Information Sciences 232 (2013) 116–129

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Soft large margin clustering

Yunyun Wang [a,b,c,d], Songcan Chen [a,*]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China
[b] College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China
[c] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China
[d] Key Laboratory of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education Jiangsu Province, Nanjing, Jiangsu 210003, China

ARTICLE INFO

Article history:
Received 29 April 2011
Received in revised form 14 December 2012
Accepted 24 December 2012
Available online 8 January 2013

Keywords:
Clustering
Large margin
Soft clustering
Soft clustering membership

ABSTRACT

Motivated by the successes of large margin principle in classification learning, the maximum margin clustering method (MMC) received intensive attention recently. It seeks a decision function and cluster labels for data simultaneously such that a supervised SVM trained on the label-assigned data could achieve the maximum margin. MMC *assigns a unique cluster label* for each instance. However, in real applications, the data distributions from different clusters are usually overlapped, and thus an instance might belong to multiple clusters with certain probabilities. Several soft clustering methods, which make use of *soft membership assignment*, have been developed in literature and lead to better data partition than their label-assignment counterparts. It motivates us to develop a novel Soft Large Margin Clustering (SLMC for short hereafter) method. SLMC enjoys the advantages of both MMC and the soft clustering methods, i.e., on one hand, it possesses a decision function with the maximal margin between clusters, and on the other hand, it accomplishes soft assignments for each instance to individual clusters to capture the nature of data structure. Its algorithmic implementation follows an alternating iterative strategy, in which each step in the iteration generates a closed-form solution, and the convergence of the whole iteration process can be theoretically guaranteed. Experiments on both synthetic and real datasets verify the effectiveness of SLMC.

1. Introduction

Clustering is an unsupervised learning process, which aims to partition data into a set of non-overlapping groups such that data within the same group are as similar as possible, whereas data belonging to different groups are as dissimilar as possible [17,25]. In the past decades, various clustering methods have been developed in literature, typically including *K*-means [12], spectral clustering [24] and mixture model [23], and they have been widely applied in many areas such as data mining, image segmentation and image compression.

Recently, motivated by the superiority of large margin principle in classification learning [7,10,16], a novel large margin clustering method named maximum margin clustering (MMC) [38] has been developed and received intensive attention in machine learning community. MMC aims to seek the decision function and cluster labels for given data simultaneously so that a supervised SVM trained on given data with the assigned labels would achieve the maximum margin. The optimization problem formulated in MMC is a non-convex integer programming problem, and then relaxed to a corresponding semi-definite programming (SDP) [6] problem with n^2 variables (for a given dataset containing n instances). Finally, it can

* Corresponding author. Tel.: +86 25 84896481x12221; fax: +86 25 84498069.
E-mail addresses: wangyunyun@njupt.edu.cn (Y. Wang), s.chen@nuaa.edu.cn (S. Chen).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.12.040

be solved by the commonly-used CSDP [5] or SeDumi [33] toolboxes with a high computational complexity of $O(n^6)$. Later, several researchers proposed several variants of MMC [20,35–36,41–42] to reduce the high computation complexity. Valizadegan et al. [35] developed a generalized maximum margin clustering method (GMMC) by reducing the variable number to n, achieving a lower computation complexity of $O(n^{4.5})$. Li et al. [20] developed a label generation-maximum margin clustering (LG-MMC) by converting the optimization problem of MMC into a multiple kernel learning problem [29–30], achieving a computational complexity proportional to that of SVM, i.e., $O(k\times n^3)$, where k is the iteration number. Zhang et al. [42] developed an iterative SVR/LS-SVM by replacing the hinge loss in MMC with the Laplacian/squared loss, reducing the complexity to $O(k\times n^3)$, the same order with that of SVR/LS-SVM [31,34]. Wang et al. [36] developed a cutting plane maximum margin clustering approach (CPMMC) by first decomposing the MMC optimization problem into a set of convex sub-problems using the constraint concave-convex program (CCCP) [32,40], and then solving each sub-problem by the cutting plane method. Finally, CPMMC achieves a linear-time complexity, i.e., $O(k\times s\times n)$, where s is the number of non-zero elements in the given data matrix.

In MMC and its variants, each instance is assigned to just one cluster. However, such a cluster label assignment could be inadequate when the data distributions are overlapping. For example, some instances might be equally distant from two or more clusters, and consequently belong to multiple clusters with different memberships [13,25]. Moreover, a cluster label assignment can not truly reflect the different degrees of an instance belonging to individual clusters, since it assigns the instance to a single cluster as long as the instance is more similar to the cluster than the others. In fact, researchers have considered this issue, and attempted to improve the label-assignment clustering methods by using the soft membership assignment. Finally they developed a series of soft clustering methods [19,22,26], for instance, the most well-known fuzzy c-means (FCM) [4,26] clustering from K-means (KM) clustering [21]. Such soft (or soft-membership-assignment) clustering methods can reflect cluster structure in a more natural way, and indeed provide better and more meaningful data partition than the corresponding label-assignment ones [11].

In this paper, aiming to exert the benefits of soft clustering in large margin clustering, we develop a novel clustering method referred to as **S**oft **L**arge **M**argin **C**lustering (SLMC for short hereafter). SLMC seeks the decision function and soft cluster memberships (for data to individual clusters) simultaneously with the cluster centers fixed to the given cluster encodings in the output space. SLMC enjoys the advantages of both MMC and the soft clustering methods, i.e., on one hand, it possesses a decision function with the maximal margin between clusters, on the other hand, it accomplishes soft assignments for each instance to individual clusters to capture the real data structure. Its algorithmic implementation follows an alternating iterative strategy, and the whole iteration process can be theoretically guaranteed to converge [14]. Following iterative LS-SVM [42], SLMC adopts the squared loss function so that each step in the iteration generates a closed-form solution, and the formulation of SLMC can straightforwardly be extended to the multi-class cases. Finally, the effectiveness of SLMC is verified by empirical comparisons with MMCs, and also KM and FCM as the baselines. Moreover, the convergence of the iterative solving process for SLMC is also empirically demonstrated.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 presents the proposed SLMC algorithm. Section 4 shows the empirical results and some conclusions are given in section 5.

2. Related work

Aiming to develop a soft large margin clustering method combining the advantages of both the large margin principle and the soft clustering idea, we first briefly introduce the related work about maximum margin clustering and soft clustering in separated sub-sections respectively.

2.1. Maximum margin clustering

Motivated by the large margin principle in classification learning, a large margin clustering method named maximum margin clustering (MMC) has been developed. It seeks the decision function and cluster labels for given data simultaneously so that the margin between clusters is maximized [38]. Specifically, given a dataset $X = \{x_i\}_{i=1}^n$ where each $x_i \in R^d$, then with a decision function $f(x) = w^T\phi(x)$ (the threshold b in $f(x)$ has been omitted here since it can be added implicitly by augmenting each instance with a one-valued element), the optimization problem of MMC can be formulated as

$$\begin{aligned} \min_{y_i}\min_{w,\xi_i} \quad & \frac{1}{2}\|w\|^2 + \frac{\lambda}{2}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i w^T\phi(x_i) \geqslant 1-\xi_i,\ \xi_i \geqslant 0 \\ & y_i \in \{-1,+1\},\ i = 1\ldots n \end{aligned} \tag{1}$$

where the variables $\{\xi_1\ldots\xi_n\}$ are introduced as the error-tolerances for given instances, λ is a trade-off parameter balancing between the margin maximization and data clustering, $\phi(\bullet)$: $R^d \to R^{d'}$ is a non-linear kernel mapping from the original input space to a higher dimension feature space or Reproducing Kernel Hilbert Space (RKHS), in which instances from different clusters are more likely to be linearly separable, and $w \in R^{d'}$ is a weight vector for features in the feature space. Note that though formally using the kernel mapping $\phi(\bullet)$, we actually do not need to formulate it explicitly. Specifically, if all calcu-

lations between instances in the algorithm can be expressed by dot products, we can replace those dot products by a reproducing kernel, i.e., $K(x_i,x_j) = \phi(x_i)^T\phi(x_j)$ rather than explicitly formulate $\phi(\bullet)$, which is the so-called "kernel trick"[7,10]. Moreover, in order to prevent problem (1) from a trivial solution, or an infinite margin for which all instances are assigned to a single cluster, or the outliers are simply separated from the rest instances, the authors imposed a constraint $-l \leqslant e^T y \leqslant l$ to (1), where l is a constant controlling the balance between clusters and $e \in R^n$ is an all-one vector.

For a multi-class (or multi-cluster) case, multi-class MMC [39] borrows the design idea from multi-class SVM [9]. Specifically, it defines a decision function $f_k(x) = w_k^T\phi(x)$ for each of the C clusters, and finally assigns instance x to the class with the maximum classification score, i.e., $\hat{y} = \arg\max_{k=1\ldots C} f_k(x)$. The optimization problem of the multi-class MMC can be formulated as

$$\begin{aligned} &\min_{y_i}\min_{w_k,\xi_i} \tfrac{1}{2}\sum_{k=1}^{C}\|w_k\|^2 + \tfrac{\lambda}{2}\sum_{i=1}^{n}\xi_i \\ &s.t. w_{y_i}^T\phi(x_i) + \delta_{y_i,r} - w_r^T\phi(x_i) \geqslant 1-\xi_i, \xi_i \geqslant 0 \\ &\forall i = 1\ldots n, r = 1\ldots C \end{aligned} \tag{2}$$

where $\delta_{yi,r} = 1$ if $y_i = r$ and 0 otherwise.

Clearly, the MMCs prescribe each instance to belong to a single cluster, whereas our SLMC allows each instance to belong to more than one cluster with the corresponding soft memberships. Moreover, it can be directly extended to the multi-class cases.

2.2. Soft (fuzzy) clustering

A class of clustering methods can be formulated as the minimization of some objective function [8], such as the well-known and widely-used K-means clustering (KM) [21]. As a clustering method based on label assignment, KM requires each instance to belong to exactly one cluster, whereas in real applications, data distributions from different clusters are usually overlapped, and thus each instance might belong to more than one cluster with different memberships [25]. As a result, to better capture the real data structure, Bezdek [4] developed the corresponding fuzzy c-means clustering (FCM) method by using the soft membership assignment, i.e.,

$$\begin{aligned} &\min \sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}^m (x_i - v_k)^2 \\ &s.t. \sum_{k=1}^{C} u_{ki} = 1 \\ &0 \leqslant u_{ki} \leqslant 1, \forall k = 1\ldots C, i = 1\ldots n \end{aligned} \tag{3}$$

where v_k denotes the center for the kth cluster, u_{ki} denotes the soft membership of instance x_i to the kth cluster, and m (>1) denotes the fuzzier or weight exponent for the cluster memberships. Through allowing each instance to belong to multiple clusters with the corresponding soft memberships, FCM and its variants can indeed provide better and more meaningful data partitions than their corresponding label-assignment ones [11].

Minimizing the objective function in (3) is equivalent to minimizing the trace sum of the fuzzy within-cluster scatter matrices [15], as a result, FCM actually seeks the cluster centers and fuzzy cluster memberships for data *in data space* so that the within-cluster compactness is minimized.

In this paper, we incorporate the benefits of soft clustering into large margin clustering learning and present a new soft large margin clustering method detailed in the next section.

3. Soft large margin clustering

In this section, we present the soft large margin clustering (SLMC) method, including its model formulation, problem solution, data prediction and algorithmic description in separated sub-sections respectively.

3.1. Model formulation

Given a dataset $X = \{x_i\}_{i=1}^n$ where $x_i \in R^d$. Let $f(x)=\mathbf{w}^T\phi(x)$ denote a decision function for a C-cluster clustering, where $\mathbf{w}\in R^{d'\times C}$ is a weight matrix. Then the prediction for each instance x_i is made by $\hat{y}_i = \arg\max_{k=1\ldots C} f_k(x_i)$, where $f_k(x_i)$ denotes the kth component of $f(x_i)$. Let $\mathbf{U}=[u_{ki}]_{C\times n}$ denote the soft partition matrix, in which each entry $u_{ki}\in[0,1]$ represents the soft membership of x_i to the kth cluster. Let $\{l_1,\ldots l_C\}$ denote the given encodings for the C clusters respectively, where each $l_k\in R^C$ (corresponding to the kth class) is encoded by the commonly-used one-of-C rule, i.e., the kth entry of l_k is set to 1 and the rest are 0, $\forall k$=1…C. For simpler optimization and direct extension to multi-cluster cases, we adopt the squared loss function as in iterative LS-SVM [42], and formulate the optimization problem of SLMC as

$$\begin{aligned}&\min_{u_{ki}}\min_{w}\tfrac{1}{2}\|\mathbf{w}\|^2+\tfrac{\lambda}{2}\sum_{k=1}^{C}\sum_{i=1}^{n}u_{ki}^m\|\mathbf{w}^T\phi(x_i)-l_k\|^2\\&s.t.\sum_{k=1}^{C}u_{ki}=1\\&0\leqslant u_{ki}\leqslant 1,\forall k=1\ldots C,i=1\ldots n.\end{aligned}\tag{4}$$

Different from FCM seeking a set of cluster centers and soft memberships for data in the data space, SLMC actually performs clustering in the output space by adopting the classification learning principle. Specifically, SLMC anchors the cluster centers to the predefined encodings for the C clusters, which is analogous to the predefined encodings for class labels in classification learning, and seeks both the decision function (as done in MMCs [38,42]) and the soft memberships for data in the output space. Moreover, different from MMC, SLMC allows each instance to belong to multiple clusters with the corresponding soft memberships through introducing the soft learning principle, consequently, it is more suitable to handle ambiguous cluster assignments than MMC, and can reflect the degrees of instances belonging to individual clusters by such soft memberships.

From (4), it is easily observed that SLMC maximizes the margin between clusters in the output space by minimizing the first term of the objective function [34]. Moreover, SLMC minimizes the sum of distances between given instances and the cluster centers in the output space (or more specifically, distances between the classification scores for given instances and cluster encodings) weighted by the corresponding fuzzy memberships, thereby, it minimizes the fuzzy within-cluster scatter in the output space simultaneously.

Note that though the one-of-C rule is adopted here to encode the l_ks in SLMC, some other encoding strategies, such as the regular simplex vertices encoding [1], can also be adopted for designing each l_k, but it is not the focus of this paper.

3.2. Problem solution

The optimization problem of SLMC is non-convex with respect to joint ($\mathbf{w}$, u). In this paper, we propose to solve it using an alternating iterative strategy to seek the decision function (w.r.t. the weight matrix $\mathbf{w}$) and the soft memberships for data respectively. Each step in the iteration generates a closed-form solution.

With fixed soft memberships, the optimization problem of SLMC for the decision function can be rewritten as

$$\min_{\mathbf{w}}\frac{1}{2}\|\mathbf{w}\|^2+\frac{\lambda}{2}\sum_{k=1}^{C}\sum_{i=1}^{n}u_{ki}^m\|\mathbf{w}^T\phi(x_i)-l_k\|^2\tag{5}$$

Obviously, it is a quadratic convex problem (w.r.t. $\mathbf{w}$ with fixed u_{ki}s) and formally similar to LS-SVM. According to the Representer Theorem [2], the minimization of (5) can yield a solution of the form $f(x)=\sum_{i=1}^{n}\alpha_i K(x_i,x)$, where each $\alpha_i\in R^{C\times 1}$, thus (5) can be translated into

$$\min_{\alpha}J_1(\alpha)=tr(\alpha\mathbf{K}\alpha^T)+\lambda\sum_{k=1}^{C}tr((\alpha\mathbf{K}-\mathbf{L}_k)\hat{\mathbf{U}}_k(\alpha\mathbf{K}-\mathbf{L}_k)^T)\tag{6}$$

where $\alpha=[\alpha_1,\alpha_2\ldots\alpha_n]\in R^{C\times n}$is the Lagrange multiplier matrix, $\mathbf{K}$ is the kernel matrix over the given data, each $\mathbf{L}_k$ is a $C\times n$ matrix with the kth row being an all-one vector, and the rests being all-zero vectors, $\mathbf{U}_k$ denotes the kth row of $\mathbf{U}$, and$\hat{\mathbf{U}}_k$denotes a diagonal matrix with the diagonal elements equaling to the squaredd values of the entries in $\mathbf{U}_k$.

Setting the derivative of J_1 w.r.t. α to zero, we have

$$\partial J_1/\partial\alpha=\alpha\mathbf{K}+\lambda\sum_{k=1}^{C}(\alpha\mathbf{K}-\mathbf{L}_k)\hat{\mathbf{U}}_k\mathbf{K}^T=0\tag{7}$$

which leads to a closed-form solution as

$$\alpha=\sum_{k=1}^{C}\mathbf{L}_k\hat{\mathbf{U}}_k\mathbf{K}^T(\sum_{k=1}^{C}\mathbf{K}\hat{\mathbf{U}}_k\mathbf{K}^T+\lambda\mathbf{K})^{-1}\tag{8}$$

Next, with a fixed decision function (corresponding to $\mathbf{w}$ or α), the optimization problem for the soft memberships can be reformulated as

$$\begin{aligned}&\min_{u_{ki}}\sum_{k=1}^{C}\sum_{i=1}^{n}u_{ki}^m\|\mathbf{w}^T\phi(x_i)-l_k\|^2\\&s.t.\sum_{k=1}^{C}u_{ki}=1\\&0\leqslant u_{ki}\leqslant 1,\forall k=1\ldots C,i=1\ldots n.\end{aligned}\tag{9}$$

Through adopting the Lagrange multiplier method, (9) can be rewritten as

$$J_2(u_{ki}) = \sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}^m \|f(x_i) - l_k\|^2 - \sum_{i=1}^{n}\gamma_i\left(\sum_{k=1}^{C} u_{ki} - 1\right) \tag{10}$$

Likewise, setting the derivative of J_2 w.r.t. each u_{ki} to zero, we have

$$\partial J_2/\partial u_{ki} = m\|f(x_i) - l_k\|^2 u_{ki}^{m-1} - \gamma_i = 0 \tag{11}$$

finally we have a closed-form solution as

$$u_{ki} = (\gamma_i/m\|f(x_i) - l_k\|^2)^{\frac{1}{m-1}} \tag{12}$$

Further, combining the constraint $\sum_{k=1}^{C} u_{ki} = 1$, we have

$$u_{ki} = \frac{1/\|f(x_i) - l_k\|^{2/(m-1)}}{\sum_{k=1}^{C} 1/\|f(x_i) - l_k\|^{2/(m-1)}} \tag{13}$$

where $k \in \{1 \ldots C\}$, $i \in \{1 \ldots n\}$.

3.3. Data prediction

Interestingly, prediction for each given instance in SLMC can be performed by not only the decision function but also the soft cluster membership, which is a major difference between SLMC and previous clustering methods such as MMC. Specifically, for any instance x_i, its cluster assignment made by the decision function is $\hat{y}_i = \arg\max_{k=1\ldots C} f_k(x_i)$, and its cluster label predicted by the soft cluster memberships is $\tilde{y}_i = \arg\max_{k=1\ldots C} u_{ki}$. Those two predictions are always consistent by the following proposition.

Proposition 1 Predictions for each given instance by the decision function and soft cluster membership are always consistent.

Proof: For an arbitrary instance x_i, its cluster label predicted by the decision function is $\hat{y}_i = \arg\max_{k=1\ldots C} f_k(x_i)$, thus $x_i \in X_k$ implies that $f_k(x_i) > f_j(x_i), \forall j = 1 \ldots C, j \neq k$, where X_k denotes the set of instances belonging to the kth cluster. At the same time, its cluster label predicted by the soft cluster membership is $\tilde{y}_i = \arg\max_{k=1\ldots C} u_{ki}$, thus from (13), $x_i \in X_k$ implies that $\|f(x_i) - l_k\|^2 < \|f(x_i) - l_j\|^2$, then $f(x_i)^T l_k > f(x_i)^T l_j$, or equivalently, $f_k(x_i) > f_j(x_i), \forall j = 1 \ldots C, j \neq k$. As a result, the prediction conditions for $x_i \in X_k$ by the decision function and soft cluster membership are equivalent, and thus the predictions for x_i by the decision function and soft cluster membership are consistent. ■

However, if the top-two cluster scores for x_i are equal, i.e., $f_k(x_i) = f_t(x_i)$ and $f_k(x_i) > f_j(x_i)$, $\forall j = 1 \ldots C, k \neq t, j \neq k, j \neq t$, then the corresponding two cluster memberships for x_i are equal as well, i.e., $u_{ki} = u_{ti}$ and $u_{ki} > u_{ji}$, $\forall j = 1 \ldots C, k \neq t, j \neq k, j \neq t$. In this case, x_i can be assigned to either cluster (between the kth and tth ones) by the decision function or the label memberships, however, we can always assign x_i to the cluster with the smaller or larger numbering (between k and t) to guarantee the desired consistency.

Table 1
The algorithm description of SLMC

Input	X— the input data
λ — the regularization parameter	
ε — the iteration stop parameter	
σ — the kernel parameter	
Maxiter — the maximum number for iteration	
Output	$f(x)$ — the decision function
U— the soft partition matrix for given data	
Procedure	
Obtain the initial U by FCM;	
Set the initial objective function value to infinity, i.e., J_0=INF;	
For k=1…Maxiter	
Update α by (8), and $f(x)$ by the Represent theorem with obtained α;	
Update U by (13);	
Update the objective function vaule J_k;	
If $\lvert J_k - J_{k-1}\rvert < \varepsilon J_{k-1}$	
Break, return $f(x)$ and U;	
Endif	
Endfor	

3.4. Algorithmic description

The optimization of SLMC follows an alternating iterative strategy. The iteration starts from an initial soft partition matrix learned by a simple soft clustering method such as FCM (analogous to iterative LS-SVM starting from an initial cluster labels learned by KM). The iteration terminates when $|J_k - J_{k-1}| < \varepsilon J_{k-1}$, where J_k is the objective function value at the kth iteration, and ε is a pre-defined threshold. The algorithm description of SLMC is given in Table 1.

Proposition 2 The sequence $\{J(\boldsymbol{\alpha}_t, u_t)\}$ obtained in the above algorithm w.r.t. SLMC converges.

Proof: First, the sequence of the objective function values generated by the above algorithm decreases monotonically. In fact, the objective function $J(\boldsymbol{\alpha}, u)$ is biconvex [14] in $(\boldsymbol{\alpha}, u)$. Specifically, with fixed u_t, the objective function is convex in $\boldsymbol{\alpha}$, thus the optimal $\boldsymbol{\alpha}^*$ can be obtained by minimizing $J(\boldsymbol{\alpha}, u_t)$, or equivalently optimizing (5). Now set $\boldsymbol{\alpha}_{t+1}=\boldsymbol{\alpha}^*$, then $J(\boldsymbol{\alpha}_{t+1}, u_t) = J(\boldsymbol{\alpha}^*, u_t) \leqslant J(\boldsymbol{\alpha}_t, u_t)$. Simultaneously, with current $\boldsymbol{\alpha}_{t+1}$, the objective function is convex in u, thus the optimal u^* can be obtained by minimizing $J(\boldsymbol{\alpha}_{t+1}, u)$, or equivalently optimizing (9). Now set $u_{t+1}=u^*$, then $J(\boldsymbol{\alpha}_{t+1}, u_{t+1}) = J(\boldsymbol{\alpha}_{t+1}, u^*) \leqslant J(\boldsymbol{\alpha}_{t+1}, u_t)$. Finally, $J(\boldsymbol{\alpha}_{t+1}, u_{t+1}) \leqslant J(\boldsymbol{\alpha}_{t+1}, u_t) \leqslant J(\boldsymbol{\alpha}_t, u_t)$, $\forall t \in N$. Hence, the consequence $\{J(\boldsymbol{\alpha}_t, u_t)\}$ decreases monotonically.

Further, since the objective function is non-negative, thus lower-bounded, as a result, the sequence $\{J(\boldsymbol{\alpha}_t, u_t)\}$ converges.

4. Experiment

In this section, we verify the effectiveness of SLMC over both synthetic and real UCI datasets. Sub-section 4.1 describes the experimental setups, sub-section 4.2 and 4.3 show the experimental results.

4.1. Experimental setups

In our experiments, we compare SLMC with MMC [38–39], in which the SDP problem is solved by the YALMIP [18] and SeDuMi [28] toolboxes. We also compare SLMC with an improved-version of MMC called iterative LS-SVM [42], since iterative LS-SVM has much lower computation complexity and usually better clustering performance than MMC. Moreover, LS-SVM also adopts the squared loss function and alternating iterative solving strategy as in SLMC, thus is exactly the label-assignment counterpart of SLMC. The optimization problem of iterative LS-SVM is formulated as follows,

$$\min_{y_i \in \{-1,1\}} \min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\lambda}{2}\sum_{i=1}^{n}\|\mathbf{w}^T\phi(x_i) - y_i\|^2 \tag{14}$$

Besides, we also compare SLMC with KM and FCM. For non-linear KM and FCM, we adopt the RBF kernel, and define the cluster centers in the data space, i.e.,

$$\min \sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}^m \|\phi(x_i) - \phi(v_k)\|^2 \tag{15}$$

where $u_{ki} \in \{0, 1\}$ for KM and $u_{ki} \in [0, 1]$ for FCM.

Following [36,38,42–43], we adopted both linear and RBF kernels in the experiments, and set the regularization and kernel parameters by exhaustive search from the grids $\{0.1, 0.5, 1, 5, 10\}$ and $\{0.25\sigma_0, 0.5\sigma_0, \sigma_0, 2\sigma_0, 4\sigma_0\}$, respectively, where σ_0 is the average distance between all instance pairs. We performed each method for 20 runs, kept the performance with the best kernel and parameters setting in each run, and finally reported the average in all runs.

Since the true class labels are already provided in evaluating the clustering performances, we adopt **clustering accuracy (CA)** [38,42] as the main performance index. Specifically, we first remove the class labels for the given instances, and perform clustering with the cluster number set to the given class number. Then we assign each cluster the majority label within it, and evaluate the clustering accuracy by measuring the consistency between the predicted cluster labels and the given class labels, i.e.,

$$CA = \frac{1}{N}\sum_{k=1}^{C}\max_{t=1\ldots C} T(C_k, L_t) \tag{16}$$

where C_k and L_t denote the kth cluster and tth class respectively, and $T(C_k, L_t)$ represents the number of instances belonging to the tth class and assigned to the kth cluster. As a result, a larger clustering accuracy indicates a better clustering performance.

Moreover, we also adopt three fuzzy validity indices for more comparisons between (fuzzy) SLMC and FCM, i.e., **partition coefficient (PC)** [4], **partition entropy (PE)** [3], and **Xie-Beni index (XB)** [27,37]. Specifically, the partition coefficient is defined by

$$PC = \frac{1}{n}\sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}^2 \tag{17}$$

$1/C \leqslant PC \leqslant 1$, and a larger PC value indicates a better clustering performance, corresponding to a clustering partition with more definite clustering partition or memberships. The partition entropy is defined by

$$PE = -\frac{1}{n}\sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}\log_2 u_{ki} \tag{18}$$

$0 \leqslant PE \leqslant \log_2 C$, and a smaller PE value indicates a better clustering performance, corresponding to a clustering partition with more definite clustering memberships. The Xie-Beni index is a performance index considering both cluster compactness and separation. It was originally proposed with m=2 [37], and then modified as

$$XB = \frac{\sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}^m \|x_i - v_k\|^2}{n\min_{i,j}\|v_i - v_j\|^2} = \frac{J(u,v)/n}{Sep(v)} \tag{19}$$

where $J(u, v)$ and Sep(v) measure the cluster compactness and separation respectively [27]. A smaller XB value indicates a better clustering performance. Since the clustering occurs in the kernel space in our experiments, we re-formulate the XB in the kernel space as

$$XB^{\Phi} = \frac{\sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}^m \|\phi(x_i) - \phi(v_k)\|^2}{n\min_{i,j}\|\phi(v_i) - \phi(v_j)\|^2} \tag{20}$$

We abuse the notation XB for XB^{ϕ} hereafter for simplicity, and apply the RBF kernel to (20), finally we have

$$XB = \frac{\sum_{k=1}^{C}\sum_{i=1}^{n} u_{ki}^m (1 - \exp^{\frac{\|x_i - v_k\|^2}{-2\sigma^2}})}{n\min_{i,j}(1 - \exp^{\frac{\|v_i - v_j\|^2}{-2\sigma^2}})} \tag{21}$$

where σ is the band width of the RBF kernel.

4.2. Synthetic dataset

Since the correct cluster number may be unknown in real application, we use a 2d synthetic dataset for demonstrating how SLMC performs with different settings of cluster number in this sub-section. However, in experiments over the real datasets in the next sub-section, we directly set the cluster number to the given class number following Refs. [36,38,42–43].

Table 2
The attributes of the synthetic dataset

Class	1 (·)	2 (*)	3 (o)	4 (◇)
Mean	(0, 0)	(0, 1.5)	(1.6, 0)	(1, 1.5)
Covariance	$\begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$,	$\begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$
Number	100	100	100	100

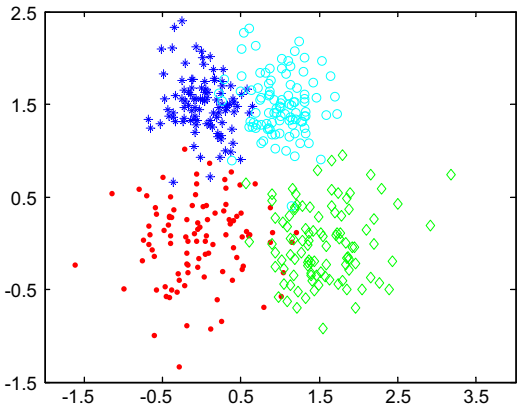

Fig. 1. The distribution of the synthetic dataset

The synthetic dataset is generated from four Gaussian distributions, one for each cluster. Table 2 gives the attributes of the dataset, and Fig. 1 further displays the distribution.

The correct cluster number of the synthetic dataset is 4, however, we performed performance comparison with the cluster number set to 3, 4, 5 and 6, respectively. Table 3 gives the clustering accuracies of the four compared methods, and the bod value in each column indicates the best cluster accuracy obtained for each cluster number. Table 4 gives the PC, PE and

Table 3
Clustering accuracies of KM, FCM, iter_LSSVM and SLMC

Cluster number	3	4	5	6
KM	0.7000	0.9125	0.8850	0.8775
FCM	0.7100	0.9225	0.9075	0.9050
ITER_LSSVM	0.7100	0.9275	0.9175	0.9250
SLMC	**0.7125**	**0.9375**	**0.9300**	**0.9325**

Table 4
PC, PE and XB performances of FCM and SLMC

Cluster number	FCM			SLMC		
	PC	PE	XB	PC	PE	XB
3	0.6918	0.8171	0.1192	**0.7115**	**0.7970**	**0.0978**
4	0.6653	0.9532	0.1700	**0.6743**	**0.9428**	**0.1653**
5	0.6035	1.1593	0.2367	**0.6253**	**1.0156**	**0.2354**
6	0.5453	1.3639	0.3976	**0.5579**	**1.2557**	**0.3879**

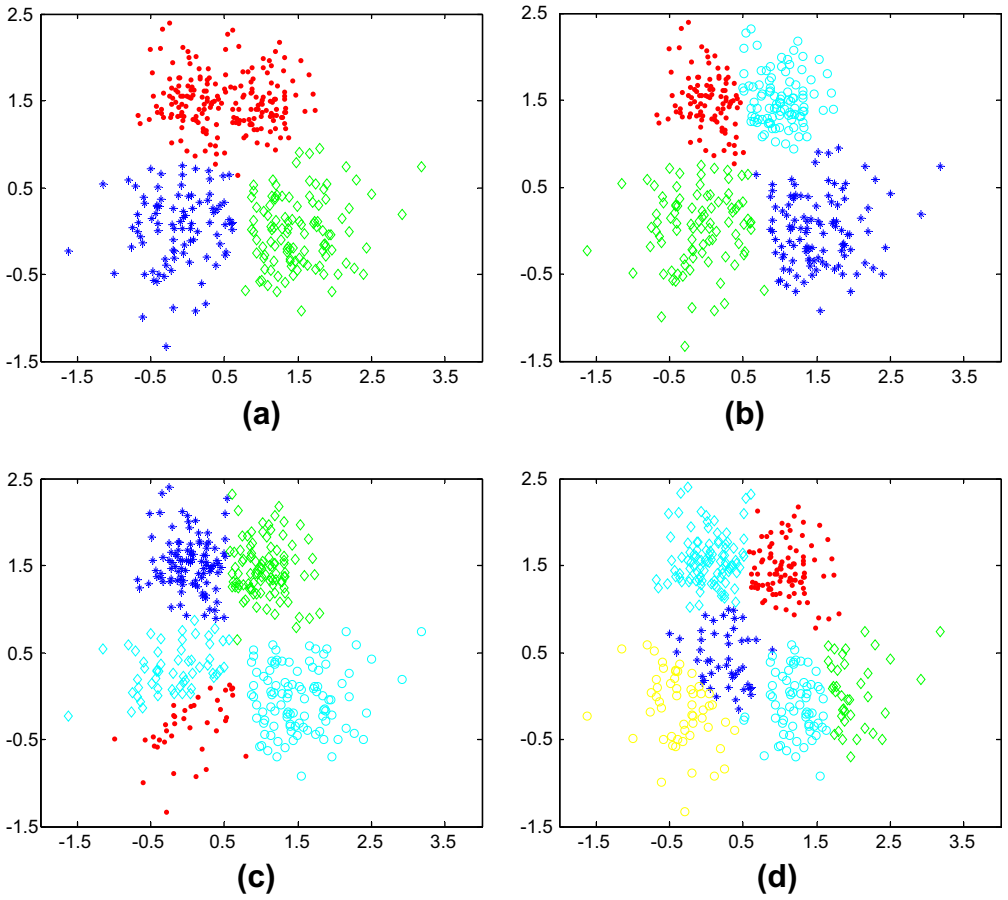

Fig. 2. The clustering results of SLMC with the cluster number set to (a) 3, (b) 4, (c) 5 and (d) 6, respectively

XB performances of FCM and SLMC, in which the bold values in each row indicate the better performances for each cluster number, and Fig. 2 displays the clustering results of SLMC under different settings of clustering number (The clustering results of the other three methods are not displayed here due to their visual similarities with those of SLMC). From Tables 3 and 4, we can observe that with different settings of cluster number, SLMC obtains the best clustering accuracies among those compared methods, and achieves better PC, PE and XB performances than FCM.

4.3. Real datasets

In this sub-section, we report the results on 18 UCI datasets. Sub-section 4.3.1 describes the datasets adopted for comparison, section 4.3.2 analyzes the value for the fuzzier m in SLMC, section 4.3.3 gives the comparison results, and section 4.3.4 empirically reveals the convergence of SLMC.

4.3.1. Datasets description

We evaluate the performance of SLMC over 18 UCI datasets, whose attributes are respectively described in Table 5, including the total number of instances, number of features, number of classes, and number of instances in individual classes (in the brackets). For the *optdigits* and *pendigits* datasets, we focus on the pairs which are difficult to distinguish [35,42], i.e., 3 versus 8, 3 versus 9, and 8 versus 9. For nominal features in datasets such as *lenses*, *hepatitis*, *heart* and *arrhythmia*, we simply treated them as numeric ones for all compared methods, which would not influence the fairness for comparison.

4.3.2. Analysis on the fuzzier m in SLMC

In this sub-section, we analyze the influence of m on the performance of SLMC. Specifically, we exhibit the performances of SLMC with respect to different values of m from [1.5,2,2.5,3,3.5,4,4.5,5] over the 18 datasets in Fig. 3 below.

From Fig. 3, we can make several observations as follows,

- Fig. 3 (a) shows the CA performances of SLMC with respect to different m values. From Fig. 3 (a), we find that SLMC achieves the best average CA performance over all 18 datasets when m is set to 2, as a result, we set m to 2 in our whole experiments.
- Fig. 3 (b) and (c) shows the PC and PE performances of SLMC with respect to different m values respectively. From those figures, we can find that the PC performances descend as m ascends, and the PE performances ascend as m ascends, since a larger m in SLMC usually corresponds to a less definite cluster partition. However, it is not the case for the *arrhythmia* dataset, corresponding to the PC curve at the bottom of Fig. 3 (b) and the PE curve at the top of Fig. 3 (c), since now the obtained u_{ki}s are all close to 0.5 even when m is as small as 1.5.
- Fig. 3 (d) shows the XB performances of SLMC with respect to different m values. We can observe that when m is large, SLMC usually yields a large XB performance, in this case, the obtained u_{ki}s would be close to 0.5, corresponding to a single sample mean in the data space.

4.3.3. Performance comparison

In this sub-section, we first compare SLMC (m=2) with MMC and iterative LS-SVM in terms of clustering accuracy, along with FCM and KM as the baselines. The comparison results are given in Table 6, in which each row gives the clustering performances (including the average accuracy and variance over 20 independent runs) over each dataset. However, since the SDP formulation of MMC is quite expensive in terms of both time and memory [42], we only provide its results on the first

Table 5
The attributes of UCI datasets used

Dataset	# Instance	# Feat.	# Class (# instance in individual classes)
Lenses	24	4	3 (4, 5, 15)
Soybean	47	35	4 (10, 10, 10, 17)
Echocardiogram	132	12	2 (89, 43)
Hepatitis	155	19	2 (32, 123)
Wine	178	13	3 (59, 71, 48)
Glass	214	10	6 (70, 76, 17, 0, 13, 9, 29)
Heart	270	13	2 (150, 120)
Ecoli	336	8	6 (143, 77, 52, 35, 20, 5, 2, 2)
Ionosphere	351	34	2 (225, 126)
Optdigits89	354	64	2 (174, 180)
Optdigits38	357	64	2 (183, 174)
Optdigits39	363	64	2 (183, 180)
Arrhythmia	452	279	13 (245, 44, 15, 15, 13, 25, 3, 2, 9, 50, 4, 5, 22)
Austra	690	14	2 (307, 383)
Pendigits89	1438	16	2 (719, 719)
Pendigits38	1438	16	2 (719, 719)
Pendigits39	1438	16	2 (719, 719)
Image Segment	2310	19	7 (330, 330, 330, 330, 330, 330, 330)

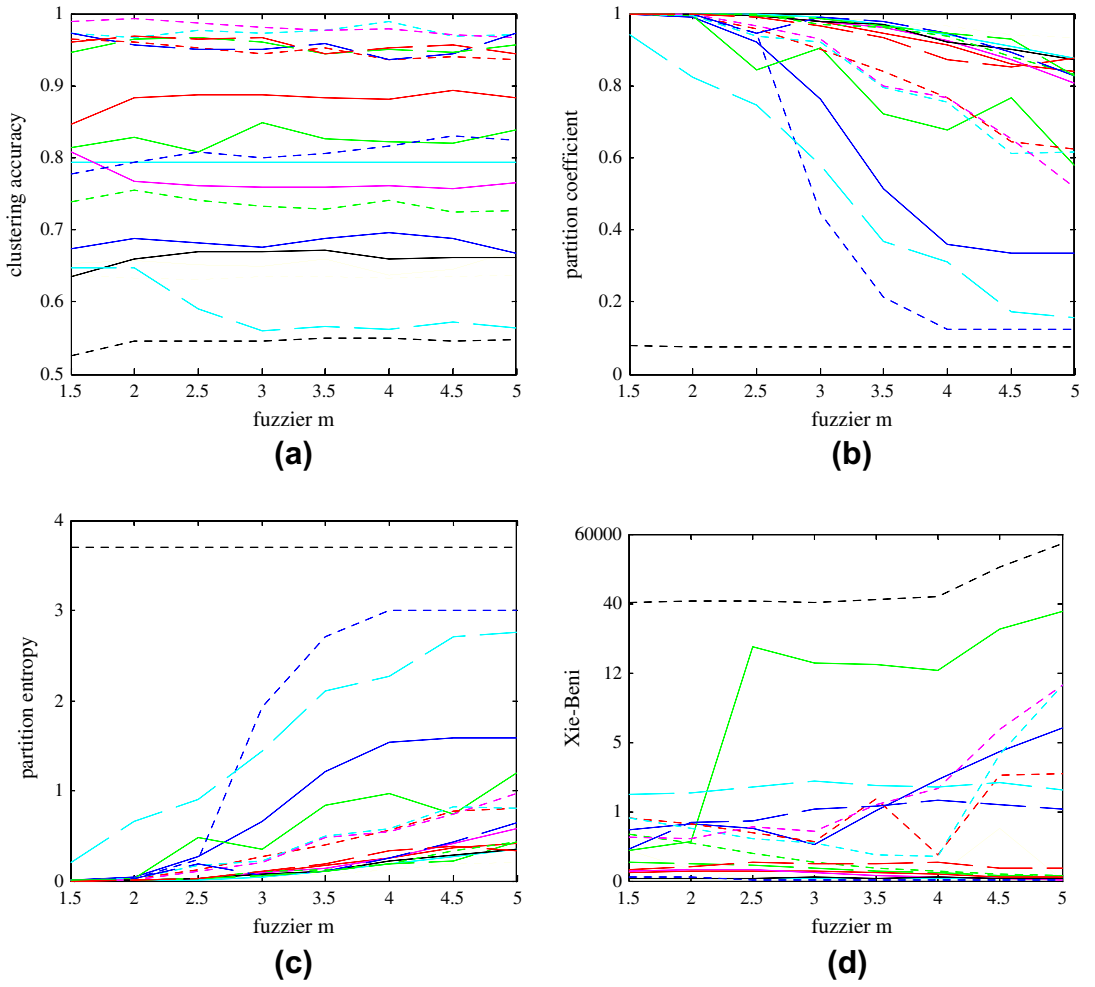

Fig. 3. The (a) clustering accuracy (b) partition coefficient (c) partition entropy and (d) Xie-Beni index of SLMC with respect to different values of fuzzier m from [1.5,2,2.5,3,3.5,4,4.5,5]

Table 6
The clustering accuracies of compared methods on 18 UCI datasets

Dataset	KM	FCM	MMC	Iterative LS-SVM	SLMC
Lenses	0.5546±0.0015	0.5683±0.0039	0.6548±0.0021	0.6729±0.0039	**0.6886±0.0018**∗
Soybean	0.7544±0.0064	0.7762±0.0017	0.7835±0.0039	0.8141±0.0078	**0.8287±0.0026**∗
Echocardiogram	0.8349±0	0.8264±0	0.8372±0	0.8460±0	**0.8836±0**∗
Hepatitis	0.6713±0.0004	0.6748±0	0.7756±0	**0.7935±0**	**0.7935±0**
Wine	0.7135±0	0.7027±0	0.7896±0	**0.8081±0.0003**∗	0.7681±0
Glass	0.5556±0.0006	0.5523±0	–	0.6501±0.0007	**0.6567±0.0013**
Heart	0.6189±0	0.6456±0.0008	–	0.6311±0.0001	**0.6593±0.0004**∗
Ecoli	0.5385±0.0003	0.5917±0	–	0.7749±0.0006	**0.7946±0**∗
Ionosphere	0.7124±0.0001	0.7173±0.0004	–	0.7392±0	**0.7554±0**∗
Optdigits89	0.9313±0	0.9291±0.0002	–	**0.9652±0**∗	0.9602±0
Optdigits38	0.9568±0	0.9419±0	–	**0.9794±0**∗	0.9743±0
Optdigits39	0.9327±0	0.9451±0.0002	–	0.9842±0	**0.9922±0**∗
Arrhythmia	0.3342±0	0.3695±0	–	0.5230±0	**0.5465±0**∗
Austra	0.6670±0	**0.6508±0**∗	–	0.6378±0	0.6330±0
Pendigits89	0.9483±0.0001	0.9426±0.0001	–	**0.9731±0**∗	0.9567±0
Pendigits38	0.9271±0.0002	0.9415±0.0003	–	0.9430±0	**0.9642±0.0001**∗
Pendigits39	0.9155±0.0014	0.9230±0.0007	–	0.9580±0.0002	**0.9691±0**∗
Image Segment	0.5251±0.0019	0.5419±0.0030	–	**0.6490±0.0012**	0.6473±0.0028

5 (small-scale) datasets. In each row, the bold value indicates the best performance among all compared ones, and the performance marked by "$*$" indicates that the corresponding method obtains the best performance with statistically significant difference by t-test.

From Table 6, we can observe that SLMC performs the best over 12 out of the 18 datasets, and has statistically significant superiority over 10 ones, thus SLMC is relatively effective in terms of CA. More specifically, compared with FCM, SLMC performs better on 17 out of the 18 datasets, indicating the superiority of the large margin principle. On the other hand, compared with iterative LS-SVM, SLMC performs better on 11 datasets, comparable on 1 datasets, and worse on only 6 datasets. As a result, the assumption that each instance belongs to more than one cluster with the corresponding soft memberships is usually more suitable to capture the real data distribution, and consequently, SLMC can usually achieve better CA performances than the large margin clustering methods, including MMCs and iterative LS-SVM. Besides, we can also observe that when SLMC performs worse than iterative LS-SVM (over datasets such as *wine, optdigits89* and *optdigits38* here), FCM also performs worse than KM, and a major reason seems to be that the above assumption does not cater well for all cases.

Moreover, we also compare SLMC with FCM in terms of fuzzy indices PC, PE, and XB. The results are shown in Table 7, in which each row gives the performances over each dataset, a bold value indicates the better performance between the two compared methods according to some index, and a performance marked by "$*$" indicates that the corresponding method obtains the better performance with statistically significant difference by t-test.

From Table 7, we can observe that when m is set to 2, the PC performances by SLMC are significantly larger (better) than those by FCM on 16 out of the 18 datasets, and similarly, the PE performances by SLMC are significantly smaller (better) than those by FCM on 17 datasets. Moreover, the XB performances by SLMC are significantly smaller (better) than those by FCM over 9 datasets, and significantly larger (worse) than those by FCM over 9 datasets. As a result, SLMC usually achieves better PC and PE performances, and comparable XB performances compared with FCM when m is set to 2.

It is also worth noting that the XB performances of FCM are much larger than those of SLMC over *optdigits*89, *optdigits*38, and *optdigits*39, and a possible reason is the fixation of m to 2 in FCM. As can be seen in Table 8, FCM with m=1.5 can achieve much better XB performances than FCM with m=2 over those three datasets, and when $m \geqslant 2$ here, FCM actually yields a cluster partition with each u_{ki} close to 0.5 (the PC and PE performances are close to $1/C$ and $\log_2 C$ respectively, where C=2 here).

4.3.4. Empirical demonstration for convergence of SLMC

Though having theoretically proved the convergence for the iterative solving process of SLMC by proposition 2, in this sub-section, we also demonstrate it over 6 datasets as empirical justifications. However, since the observations are similar

Table 7
The comparison results between SLMC and FCM on 18 UCI datasets

Dataset	FCM			SLMC		
	PC	PE	XB	PC	PE	XB
Lenses	0.4231±0	1.3912±0	**0.7338±0**$*$	**0.9909±0**$*$	**0.0387±0**$*$	0.8137±0
soybean	0.4755±0	1.4064±0	1.7399±0	**0.9940±0**$*$	**0.0303±0**$*$	**0.5670±0**$*$
Echocardiogram	0.7014±0	0.6677±0	0.3715±0	**0.9994±0**$*$	**0.0032±0**$*$	**0.1374±0**$*$
Hepatitis	0.7266±0	0.6252±0	0.3316±0	**0.9998±0**$*$	**0.0011±0**$*$	**0.0401±0**$*$
Wine	0.7909±0	0.5488±0	**0.1257±0**$*$	**0.9997±0**$*$	**0.0018±0**$*$	0.1527±0
Glass	0.7370±0	0.7726±0	**0.0583±0**$*$	**0.9997±0**$*$	**0.0022±0**$*$	0.1628±0
Heart	0.7126±0	0.6493±0	0.2562±0	**0.9997±0**$*$	**0.0015±0**$*$	**0.0355±0**$*$
Ecoli	0.7271±0	0.8277±0	**0.0504±0**$*$	**0.9928±0**$*$	**0.0322±0**$*$	0.0537±0
Ionosphere	0.6512±0	0.7522±0	0.7117±0	**0.9998±0**$*$	**0.0010±0**$*$	**0.5329±0**$*$
Optdigits89	0.5005±0	0.9993±0	105.2974±62.1058	**0.9920±0**$*$	**0.0275±0**$*$	**0.8281±0**$*$
Optdigits38	0.5006±0	0.9991±0	93.3301±52.0919	**0.9957±0**$*$	**0.0158±0**$*$	**0.7546±0**$*$
Optdigits39	0.5004±0	0.9995±0	187.0511±129.1753	**0.9978±0**$*$	**0.0181±0**$*$	**0.5996±0**$*$
Arrhythmia	**0.0770±0**	**3.6998±0**	**1080.6331±214.7289**$*$	0.0769±0	3.7004±0	1504.2701±198.3129
Austra	**0.9995±0**$*$	0.0136±0	**0.0021±0**$*$	0.9974±0	**0.0064±0**$*$	0.0044±0
Pendigits89	0.6400±0	0.7816±0	**0.4468±0**$*$	**0.9920±0**$*$	**0.0275±0**$*$	0.8370±0
Pendigits38	0.7419±0	0.5795±0	**0.2242±0**$*$	**0.9992±0**$*$	**0.0031±0**$*$	0.2473±0
Pendigits39	0.6235±0	0.8062±0	0.6712±0	**0.9984±0**$*$	**0.0054±0**$*$	**0.2056±0**$*$
Image Segment	0.3807±0	1.9633±0	**0.4571±0.0071**$*$	**0.8215±0**$*$	**0.6505±0**$*$	2.0826±0

Table 8
The performances over *optdigits*89, *optdigits*38, and *optdigits*39 with m from {1.5, 2, 2.5}

dataset	Optdigits89			Optdigits38			Optdigits39		
m	1.5	2	2.5	1.5	2	2.5	1.5	2	2.5
CA	0.9049	0.9008	0.9003	0.9405	0.9419	0.9444	0.9449	0.9451	0.9327
PC	0.6379	0.5005	0.5004	0.6566	0.5006	0.5003	0.5445	0.5005	0.5003
PE	0.7833	0.9992	0.9995	0.7517	0.9991	0.9995	0.9339	0.9993	0.9996
XB	1.6137	105.2974	94.7401	1.3532	93.3301	84.6727	6.1745	187.0511	125.1963

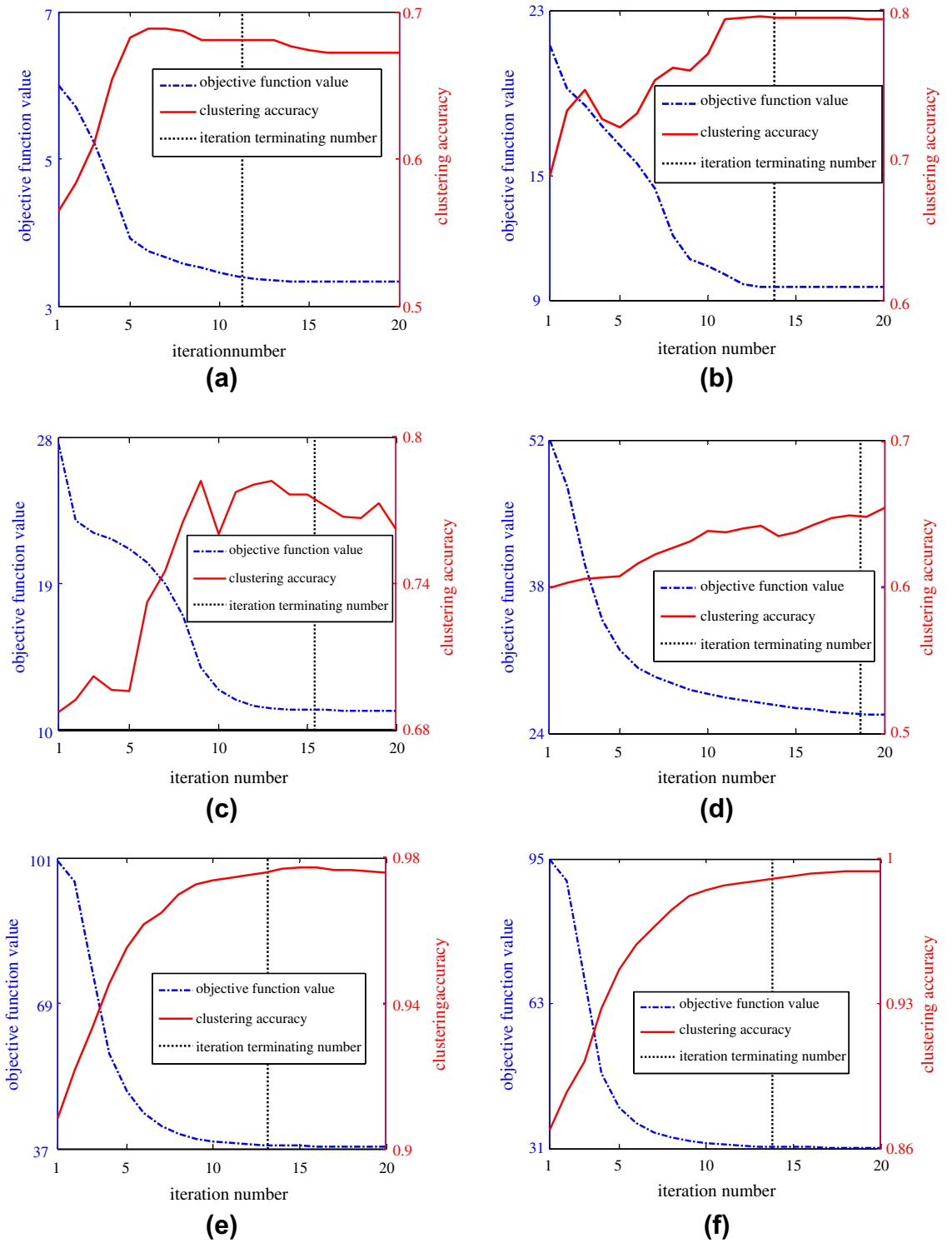

Fig. 4. The average objective function values and clustering accuracies of SLMC in the first 20th iterations, as well as the average iteration numbers on (a) *Lenses* (b) *Hepatitis* (c) *Wine* (d) *Heart* (e) *Optdigits38* and (f) *Optdigits39*

over all 18 datasets, we simply use 6 ones as representatives. Specifically, we provide the objective values and clustering accuracies of SLMC in the first 20 iterations, as well as the average number of iterations in Fig. 4. In Fig. 4, the reported results are all average ones over 20 independent runs, and in each run, only the results corresponding to the best parameters and kernel combinations contribute to the final results.

From Fig. 4, we can observed that the objective function value decreases monotonically with the increase of the iterative number, and the iterations all terminate within 20 rounds, which demonstrates that the iterative solving process of SLMC is indeed convergent, and the convergence speed is acceptable as that of FCM. Moreover, the clustering accuracy tends to increase with the growth of the iterative number, demonstrating that SLMC can achieve better performance than its initializer FCM. However, it can also be observed that the clustering accuracy does not increase monotonically, which indicates that early termination might occur in SLMC, as a result, some heuristic terminative strategy for SLMC is one of our future works.

5. Conclusion

In this paper, we develop a new soft large margin clustering method referred to as soft large margin clustering (SLMC for short), which combines the advantages of both the large margin principle and the soft clustering idea. SLMC possesses a decision function with the maximal margin between clusters, and at the same time, accomplishes soft assignments for each instance to individual clusters to reflect the nature of given data. The resulting optimization problem of SLMC is solved using an alternating iterative strategy, in which each step has a closed-form solution. The convergence for the iterative solving process has been theoretically proved, and empirically demonstrated as well. The formulation of SLMC can directly be extended to the multi-class cases. Experiments on several real datasets demonstrate its competitiveness compared with both FCM and MMCs.

In the future, there are still some worth-studying issues summarized as follows:

- In this paper, the cluster centers in the output space (or cluster encodings) are simply encoded by the one-of-C rule, while some other encoding strategies can also be adopted, or those cluster centers can also be optimized in the learning phase. Thereby, we will investigate how the cluster encoding manner affects the performance of SLMC in our future work.
- We will study the application of SLMC to unbalanced clustering problem by, e.g., enforcing balance constraints among multiple clusters.
- We will seek for a heuristic termination strategy for SLMC such that the "optimal" clustering accuracy can be achieved.

Acknowledgments

We would like to thank the National Science Foundations of China (NSFC) under Grant Nos. 61035003 and 60905002, Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (Information and Communication, yx002001).

References

[1] S. An, W. Liu, and S. Venkatesh, Face Recognition Using Kernel Ridge Regression, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.
[2] M. Belkin, P. Niyogi, V. Sindhwani, Manifold Regularization: A geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research 7 (2006) 2399–2434.
[3] J.C. Bezdek, Cluster validity with fuzzy sets, Journal of Cybernetics 3 (1974) 58–73.
[4] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algoritms, Plenum Press, New York, 1981.
[5] B. Borchers, CSDP: A C library for semidefinite programming, Optimization Methods Software 11&12 (1999) 613–623.
[6] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, U. K., 2004.
[7] C.J.C. Burges, A tutorial on support vector machine for pattern recognition, Data Mining Knowl. Discovery 2 (1998) 121–167.
[8] S. Chen, D. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure, IEEE Transactions on System, Man and Cybernetics-Part B 34 (2004) 1907–1916.
[9] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, Journal of Machine Learning Research 2 (2001) 265–292.
[10] N. Cristianini, J.S. Taylor, An introduction to Support vector Machines and other kernel-based learning methods, Cambridge University Press, UK, 2000.
[11] S. Eschrich, J. Ke, L.O. Hall, D.B. Goldgof, Fast Accurate Fuzzy Clustering through Data Reduction, IEEE Transactions on Fuzzy Systems 11 (2003) 262–270.
[12] A. Gersho, R.M. Gray, Vector quantization and signal compression, Kluwer, Boston, MA, 1992.
[13] A. Ghosh, N.S. Mishr, S. Ghosh, Fuzzy clustering algorithms for unsupervised change detection in remote sensing images, Information Sciences 181 (2011) 699–715.
[14] J. Gorski, F. Pfeuffer, Biconvex sets and optimization with biconvex functions: a survey and extensions, Mathematical Methods of Operations Research 66 (2007) 373–407.
[15] D. E. Gustafson and W. C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in Proceedings of IEEE Conference Decision Control, San Diego, CA, 1979.
[16] C.-H. Huang, A reduced support vector machine approach for interval regression analysis, Information Sciences 217 (2012) 56–64.
[17] P. Huang, D. Zhang, Locality sensitive C-means clustering algorithms, Neurocomputing 73 (2010) 2935–2943.
[18] J. Löfberg, YALMIP : A Toolbox for Modeling and Optimization in MATLAB, Proceedings of the CACSD Conference Taipei, Taiwan 2004.
[19] X. Li, H.-S. Wong, S. Wu, A fuzzy minimax clustering model and its applications, Information Sciences 186 (2012) 114–125.
[20] Y. Li, I. W. Tsang, J. T. Kwok, and Z. Zhou, Tighter and convex maximum margin clustering, Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater Beach, Florida, USA, 2009.
[21] J. B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, 1967.
[22] E.A. Maharaj, P. D'Urso, Fuzzy clustering of time series in the frequency domain, Information Sciences 181 (2011) 1187–1211.
[23] G. McLachlan, D. Peel, Finite mixture models, Wiley, New York, 2000.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss, On spectral clustering: Analysis and an algorithm, Advances in Neural Information Processing Systems 17, Vancouver, British Columbia, Canada, 2002.
[25] J.V. Oliveira, W. Pedrycz, Advances in Fuzzy Clustering and its Applications, Wiley, London, 2007.
[26] I. Ozkan, I.B. Turksen, Upper and lower values for the level of fuzziness in FCM, Information Sciences 177 (2007) 5143–5152.
[27] N.R. Pal, J.C. Bezdek, On cluster validity for fuzzy c-means model, IEEE Transactions on Fuzzy Systems 3 (1995) 370–379.
[28] I. Polik, SeDuMi: a Matlab toolbox for optimization over symmetric cones. Available at http://sedumi.ie.lehigh.edu/.
[29] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, More efficiency in multiple kernel learning, Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, 2007.
[30] A. Rakotomamonjy, F.R. Bach, S. Canu, Y. Grandvalet, SimpleMKL, Journal of Machine Learning Research 9 (2008) 2491–2521.
[31] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (2004) 199–222.
[32] A. J. Smola, S. V. N. Vishwanathan, T. Hofmann, Kernel methods for missing variables, Proceedings of the 10th International Workshop on Artificial Intelligence and, Statistics, 2005.
[33] J.F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization Methods Software 11–12, Special Issue on Interior Point, Methods (1999) 625–653.
[34] J.A.K. Suykens, J. Vandewalle, Least Squareds Support Vector Machine Classifiers, Neural Processing Letters 9 (1999) 293–300.
[35] H. Valizadegan and R. Jin, Generalized Maximum Margin Clustering and Unsupervised Kernel Learning, Advances in neural information processing systems, Vancouver, B.C., Canada, 2007.
[36] F. Wang, B. Zhao, C. Zhang, Linear Time Maximum margin clustering, IEEE Transactions on Neural Networks 21 (2010) 319–332.
[37] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Transaction on Pattern Analysis and Machine Intelligence 13 (1991) 841–847.
[38] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, Maximum Margin Clustering, Advances in neural information processing systems, Whistler, Canada, 2005.
[39] L. Xu and D. Schuurmans, Unsupervised and Semi-supervised Multi-class Support Vector Machines, Proceedings of the 20th National Conference on Artificial Intelligence, Pittsburgh, PA, 2005.
[40] A. Yuille, A. Rangarajan, The concave-convex procedure, Neural Computation 15 (2003) 915–936.
[41] H. Zeng, Y.-M. Cheung, Semi-Supervised Maximum Margin Clustering with Pairwise Constraints, IEEE Transaction on Knowledge and Data Engineering 24 (2012) 926–939.
[42] K. Zhang, I.W. Tsang, J.T. Kwok, Maximum margin clustering made practical, IEEE Transactions on Neural Networks 20 (2009) 583–596.
[43] B. Zhao, F. Wang, and C. Zhang, Efficient maximum margin clustering via cutting plane algorithm, Proceedings of the 8th SIAM International Conference on Data Mining, Atlanta,GA, 2008.