

# A Multiobjective Simultaneous Learning Framework for Clustering and Classification

Weiling Cai, Songcan Chen, and Daoqiang Zhang

**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 has to sacrifice either the clustering performance or the classification performance. To overcome that problem, in this paper, we present a multiobjective simultaneous learning framework (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.

**Index Terms**—Bayesian theory, classification learning, clustering learning, multiobjective optimization, pattern recognition.

## I. INTRODUCTION

**T**RADITIONAL pattern recognition involves two tasks [14]: 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

[31]. 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 [12]. 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 [5], 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 [12]. Our experimental results (refer to Section IV 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. For example, by utilizing the class information to guide the clustering process, some supervised clustering [26], [28], [33] or semisupervised clustering algorithms [3], [20], [39] 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 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 [6], [7], [16], [19], [23], [27], [30], [32], [36], [37] handle the clustering learning and classification learning in a *sequential* or *independent* manner. As illustrated in Fig. 1, these algorithms first 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 (SCC) [8]. In SCC, the classification criterion and clustering criterion are combined to a *single* objective function by a tradeoff parameter, which 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 aforementioned algorithms usually have

Manuscript received September 19, 2008; revised September 09, 2009; accepted October 03, 2009. First published December 18, 2009; current version published February 05, 2010. This work was supported in part by the National Science Foundation of Jiangsu Higher Education Institutions of China under Grant 09KJB520007, the Natural Science Foundation of Jiangsu Province under Grant BK2008381, and the National Science Foundation of China under Grants 60773061, 60875030, and 60873176.

W. Cai is with the Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China and also with the Department of Computer Science and Engineering, Nanjing Normal University, Nanjing 210097, China (e-mail: caiwl@nuaa.edu.cn).

S. Chen and D. Zhang are with the Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China (e-mail: s.chen@nuaa.edu.cn; dqzhang@nuaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2009.2034741

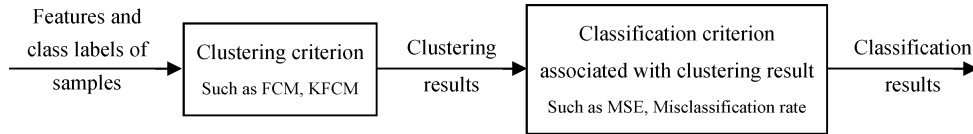


Fig. 1. Sequential optimization for the clustering and classification criteria.

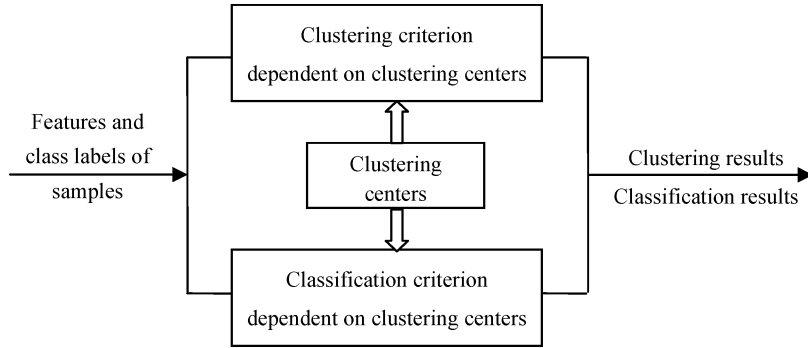


Fig. 2. Simultaneous optimization for the clustering and classification criteria.

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.

To overcome this defect, in this paper, we present a multi-objective simultaneous learning framework (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 intracluster 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's 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 either form a compromise or be more prone to be consistent for their performance improvement. This is our starting point of using multiobjective optimization technique to achieve simultaneous optimality for both. To this end, concretely, we adopt the multiobjective particle swarm optimization (MOPSO) [9] 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 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

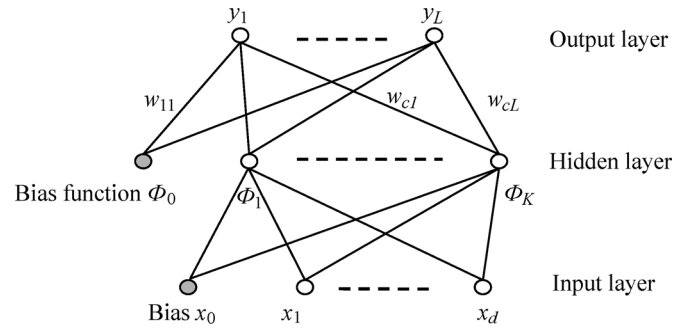


Fig. 3. Architecture of RBFNN.

both clustering compactness and classification error rate on the training data set 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 data sets all demonstrate also the effectiveness and potential of MSCC.

The outline of this paper is as follows. In Section II, we discuss the related work. In Section III, we present the main ideas of the MSCC algorithm. The experimental results are provided in Section IV. We conclude in Section V.

## II. 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) [23], [36], as shown in Fig. 3, is a feedforward multilayer network. It usually consists of three layers: an input layer, a hidden layer, and an output layer. Each basis function  $\Phi_k$  corresponds to a hidden unit and  $w_{kl}$  represents the weight from the  $k$ th basis function or hidden unit to the  $l$ th output units. In the training phase, RBFNN

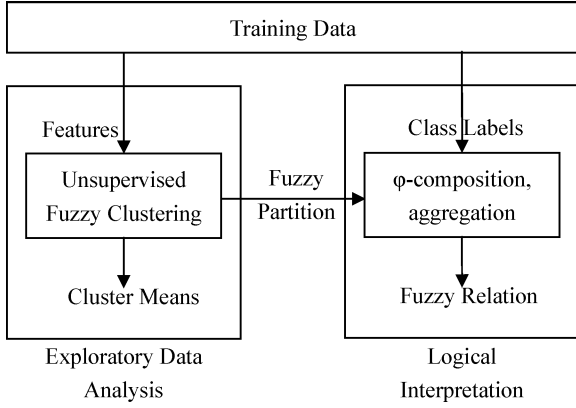


Fig. 4. Training process of FRC and RFRC.

first executes unsupervised clustering process to determine the parameters of the basis function  $\Phi_k$  under the guidance of fuzzy  $c$ -means (FCM) clustering criterion [23]. 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 cannot 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).

Setnes *et al.* proposed fuzzy relational classifier (FRC) [32] 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  $\mathbf{R}$  under the implicit classification criterion to reflect the relationship between clusters and classes.

Lately, in our previous work, we have presented robust FRC (RFRC) [7] 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) [38] to replace FCM; second, employ the soft class label motivated by the fuzzy  $k$ -nearest-neighbor [17] to replace the hard class label. This way, with incorporation of both KFCM and soft class labels, RFRC makes the constructed relation matrix  $\mathbf{R}$  reflect more the relationship between 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 cannot be guaranteed to obtain satisfactory clustering and classification results simultaneously. In addition, the entries in the relation matrix  $\mathbf{R}$  lack the statistical meaning, thus it is difficult to judge whether the obtained relationship is really reliable.

Likewise, Kim and Oommen [19] 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. Similarly to VQ+LVQ3, a supervised clustering and classification algorithm named CCAS [21], [37] and its extended version ECCAS [22] also fall into such a two-step framework. Since both VQ+LVQ3 and CCAS (or ECCAS) adopt the 1NN or the weighted  $k$ -nearest neighbor (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 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 it all 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.

### III. THE PROPOSED METHOD

To obtain the satisfactory clustering and classification result and inspired by our previous work [8], we present an MSCC for both clustering and classification learning. In its implementation, 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 multiobjective framework to formulate the clustering and classification problems. Finally, we adopt MOPSO to simultaneously optimize the clustering centers embedded in these functions.

#### A. 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  $k$ th cluster can be computed

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

where  $\text{dist}$  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

$$f(\mathbf{x}_i) = \arg \max_{1 \leq l \leq L} p(\omega_l|\mathbf{x}_i). \quad (2)$$

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

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

where  $\omega_l$  denotes the  $l$ th class,  $c_k$  represents the  $k$ th 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 (1). Now  $p(\omega_l|c_k)$  can be computed through Bayesian theorem

$$p(\omega_l|c_k) = \frac{p(\omega_l, c_k)}{p(c_k)} \quad (4)$$

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

$$p(\omega_l|c_k) = \frac{\text{Num}(\mathbf{x} \in \omega_l \text{ and } \mathbf{x} \in c_k)}{\text{Num}(\mathbf{x} \in c_k)} \quad (5)$$

For each cluster  $c_k$ , the constraint  $\sum_{l=1}^L p(\omega_l|c_k) = 1$  should be satisfied where  $L$  is the class number. Equation (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  $\mathbf{P}$

$$\mathbf{P} = \begin{bmatrix} p(\omega_1|c_1) & p(\omega_2|c_1) & \dots & p(\omega_L|c_1) \\ p(\omega_1|c_2) & p(\omega_2|c_2) & \dots & p(\omega_L|c_2) \\ \dots & \dots & \dots & \dots \\ p(\omega_1|c_K) & p(\omega_2|c_K) & \dots & p(\omega_L|c_K) \end{bmatrix}. \quad (6)$$

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 data set 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.

## B. Multiobjective Functions for Clustering and Classification

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

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

where  $M$  is the number of objective functions and  $J_m(\{\mathbf{v}_k\})$  is the  $m$ th 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 intraclass compactness and interclass separability, different clustering objective functions can be designed. Here we just introduce three clustering criteria:

1) Xie-Bi index [25] which is presented by

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

2)  $v_{sv}$  index [18] which is proposed by Kim

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

where  $c_k$  is the set of the samples falling into the cluster  $k$  and  $|c_k|$  is the number of samples in  $c_k$ ;

3) in order to introduce the kernel trick to the clustering objective function, we design the intraclass compactness in the feature space

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

where  $\Phi$  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), equation (10) can be rewritten

$$\begin{aligned} J_m(\{\mathbf{v}_k\}) &= \sum_{k=1}^K \sum_{i=1}^N u_{ik}^m(\mathbf{v}_k) \\ &\quad \times (K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{v}_k, \mathbf{v}_k) - 2K(\mathbf{x}_i, \mathbf{v}_k)). \end{aligned} \quad (11)$$

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

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

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 the centers in the feature space. The final objective function can be written as

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

Second, based on the classification mechanism designed in Section III-A, the different classification objective functions can be designed. Here we just list the two classification criteria:

- 1) minimization of the misclassification rate

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

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

- 2) minimization of a squared error between the target outputs and the actual outputs

$$J_m(\{\mathbf{v}_k\}) = \sum_{i=1}^N \sum_{j=1}^L (p(\omega_j | \mathbf{x}_i) - y_{ij})^2 \quad (15)$$

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  $l$ th class. Here  $y_i$  is represented by one-of- $c$  coding. For example, if there are four classes in the given data set 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

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

where  $J_1(\{\mathbf{v}_k\})$  is the misclassification rate and  $J_2(\{\mathbf{v}_k\})$  measures the compactness in the feature space. Equation (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.

### C. Optimization of Multiobjective Functions

To describe the concept of optimality in the multiobjective functions, we will introduce a few definitions [9] involved in multiobjective optimization.

**Definition 1 (Dominance):** For a given multiobjective problem  $\min J(\mathbf{x}) = [J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_M(\mathbf{x})]$ , the so-

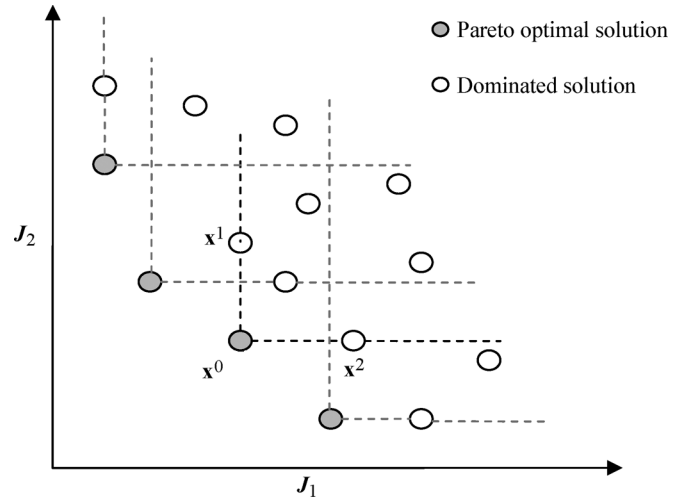


Fig. 5. Pareto front of a set of solutions in a two objective space.

lution  $\mathbf{x}^1$  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)  $\forall i \in [1, 2, \dots, M], J_i(\mathbf{x}^1) \leq J_i(\mathbf{x}^2)$ ; and 2)  $\exists i \in [1, 2, \dots, M], J_i(\mathbf{x}^1) < J_i(\mathbf{x}^2)$ .

**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  $P_s = \{\mathbf{x}^0 | \nexists \mathbf{x}^1 \prec \mathbf{x}^0\}$ .

**Definition 4 (Pareto Front):** For a given Pareto optimal set  $P_s$ , the Pareto front is defined as  $P_F = \{J(\mathbf{x}) = (J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_M(\mathbf{x})) | \mathbf{x} \in P_s\}$ .

To explain the above concepts clearly, we give Fig. 5 under the condition of two objective functions. The empty circle denotes a dominated solution and the filled circle represents a Pareto-optimal solution which is also termed *nondominated* solution. According to Definition 1, the solution  $\mathbf{x}^0$  dominates  $\mathbf{x}^1$  and  $\mathbf{x}^2$ ; the solution denoted by the empty circle is Pareto-optimal in terms of Definition 2; all the filled circles 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 filled circles.

Next, we employ the above concepts to briefly discuss the existing optimization methods for multiobjective problems. Classical methods suggest converting the multiobjective optimization problem to a single-objective optimization problem by objective weighting. By introducing a weight parameter  $\beta$ , the optimization for the multiobjective functions in (16) can be transformed to

$$\min J(\{\mathbf{v}_k\}) = J_1(\{\mathbf{v}_k\}) + \beta J_2(\{\mathbf{v}_k\}). \quad (17)$$

By optimizing (17) instead of (16), a single Pareto-optimal solution (i.e., clustering centers) that makes a balance between the clustering performance and the classification performance can be obtained. However, this single point solution is usually sensitive to the weight  $\beta$  [11]. As a result, in order to get a solution as optimal as possible, multiple sets of different weights have to be used, leading to the same problem being solved many times.

In recent years, a number of multiobjective evolutionary algorithms (MOEA) [9]–[11] have been suggested such as non-dominated sorting genetic algorithm (NSGA) [11] and Pareto archive evolutionary strategy (PAES) [10]. 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 multiobjective search and optimization might be a problem area where evolutionary algorithms (EAs) do better than other blind search strategies [10], [11]. In 2004, Coello [9] *et al.* proposed an MOPSO and proved its good performance and high speed of convergence. MOPSO is an evolutionary technique through individual improvement plus population cooperation and competition. Many works [29], [40] have shown that PSO-type methods are the prevailing population-based optimization algorithms and they have been successfully applied to 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 nondominated solutions found along the search process. In its implementation, MOPSO employs this repository to guide the flight of the current particles and store the nondominated solutions.

In this paper, we adopt the simplified version of MOPSO to solve the multiobjective optimization of the 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 the 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  $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{iD}]$  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 multidimensional research spaces with a velocity  $\mathbf{vel}_i = [\text{vel}_{i1}, \text{vel}_{i2}, \dots, \text{vel}_{id}, \dots, \text{vel}_{iD}]$ . This velocity is updated by the experience of particle itself and repository

$$\text{vel}_{id}^{t+1} = w \times \text{vel}_{id}^t + r_1 \times (\text{pbset}_{id}^t - x_{id}^t) + r_2 \times (\text{Repository}_d(h) - x_{id}^t) \quad (18)$$

where  $t$  is the current iteration number,  $w$  is inertia weight and set to 0.4, and  $r_1$  and  $r_2$  are two independent random numbers uniformly distributed in the range of  $[0, 1]$ .  $\text{pbset}_i = [\text{pbset}_{i1}, \text{pbset}_{i2}, \dots, \text{pbset}_{iD}]$  represents the best position that the  $i$ th particle has had.  $\text{Repository}(h) = [\text{Repository}_{h1}, \text{Repository}_{h2}, \dots, \text{Repository}_{hD}]$  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

$$x_{id}(t+1) = x_{id}(t) + \text{vel}_{id}(t+1). \quad (19)$$

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

---

#### MSCC Learning Algorithm

---

Step 1: 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.

Step 2: Evaluate the two objective values of all particles according to (13) and (14) and set  $\text{pbset}_i$  of each particle equal to its current position.

Step 3: Store the positions of the particles that represent nondominated solutions in the repository.

Step 4: While  $I > t$ :

(a) Compute the speed of each particle by (18).

(b) Compute the new position  $\mathbf{x}_i$  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 nondominated locations (the nondominated 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

$x_m$  is the currently nondominated location.

End

End

(e) Insert all the currently nondominated locations into

**Repository**;

Eliminate any dominated locations from the **Repository**.

(f) If the current position  $\mathbf{x}_i(t)$  of the particle dominates  $\text{pbset}_i$

$\text{pbset}_i = \mathbf{x}_i(t)$

else if the  $\text{pbset}_i$  dominates  $\mathbf{x}_i(t)$

$\text{pbset}_i$  is kept

else if neither of them is dominated by the other


















$\text{pbset}_i$  is updated or kept randomly

End

(g) Update  $t = t + 1$ .

---

TABLE I  
CLUSTERING RESULTS OF THE TEST SAMPLES ON THE COIL DATA SET

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

#### D. Time Complexity Analysis of MSCC

The time complexity of MSCC is  $O(I \times P \times \max(K \times d, P \times M, N \times K \times L))$  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 they are set to the constant values 500, 100, and 2, respectively. Moreover,  $K$ ,  $d$ ,  $N$ , and  $L$  are the variable parameters dependent on the chosen data set. It is worth pointing out that the larger the cluster number (or the data dimension, the sample number, the class number), the more the computational time there is.

#### E. A Toy Illustration for MSCC Benefit

Here we give a toy illustration on the data set Coil [24] to explain why simultaneous classification and clustering learnings can give more than just either classification learning or clustering learning.<sup>1</sup> The full Coil data set consists of images of 100 objects where the images of the objects were taken at pose intervals of  $5^\circ$ , i.e., 72 poses per object. In this paper, we have used a

<sup>1</sup>Coil is available at <http://www.cs.columbia.edu/CAVE>

TABLE II  
CLASSIFICATION RESULTS AND THE PARAMETERS ON COIL DATA SET

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

part of the Coil database by involving only the first two objects, with 144 images in total. The training set consists of 36 images (one for every  $10^\circ$ ) for each object, and the test set consists of the remaining 36 images for each object [35]. For such data set, 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 the structural information to not only classify the objects to different classes, but also discover the objects with similar poses. As shown in Table I, the objects grouped to the same clusters have very similar poses, which indicates that the structure hidden in the data is discovered. Moreover, the relation matrix in Table II means that the objects falling into clusters  $c_1$ ,  $c_2$ ,

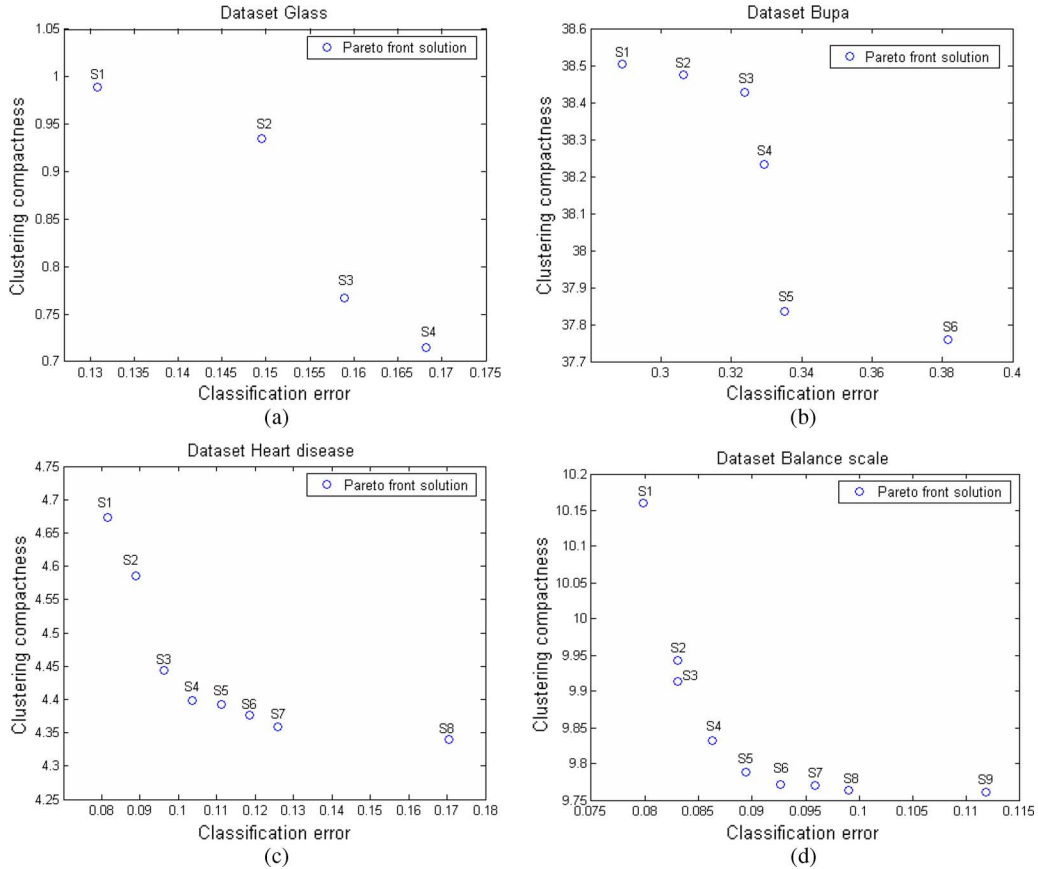


Fig. 6. Pareto fronts obtained on the data sets *Glass*, *Bupa*, *Heart\_disease*, and *Balance\_scale*, respectively.

TABLE III  
MISCLASSIFIED RATE AND CLUSTERING COMPACTNESS ON THE TRAINING AND TEST DATA SET 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	<b>0.2897</b>	4.1861
S3	0.1589	0.7668	0.3178	<b>4.0213</b>
S4	0.1682	0.7146	0.4206	4.0632

$c_3$ , and  $c_4$  belong to class  $\omega_1$ , and similarly, the objects in clusters  $c_5$ ,  $c_6$ ,  $c_7$ , and  $c_8$  belong to class  $\omega_2$ . Due to the correct clustering and so-generated relationship matrix  $\mathbf{P}$ , MSCC achieves the classification accuracy of 100%. From this example, we can see that MSCC discovers both structures hidden in the data and the relationship between the structures and their classes, which makes Coil data set prone to be transparent and interpretable. However, SVM has difficulty to great extent to *simultaneously* achieve the two aspects.

#### IV. EXPERIMENTAL RESULTS

##### A. Pareto Optimal Solution

To investigate the property of the Pareto-optimal solutions, we give the Pareto-optimal front on the data sets *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 data sets, respectively. This result implies that a *solution inconsistency* can in general occur in this multiple objective problem [9].

Furthermore, Tables III–VI list the  $J_1$  values and  $J_2$  values of the Pareto-optimal solutions on both training data and test data of *Glass*, *Bupa*, *Heart\_disease*, and *Balance\_scale*, respectively. From these tables, we can find that on the test data sets, S2, S2, S7, and S2 in each table obtain the best classification performance, and S3, S3, S5, and S8 achieve the best clustering performance. From this result, we have an interesting observation that the best performance of clustering or classification on the test data sets corresponds to those solutions which achieve relatively low values of the objectives of both clustering compactness and classification error rate on the training data sets. This conclusion empirically demonstrates the consistency or complementarity between the clustering and classification learnings. In

TABLE IV  
MISCLASSIFIED RATE AND CLUSTERING COMPACTNESS ON THE TRAINING AND TEST DATA SET 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	<b>0.3372</b>	44.1799
S3	0.3237	38.4271	0.3605	<b>43.6214</b>
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 V  
MISCLASSIFIED RATE AND CLUSTERING COMPACTNESS ON THE TRAINING AND TEST DATA SET 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	<b>5.6571</b>
S6	0.1185	4.3762	0.2074	5.7172
S7	0.1259	4.3589	<b>0.1852</b>	5.9531
S8	0.1704	4.3402	0.2593	5.9170

TABLE VI  
MISCLASSIFIED RATE AND CLUSTERING COMPACTNESS ON THE TRAINING AND TEST DATA SET 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	<b>0.0962</b>	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	<b>10.0047</b>
S9	0.1118	9.7616	0.1346	10.0708

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.

### B. Synthetic Data Set

We apply RBFNN, RFRC, VQ+LVQ3, SCC, and MSCC on a synthetic data set in Table VII to compare both their classification and clustering abilities. Here the number  $K$  of cluster centers is set to 5 and the scale factor  $\lambda$  of the RBF kernel is 1. To evaluate their clustering effectiveness, we list the obtained

TABLE VII  
SYNTHETIC DATA SET WITH THREE CLASSES IN FIVE GROUPS

Group	Class label	Group center	Variance
Gaussian Distribution 1	$\omega_1$	(6, 12)	(1, 0.5)
Gaussian Distribution 2	$\omega_1$	(0, 5)	(2, 1)
Gaussian Distribution 3	$\omega_2$	(3, 12)	(2, 1)
Gaussian Distribution 4	$\omega_2$	(8, 5)	(1, 0.5)
Gaussian Distribution 5	$\omega_3$	(4, -2)	(2, 1)

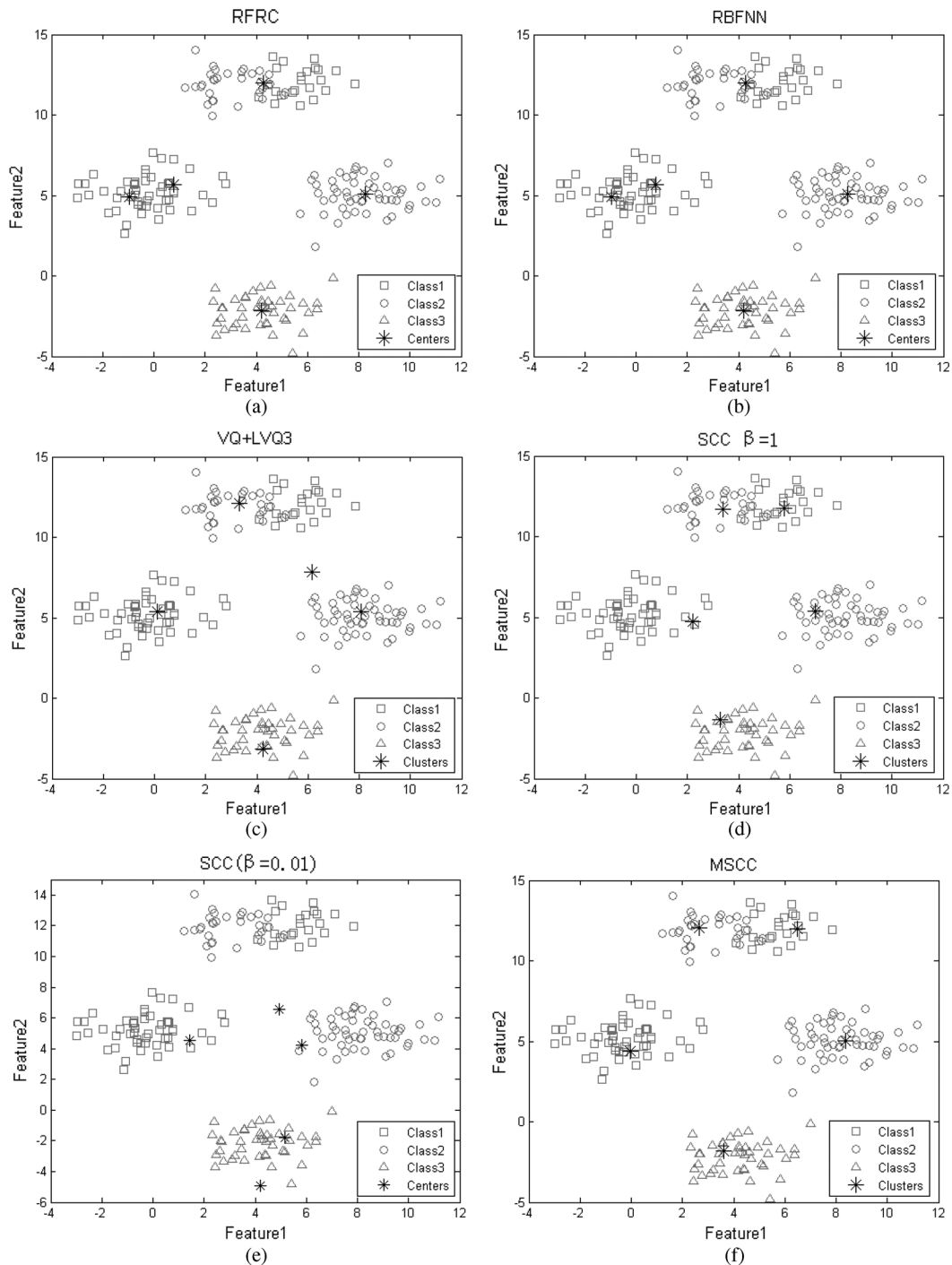


Fig. 7. Cluster centers obtained by RFRC, RBFNN, VQ+LVQ3, SCC, and MSCC, respectively.

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., Classes 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  $\beta$ , the correct clustering result is obtained as shown in Fig. 7(d); however, when an improper value is selected, the obtained clustering result is un-

able 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, thus leading to the same problem having to be solved many times. In contrast, MSCC removes the weight parameter  $\beta$  and obtains the correct clustering centers located in the proper places, and thus reflects the inherent structure in this data relatively correctly.

To further compare the classification effectiveness, we present the relation matrices (connecting weights) and classification accuracy in Table VIII. From this table, we can make the following analyses. 1) The connecting weights in RBFNN are

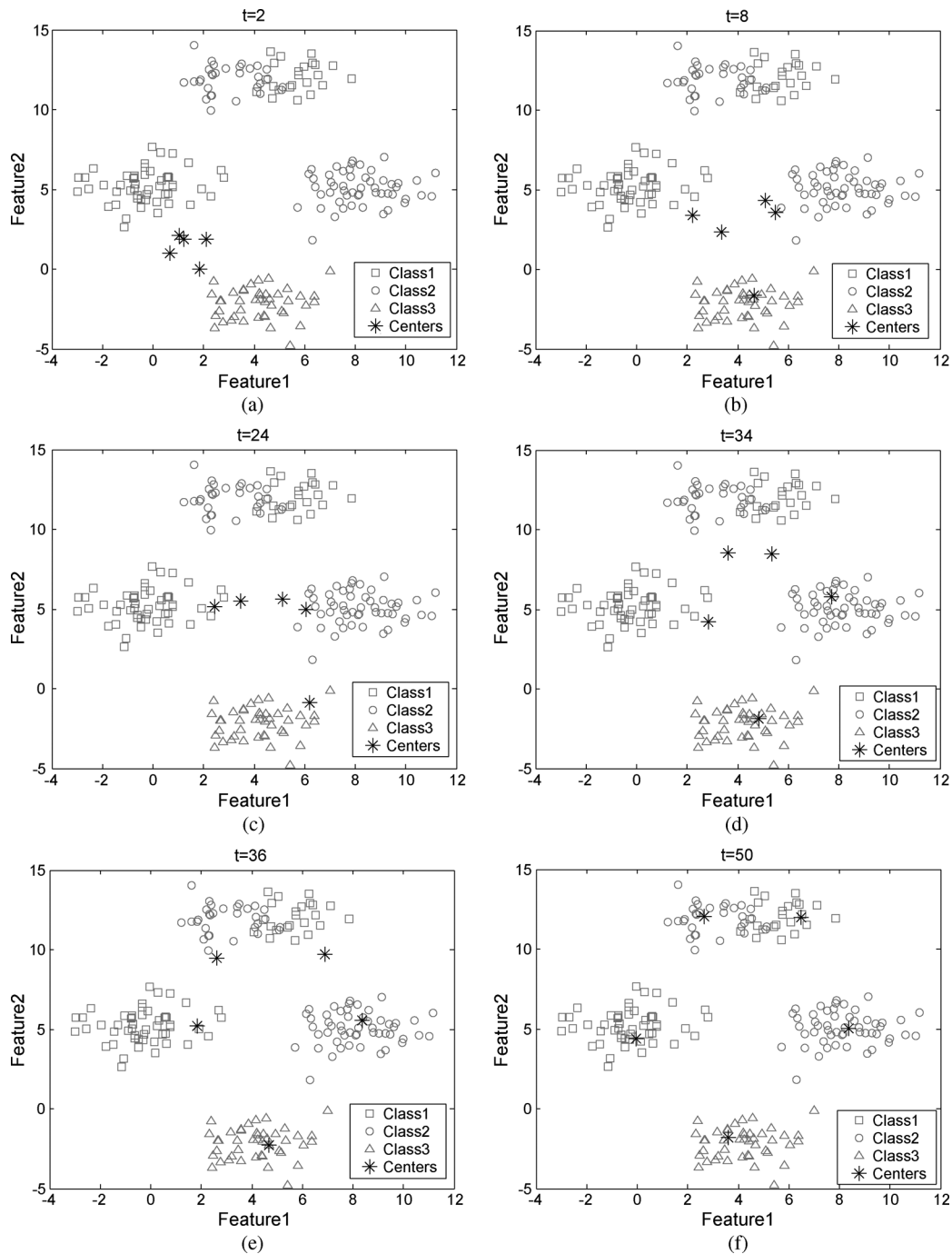


Fig. 8. Iterative process of MSCC.

TABLE VIII  
PARAMETER COMPARISON AMONG RFRC, RBFNN, VQ+LVQ3, AND MSCC

Parameters	RBFNN	RFRC	VQ+LVQ3	SCC ( $\beta=0.01$ )	SCC ( $\beta=1$ )	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%

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 is obtained by the composite operators, and thus it lacks the statistical meaning. In

TABLE IX  
PARAMETERS OF MSCC AT DIFFERENT ITERATION STEP

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

VQ+LVQ3, its relation matrix is determined by the class labels of clustering centers, and such hard values cannot quantitatively reflect the fuzzy belonging degree between clusters and classes. In SCC, the *larger* the  $\beta$  value is, the more attention the objective function pays to the classification problem; the *smaller* the  $\beta$  value is, the more attention the objective function pays to the clustering problem. As a result, a proper value should be selected for  $\beta$  so that a balance can be created between the classification and clustering performances, and thus the correct result can be obtained as shown in Table VIII ( $\beta = 1$ ). In MSCC, the relation matrix can not only reveal the underlying logical relationship in data but also a 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  $\beta$  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 than the other algorithms. Such good performance can be attributed 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, and thus does not need to sacrifice the clustering performance for the classification performance, or *vice versa*.

To make the iterative process of MSCC clearer, we give the intermediate results of the clustering centers in Fig. 8, and their corresponding relation matrix and classification accuracy in Table IX. 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 the data. Moreover, from Table IX, 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 the classes, and the corresponding classification accuracy increases from 86.7% to 99.0%.

### C. Real-Life Data Set

We evaluate the classification capability of MSCC on real-life data sets. We select 20 data sets from the University of California at Irvine (UCI) Machine Learning Repository [4], 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),<sup>2</sup> 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 [2] 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  $\lambda$  is determined by searching in  $\{0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15\}$ . In a support vector machine (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  $\beta$  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 [1] associated with the classification accuracy. Due to the multiple parameters existing in these algorithms, the discrete grid search [15] based on an 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  $\lambda$  used in the experiments in Table X.

In all of our experiments, each data set is randomly partitioned into two halves: one half is used for training and the other for testing. This process runs repeatedly and independently ten times, and only their averaged accuracies and the corresponding standard deviations are reported in Table XI.

First, we compare the classification results yielded, respectively, by SCC and MSCC. It can be seen from the table that on all of the data sets, the accuracies of MSCC are, respectively, better than those of SCC. Especially, on the data sets *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 multiobjective form and diversity of the multiple Pareto-optimal solutions. In comparison with SCC, MSCC has two advantages: 1) by utilizing the multiobjective functions to describe the clustering and classification problems, respectively,

<sup>2</sup>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 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 X  
NUMBER OF THE CLUSTERING CENTERS AND THE SCALE FACTOR OF RBF KERNEL USED IN ALGORITHMS

Dataset (#samples×#dim×#class)	RFRC		VQ+LVQ3	RBFNN		SVM	CBSVM		SCC		MSCC	
	$K$	$\lambda$	$K$	$K$	$\lambda$	$\lambda$	$K$	$\lambda$	$K$	$\lambda$	$K$	$\lambda$
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

MSCC can remove the weighting parameter in SCC, and thus the computational burden for choosing this parameter can be exempted; and 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 are just 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 data sets. Compared to RBFNN, it yields better performance on 17 data sets, comparable performance on two data sets, and worse performance on one data set. The excellent classification performance of MSCC comes from its effective learning mechanism.

Finally, to give a baseline reference, we make comparison against the state-of-the-art classifier SVM and our purposely

designed algorithm CBSVM. It is worth pointing out that CBSVM is superior to SVM in a classification ability mainly due to the incorporation of the clustering information into CBSVM, which states that combining clustering and SVM (like the algorithms introduced in Section II) 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 12 data sets, and further compared to CBSVM, MSCC possesses higher accuracy on 12 data sets and comparable accuracy on the other four data sets, 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 the clustering result can be simultaneously obtained; and 2) the class posterior probabilities computed in this framework can reflect confidence of the classification decision, which is important for reliable and interpretable classification.

TABLE XI  
CLASSIFICATION ACCURACY COMPARISON ON REAL-LIFE DATA SETS

Dataset (#samples×#dim×#class)	RFRC	VQ+LVQ3	RBFNN	SVM	<b>CBSVM</b>	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	<b>97.6 ± 0.6</b>
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	<b>99.7 ± 0.7</b>
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	<b>96.4 ± 1.6</b>
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	<b>58.1 ± 4.9</b>
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	<b>77.0 ± 2.5</b>
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	<b>100 ± 0.0</b>
WDBC (569×30×2)	92.0 ± 1.6	96.4 ± 0.9	95.0 ± 1.2	<b>97.2 ± 0.7</b>	<b>97.4 ± 0.8</b>	96.8 ± 0.7	<b>97.3 ± 0.7</b>
Waveform (5000×21×3)	83.0 ± 0.5	85.1 ± 0.6	<b>86.5 ± 0.9</b>	86.2 ± 0.4	86.0 ± 0.3	86.2 ± 0.6	<b>86.5 ± 0.3</b>
Balance_scale (625×4×3)	84.7 ± 1.5	86.0 ± 1.8	90.5 ± 1.0	90.5 ± 1.0	<b>93.3 ± 1.2</b>	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	<b>84.2 ± 1.8</b>
Pima_Indian_diabetes (768×8×2)	70.7 ± 3.2	72.6 ± 2.0	74.2 ± 2.3	76.3 ± 2.0	<b>77.0 ± 1.4</b>	76.0 ± 1.4	76.5 ± 1.1
Glass (214×9×6)	63.8 ± 3.8	63.2 ± 3.6	65.0 ± 3.8	<b>68.5 ± 3.5</b>	67.2 ± 3.0	64.9 ± 2.5	<b>68.9 ± 2.5</b>
Sonar (208×60×2)	77.5 ± 3.9	73.9 ± 2.8	80.2 ± 3.0	<b>85.4 ± 3.3</b>	<b>85.4 ± 4.1</b>	80.8 ± 5.1	<b>85.6 ± 4.1</b>
Wine (178×13×3)	96.0 ± 1.7	96.5 ± 1.5	97.3 ± 1.1	<b>98.4 ± 1.1</b>	<b>98.1 ± 1.0</b>	97.1 ± 1.8	<b>98.3 ± 1.3</b>
Ecoli (336×7×8)	81.8 ± 3.3	78.8 ± 3.0	<b>85.2 ± 2.7</b>	<b>85.0 ± 1.7</b>	<b>85.1 ± 1.8</b>	83.7 ± 1.8	<b>85.0 ± 2.4</b>
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	<b>86.7 ± 11.9</b>
Iris (150×4×3)	95.3 ± 1.1	94.7 ± 1.9	96.4 ± 1.6	95.9 ± 1.5	95.6 ± 1.7	95.2 ± 1.4	<b>97.1 ± 1.7</b>
Bupa (345×6×2)	61.0 ± 2.4	62.1 ± 3.7	<b>70.8 ± 3.6</b>	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	<b>92.2 ± 0.6</b>
Spambase (4601×57×2)	85.1 ± 1.1	88.5 ± 0.7	80.7 ± 1.0	89.2 ± 0.6	<b>90.4 ± 0.5</b>	88.1 ± 1.3	89.9 ± 0.8

## V. 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 limits the effectiveness of both clustering and classification learnings to a great extent. Differently from these algorithms, in this paper, an MSCC is presented for simultaneous clustering and classification learnings. MSCC adopts the simultaneous optimization process for the clustering and classification learnings, 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; and 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) [13] is adopted, the multiobjective functions dependent on both clustering centers and covariance can be designed. By optimizing both clustering centers and covariance in these multiobjective functions, the clustering and classification results can also be yielded. Furthermore, since the multiobjective solutions yielded by MSCC have diversity, our additional work is to employ the diversity to develop an ensemble method [34] to further improve the performance of MSCC.

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

## ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their constructive comments and suggestions that greatly improved the presentation of this paper. They would also like to thank Dr. T. Joachims for the accessible SVM-light software.

## REFERENCES

- [1] S. Abe, "Training of support vector machines with Mahalanobis kernels," presented at the Int. Conf. Artif. Netw., Warsaw, Poland, Sep. 11–15, 2005.
- [2] J. C. Bezdek, *Pattern Recognition in Handbook of Fuzzy Computation*. Boston, MA: IOP, 1998.
- [3] S. Basu, M. Bilenko, and R. Money, "A probabilistic framework for semi-supervised clustering," presented at the ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, Seattle, WA, Aug. 22–25, 2004.
- [4] C. Blake, E. Keogh, and C. J. Merz, UCI Repository of Machine Learning Databases, Dept. Inf. Comput. Sci., Univ. California Irvine, Irvine, CA, 1998 [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [5] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to statistical learning theory," 2004 [Online]. Available: <http://www.kyb.mpg.de/~bousquet>
- [6] W. L. Cai, S. C. Chen, and D. Q. Zhang, "Enhanced fuzzy relational classifier with representative training samples," presented at the Int. Conf. Wavelet Anal. Pattern Recognit. Beijing, China, Nov. 2–4, 2007.
- [7] W. L. Cai, S. C. Chen, and D. Q. Zhang, "Robust fuzzy relational classifier incorporating the soft class labels," *Pattern Recognit. Lett.*, vol. 28, pp. 2250–2263, 2007.
- [8] W. L. Cai, S. C. Chen, and D. Q. Zhang, "A simultaneous learning framework for clustering and classification," *Pattern Recognit.* 2009 [Online]. Available: <http://www.dx.doi.org/10.1016/j.patcog>, accepted for publication
- [9] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.
- [10] V. Cutello, G. Narzisi, and G. Nicosia, "A class of Pareto archived evolution strategy algorithms using immune inspired operators for ab-initio protein structure," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2005, vol. 3449, pp. 54–63.
- [11] K. Deb, S. Agrawal, and A. Pratap, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2000.
- [13] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.
- [14] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.
- [15] Á. B. Jiménez, J. L. Lázaro, and J. R. Dorronsoro, "Finding optimal model parameters by discrete grid search," *Adv. Soft Comput.*, vol. 44, pp. 120–127, 2008.
- [16] 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 Netw.*, vol. 14, no. 1, pp. 89–102, Jan. 2003.
- [17] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 4, pp. 580–585, Aug. 1985.
- [18] D. J. Kim, Y. W. Park, and D. J. Park, "A novel validity index for determination of the optimal number of clusters," *IEICE Trans. Inf. Syst.*, vol. E84D, pp. 281–285, 2001.
- [19] S. W. Kim and B. J. Oommen, "Enhancing prototype reduction schemes with LVQ3-type algorithms," *Pattern Recognit.*, vol. 36, pp. 1083–1093, 2003.
- [20] B. Kulis, S. Basu, I. Dillon, and R. J. Mooney, "Semi-supervised Graph clustering: A kernel approach," presented at the Int. Conf. Mach. Learn., Bonn, Germany, Aug. 7–11, 2005.
- [21] 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.
- [22] X. Li and N. Ye, "A supervised clustering and classification algorithm for mining data with mixed variables," *IEEE Trans. Syst. Man Cybern. A, Syst. Humans*, vol. 36, no. 2, pp. 396–406, Mar. 2006.
- [23] 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 Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 42–54, Jan. 2008.
- [24] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comput. Vis.*, vol. 14, pp. 5–24, 1995.
- [25] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity index for crisp and fuzzy clusters," *Pattern Recognit.*, vol. 37, pp. 487–501, 2004.
- [26] S. Papadimitriou, S. Mavroudi, L. Vladutu, and A. Bezerianos, "Ischemia detection with a self-organizing map supplemented by supervised learning," *IEEE Trans. Neural Netw.*, vol. 12, no. 3, pp. 503–515, May 2001.
- [27] C. E. Pedreira, "Learning vector quantization with training data selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 157–162, Jan. 2006.
- [28] W. Pedrycz and G. Vukovich, "Fuzzy clustering with supervision," *Pattern Recognit.*, vol. 37, pp. 1229–1349, 2004.
- [29] Y. Rahmat-Samii, "Genetic algorithm (GA) and particle swarm optimization (PSO) in engineering electromagnetics," presented at the 17th Int. Conf. Appl. Electromagn. Commun., Dubrovnik, Croatia, Oct. 1–3, 2003.
- [30] 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 the IEEE Can. Conf. Electr. Comput. Eng., Montreal, QC, Canada, May 4–7, 2003.
- [31] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, pp. 27–64, 2007.
- [32] M. Setnes and R. Babuška, "Fuzzy relational classifier trained by fuzzy clustering," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 29, no. 5, pp. 619–625, Oct. 1999.
- [33] H. Timm, "Fuzzy cluster analysis of classified data," presented at the 9th Int. Fuzzy Syst. Assoc. World Congr., Vancouver, BC, Canada, Jul. 25–28, 2001.
- [34] T. Windeatt, "Accuracy/diversity and ensemble MLP classifier design," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1194–1211, Sep. 2006.
- [35] M. H. Yang, D. Roth, and N. Ahuja, "Learning to recognize 3D objects with SNoW," presented at the 6th Eur. Conf. Comput., Dublin, Ireland, 2000.
- [36] Z. R. Yang, "A novel radial basis function neural network for discriminant analysis," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 604–612, May 2006.
- [37] N. Ye and X. Li, "A supervised, incremental learning algorithm for classification problems," *Comput. Ind. Eng. J.*, vol. 43, pp. 677–692, 2002.
- [38] D. Q. Zhang and S. C. Chen, "A novel kernelized fuzzy c-means algorithm with application in medical image segmentation," *Artif. Intell. Med.*, vol. 32, pp. 37–50, 2004.
- [39] X. J. Zhu, "Semi-supervised learning literature survey," *Comput. Sci.*, Univ. Wisconsin-Madison, Madison, WI, Tech. Rep. 1530, 2005.
- [40] 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.

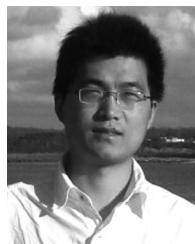


**Weiling Cai** received the B.Sc. and Ph.D. degrees in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003 and 2008, respectively. Currently, she is a Lecturer at the Department of Computer Science and Engineering, Nanjing Normal University, Nanjing, China. Her research interests focus on machine learning and pattern recognition.



**Songcan Chen** received the B.Sc. degree in mathematics from Hangzhou University (now merged into Zhejiang University), Nanjing, China, in 1983, the M.Sc. degree in computer applications from Shanghai Jiaotong University, Shanghai, China in 1985, and the Ph.D. degree in communication and information systems from Nanjing University of Aeronautics & Astronautics, Nanjing, China, in 1997.

He then worked at Nanjing University of Aeronautics & Astronautics (NUAA), Nanjing, China, in January 1986 as an Assistant Lecturer. Since 1998, he has been with the Department of Computer Science and Engineering, NUAA as a full Professor. His research interests include pattern recognition, machine learning, and neural computing. In these fields, he has authored or coauthored over 130 scientific journal papers.



**Daoqiang Zhang** received the B.Sc. and Ph.D. degrees in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1999 and 2004, respectively.

From 2004 to 2006, he was a Postdoctoral Fellow at the Department of Computer Science and Technology, Nanjing University, Nanjing, China. He joined the Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, as a Lecturer in 2004, and is currently a Professor. His research interests include machine learning, pattern recognition, data mining, and image processing. In these areas he has published over 40 technical papers in refereed international journals or conference proceedings.

Dr. Zhang was nominated for the National Excellent Doctoral Dissertation Award of China in 2006, and won the best paper award at the 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI'06). He has served as a program committee member for several international and native conferences. He is a member of the Chinese Association of Artificial Intelligence (CAAI) Machine Learning Society.