. W. Park, and D. J. Park, "A Novel Validity Index for Determination of the Optimal Number of Clusters," *IEICE Transactions Information and Systems*, vol. E84D, pp. 281, 2001.

[29] K. Deb, S. Agrawal, and A. Pratap, "A Fast and Elitist Multi-objective Genetic Algortihm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182-197, 2002.

[30] V. Cutello, G. Narzisi, and G. Nicosia, "A Class of Pareto Archived Evolution Strategy Algorithms uing Immune Inspired Operators for Ab-Initio Protein Structure," *Lecture Notes in Computer Science*, vol. 3449, pp. 54-63, 2005.

[31] Y. Rahmat-Samii, "Genetic algorithm (GA) and particle swarm optimization (PSO) in engineering electromagnetics," presented at 17th International Conference on Applied Electromagnetics and Communications, 2003.

[32] M. Zubair, M. Choudhry, A. Malik, and I. Qureshi, "Particle swarm optimization assisted multiuser detection along with radial basis function," *IEICE Trans. Commun. E90-B*, vol. 7, pp. 1861-1863, 2007.

[33] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *International Journal of Computer Vision*, vol. 14, pp. 5-24, 1995.

[34] D. R. a. N. A. M. H. Yang, "Learning to Recognize 3D Objects with SNoW," presented at Proceedings of the 6th European Conference on Computer, 2000.

[35] C. Blake, E. Keogh, and C. J. Merz, "UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]," *Department of Information and Computer Science*, 1998.

[36] J. C. Bezdek, *Pattern Recognition in Handbook of Fuzzy Computation*: IOP Publishing Lte., 1998.

[37] S. Abe, "Training of Support Vector Machines with Mahalanobis Kernels," presented at International Conference on Artificial Networks, 2005.

[38] Á. B. Jiménez, J. L. Lázaro, and J. R. Dorronsoro, "Finding Optimal Model Parameters by Discrete Grid Search," *Advances in Soft Computing*, vol. 44 pp. 120-127, 2008.

[39] M. A. T. Figueiredo and A. K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 381-396, 2002.

[40] T. Windeatt, "Accuracy/Diversity and Ensemble MLP Classifier Design," *IEEE Trans. Neural Networks*, vol. 17, pp. 1194-1211 2006.