Pattern Recognition 48 (2015) 953–967

Contents lists available at ScienceDirect

Pattern Recognition

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

Active constrained fuzzy clustering: A multiple kernels learning approach

Ahmad Ali Abin*, Hamid Beigy

Department of Computer Engineering, Sharif University of Technology, Azadi Ave., Tehran, Iran

ARTICLE INFO

Article history:
Received 29 November 2013
Received in revised form
16 August 2014
Accepted 10 September 2014
Available online 20 September 2014

Keywords:
Constrained clustering
c-Means fuzzy clustering
Multiple kernels
Active constraint selection

ABSTRACT

In this paper, we address the problem of constrained clustering along with active selection of clustering constraints in a unified framework. To this aim, we extend the improved possibilistic c-Means algorithm (IPCM) with multiple kernels learning setting under supervision of side information. By incorporating multiple kernels, the limitation of improved possibilistic *c*-means to spherical clusters is addressed by mapping non-linear separable data to appropriate feature space. The proposed method is immune to inefficient kernels or irrelevant features by automatically adjusting the weight of kernels. Moreover, extending IPCM to incorporate constraints, its strong robustness and fast convergence properties are inherited by the proposed method. In order to avoid querying inefficient or redundant clustering constraints, an active query selection heuristic is embedded into the proposed method to query the most informative constraints. Experiments conducted on synthetic and real-world datasets demonstrate the effectiveness of the proposed method.

1. Introduction

In recent years, constrained clustering has been emerged as an efficient approach for data clustering and learning the similarity measure between patterns [1]. It has become popular because it can take advantage of side information when it is available. Incorporating domain knowledge into the clustering by adding constraints enables users to specify desirable properties of the result and improves the robustness of clustering algorithm.

The first introduction of constrained clustering to the machine learning [2] focused on the use of instance-level pairwise constraints. Pairwise constraints specify whether two objects belong to the same cluster or not, known as the must-link (ML) constraints and the cannot-link (CL) constraints, respectively. Recent techniques in constrained clustering include integrating both clustering algorithm and learning the underlying similarity metric in a uniform framework [3], joint clustering and distance metric learning [4], topology preserving distances metric learning [5], Kernel approaches for metric learning [6], learning a margin-based clustering distortion measure using boosting [7], learning Mahalanobis distances metric [8,9], learning distances metric based on similarity information [10], and learning a distance metric transformation that is globally linear but locally nonlinear [11], to mention a few.

Existing methods in constrained clustering reported the clustering performance averaged over multiple randomly generated constraints [2,5,7,8]. Random constraints do not always improve the quality of results [12]. In addition, averaging over several trials is not possible in many applications because of the nature of problem or the cost and difficulty of constraint acquisition. An alternative to get the most beneficial constraints for the least effort is to actively acquire them. There is a small range of studies on active selection of clustering constraints based on: “farthest-first” strategy [13], hierarchical clustering [14], theory of spectral decomposition [15], fuzzy clustering [16], Min–Max criterion [17], graph theory [18], and boundary information of data [19]. These methods choose constraints without considering how the underlying clustering algorithm utilizes the selected constraints. If we choose constraints independent of the clustering algorithm, it will have better performance for some algorithms but perform worse for some others.

This paper proposes a unified framework for constrained clustering and active selection of clustering constraints. It integrates the improved possibilistic *c*-Means (IPCM) [20] with a multiple kernels learning setting under supervision of side information. The proposed method attempts to address the limitation of IPCM to spherical clusters by incorporating multiple kernels. In addition, it immunizes itself to inefficient kernels or irrelevant features by automatically adjusting weight of kernels in an alternative optimization manner. In order to avoid querying inefficient or redundant constraints, an active query selection heuristic is embedded into the proposed method based on the measurement of Type-II mistake in clustering.

* Corresponding author. Tel.: +98 21 66166674.
E-mail addresses: abin@ce.sharif.edu (A.A. Abin), beigy@sharif.edu (H. Beigy).

http://dx.doi.org/10.1016/j.patcog.2014.09.008
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

This heuristic attempts to query the most informative set of constraints based on the current state of the clustering algorithm. Altogether, the proposed method attempts to have the whole robustness against noise and outliers, immunization to inefficient kernels or irrelevant features, fast convergence rate, and selection of useful set of constraints in a unified framework. Experiments conducted on synthetic and real-world datasets demonstrate the effectiveness of the proposed method.

The rest of this paper is organized as follows: a brief overview of fuzzy clustering is provided in Section 2. We then introduce the proposed method in Section 3 and the proposed embedded active constraint selection heuristic is given in Section 4. Experimental results and time complexity analysis are presented in Section 5. Discussion on the proposed method is presented in Section 6. This paper concludes with conclusions and future works in Section 7.

2. Fuzzy clustering

Many clustering algorithms have been proposed over the past decades which perform hard clustering of data [21–24]. On the other hand, clusters may overlap in many real-world problems so that many items have the characteristics of several clusters [25]. In order to consider the overlaps among clusters, it is more natural to assign a set of memberships to each item, one for each cluster. This method is called fuzzy clustering. Fuzzy c-Means (FCM) algorithm is one of the most promising fuzzy clustering methods, which in most cases is more flexible than the corresponding hard clustering [26]. Given a dataset, $X=\{x_1,\ldots,x_N\}$, where $x_i\in R^l$ and l is the dimension of feature vector, FCM partitions X into C fuzzy partitions by minimizing the following objective function:

$$J_{FCM}(U,V)=\sum_{c=1}^{C}\sum_{i=1}^{N}u_{ci}^{m}d_{ci}^{2} \tag{1}$$

where $V=(v_1,\ldots,v_C)$ is a C-tuple of prototypes, d_{ci}^2 is the distance of feature vector x_i to prototype v_c, i.e. $\|x_i-v_c\|^2$, N is the total number of feature vectors, C is the number of partitions, u_{ci} is the fuzzy membership of x_i in partition c satisfying $\sum_{c=1}^{C}u_{ci}=1$, m is a quantity controlling the clustering fuzziness, and $U\equiv[u_{ci}]$ is a $C\times N$ matrix called fuzzy partition matrix, which satisfies three conditions $u_{ci}\in[0,1]$ for all i and c, $\sum_{i=1}^{N}u_{ci}>0$ for all c, and $\sum_{c=1}^{C}u_{ci}=1$ for all i. The original FCM uses the probabilistic constraint that the memberships of a data point across all partitions sum to one. While this is useful in creation of partitions, it makes FCM very sensitive to outliers or noise. Krishnapuram et al. proposed the possibilistic c-Means (PCM) clustering algorithm by relaxing the normalized constraint of FCM [27]. PCM minimizes the following objective function for clustering:

$$J_{PCM}(T,V)=\sum_{c=1}^{C}\sum_{i=1}^{N}t_{ci}^{p}d_{ci}^{2}+\sum_{c=1}^{C}\eta_c\sum_{i=1}^{N}(1-t_{ci})^{p} \tag{2}$$

where t_{ci} is the possibilistic membership of x_i in cluster c, $T\equiv[t_{ci}]$ is a $C\times N$ matrix called possibilistic partition matrix, which satisfies two conditions $t_{ci}\in[0,1]$ for all i and c and $\sum_{i=1}^{N}t_{ci}>0$ for all c, p is a weighting exponent for the possibilistic membership, and η_c are suitable positive numbers. The first term of $J_{PCM}(T,V)$ demands that the distances from data points to the prototypes be as low as possible, whereas the second term forces t_{ci} to be as large as possible. PCM determines a possibilistic partition, in which a possibilistic membership measures the absolute degree of typicality of a point in a cluster. PCM is robust to outliers or noise, because a far away noisy point would belong to the clusters with small possibilistic memberships, and consequently it cannot affect the resulting clusters significantly. However, its performance depends highly on a good initialization and has the undesirable tendency to produce coincident clusters. Zhang et al. proposed the improved possibilistic c-Means (IPCM) with strong robustness and fast convergence rate [20]. IPCM integrates FCM into PCM, so that the improved algorithm can determine proper clusters via the fuzzy approach while it can achieve robustness via the possibilistic approach. IPCM partitions X into C fuzzy partitions by minimizing the following objective function:

$$J_{IPCM}(T,U,V)=\sum_{c=1}^{C}\sum_{i=1}^{N}u_{ci}^{m}t_{ci}^{p}d_{ci}^{2}+\sum_{c=1}^{C}\eta_c\sum_{i=1}^{N}u_{ci}^{m}(1-t_{ci})^{p} \tag{3}$$

Dealing with spherical clusters is the most important limitation of IPCM that can be eliminated by integration of IPCM with multiple kernels learning setting.

3. The proposed method

In the previous section, IPCM was explained as an efficient algorithm for data clustering dealing with partially overlapping clusters and noisy datasets. While IPCM is a popular soft clustering method, its effectiveness is largely limited to spherical clusters and cannot deal with complex data structures. Also, IPCM does not take side information into account to be used for constrained clustering. To provide a better clustering result, we propose a clustering algorithm that not only can deal with the linear inseparable and partially overlapping dataset, but also gets a better clustering accuracy under the noise interference. Also, the proposed algorithm is able to incorporate domain knowledge into the clustering algorithm to take advantage of the side information. To this aim, a new objective function is introduced to consider these issues.

To consider partially overlapped clusters and noise interference, the idea of IPCM is embedded into the new objective function. In order to take into account the side information, the objective function of IPCM is extended by two extra terms for the violation of the side information. Also, by applying multiple kernels setting, we attempt to address the problem of dealing with non-linear separable data, namely by mapping data with non-linear relationships to appropriate *feature spaces*. On the other hand, Kernel combination, or selection, is crucial for effective kernel clustering. Unfortunately, for most applications, it is not easy to find the right combination of the similarity kernels. Therefore, we propose a constrained multiple kernel improved possibilistic c-means (CMKIPCM) algorithm in which IPCM algorithm is extended to incorporate the side information with a multiple kernels learning setting. By incorporating multiple kernels and automatically adjusting the kernel weights, CMKIPCM is immune to ineffective kernels and irrelevant features. This makes the choice of kernels less crucial. Let $\psi(x)=\omega_1\psi_1(x)+\omega_2\psi_2(x)+\cdots+\omega_M\psi_M(x)$ be a non-negative linear combination of M base kernels in kernel space Ψ to map data to an implicit feature space. The proposed method minimizes the following objective function for constrained clustering:

$$\begin{aligned}J_{CMKIPCM}(\mathbf{w},T,U,V)=&\sum_{c=1}^{C}\sum_{i=1}^{N}u_{ci}^{m}t_{ci}^{p}(\psi(x_i)-v_c)^{T}(\psi(x_i)-v_c)\\&+\sum_{c=1}^{C}\eta_c\sum_{i=1}^{N}u_{ci}^{m}(1-t_{ci})^{p}\\&+\alpha\left(\sum_{(i,j)\in\mathcal{M}}\sum_{c=1}^{C}\sum_{\substack{l=1\\l\neq c}}^{C}u_{ci}^{m}u_{lj}^{m}t_{ci}^{p}t_{lj}^{p}\right.\\&\left.+\sum_{(i,j)\in\mathcal{C}}\sum_{c=1}^{C}u_{ci}^{m}u_{cj}^{m}t_{ci}^{p}t_{cj}^{p}\right)\end{aligned} \tag{4}$$

where $\mathcal{M}$ is the set of must-link constraints and $\mathcal{C}$ is the set of cannot-link constraints. $v_c\in\mathbb{R}^L$ is the center of c^{th} cluster in the implicit L-dimensional feature space and $V\equiv[v_c]_{L\times C}$ is a $L\times C$ matrix whose columns correspond to cluster centers. $\mathbf{w}=(\omega_1,\omega_2,\ldots,\omega_M)^T$ is a vector consisting of kernel weights, which satisfies the condition $\sum_{k=1}^{M}\omega_k=1$. $U\equiv[u_{ci}]_{C\times N}$ is the fuzzy membership matrix whose elements are the fuzzy memberships u_{ci} with similar conditions mentioned in FCM. $T \equiv [t_{ci}]_{C\times N}$ is the possibilistic membership matrix whose elements are the possibilistic memberships t_{ci} with similar conditions mentioned in PCM. $(\psi(x_i)-v_c)^T(\psi(x_i)-v_c)$ denotes the distance between data x_i and cluster center v_c in feature space. η_c is a scale parameter and is suggested to be [27]

$$\eta_c = \frac{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p (\psi(x_i)-v_c)^T(\psi(x_i)-v_c)}{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p}. \tag{5}$$

The first two terms in Eq. (4) enhance IPCM to multiple kernels IPCM and support the compactness of the clusters in the feature space. The first term is the sum of squared distances to the prototypes weighted by two fuzzy and possibilistic memberships and the second term forces the possibilistic memberships to be as large as possible to avoid trivial solution. The third term in Eq. (4) controls the costs of violating the pairwise constraints and is weighted by α, as relative importance of supervision. It is composed of two costs for violating pairwise must-link and cannot-link constraints. The first part in the third term measures the cost of violating the pairwise must-link constraints. It penalizes the presence of two such points in different clusters weighted by the corresponding membership values. On the other hand, the second part measures the cost of violating the pairwise cannot-link constraints and the presence of two such points in the same cluster is penalized by their membership values. By minimizing Eq. (4), the final partition will minimize the sum of intra-cluster distances in kernel space so that the specified constraints are respected as well as possible. The following theorem studies the necessary conditions for the objective function given in Eq. (4) to attain its minimum.

Theorem 1. *The J_{CMKIPCM} attains its local minima when $U \equiv [u_{ci}]_{C\times N}$, $T \equiv [t_{ci}]_{C\times N}$, and $\mathbf{w} \equiv [\omega_k]_{M\times 1}$ are assigned the following values.*

$$t_{ci} = \frac{1}{1+\left(\dfrac{D_{ci}^2+\alpha\left(S_{ci}^{\mathcal{M}}+S_{ci}^{\mathcal{C}}\right)}{\eta_c}\right)^{1/(p-1)}} \tag{6}$$

$$u_{ci} = \frac{1}{\sum\limits_{k=1}^{C}\left(\dfrac{t_{ci}^{p-1}\left(D_{ci}^2+\alpha\left(S_{ci}^{\mathcal{M}}+S_{ci}^{\mathcal{C}}\right)\right)}{t_{ki}^{p-1}\left(D_{ki}^2+\alpha\left(S_{ki}^{\mathcal{M}}+S_{ki}^{\mathcal{C}}\right)\right)}\right)^{1/(m-1)}} \tag{7}$$

$$\omega_k = \frac{\dfrac{1}{\mathcal{Y}_k}}{\dfrac{1}{\mathcal{Y}_1}+\dfrac{1}{\mathcal{Y}_2}+\ldots+\dfrac{1}{\mathcal{Y}_M}} \tag{8}$$

where $D_{ci}^2 = (\psi(x_i)-v_c)^T(\psi(x_i)-v_c)$, $S_{ci}^{\mathcal{M}} = \sum_{(i,j)\in\mathcal{M}}\sum_{l=1, l\neq c}^{C} u_{lj}^m t_{lj}^p$, $S_{ci}^{\mathcal{C}} = \sum_{(i,j)\in\mathcal{C}} u_{cj}^m t_{cj}^p$, $\mathcal{Y}_k = \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^m t_{ci}^p \mathcal{Q}_{ci}^k$, *and*

$$\mathcal{Q}_{ci}^k = \kappa_k(x_i,x_i) - \frac{2\sum_{j=1}^{N} u_{cj}^m t_{cj}^p \kappa_k(x_i,x_j)}{\sum_{j=1}^{N} u_{cj}^m t_{cj}^p} + \frac{\sum_{r=1}^{N}\sum_{s=1}^{N} u_{cr}^m t_{cr}^p u_{cs}^m t_{cs}^p \kappa_k(x_r,x_s)}{\left(\sum_{r=1}^{N} u_{cr}^m t_{cr}^p\right)\left(\sum_{s=1}^{N} u_{cs}^m t_{cs}^p\right)}. \tag{9}$$

The proof of Theorem 1 will be given later in Appendix. Theorem 1 states that when the above-mentioned conditions are held, the proposed objective function attains one of its local minima while a meaningful grouping of data is obtained.

3.1. The objective function from the multiple kernels perspective

Consider a set $\Phi = \{\phi_1, \phi_2, \ldots, \phi_M\}$ of M kernel mappings, where kernel ϕ_k maps $x \in \mathbb{R}^l$ into a L_k-dimensional column vector $\phi_k(x)$ in its feature space. Let $\{\kappa_1, \kappa_2, \ldots, \kappa_M\}$ be the Mercer kernels corresponding to these mappings as $\kappa_k(x_i,x_j) = \phi_k(x_i)^T\phi_k(x_i)$. Let $\acute{\phi}(x) = \sum_{k=1}^{M}\omega_k\phi(x)$, where $\omega_k \geq 0$ is the weight of ith kernel denotes the non-negative combination of these mappings. A linear combination of these mappings may be impossible because they do not necessarily have the same dimensionality. Hence, a new set of independent mappings, $\Psi = \{\psi_1, \psi_2, \ldots, \psi_M\}$, is constructed from the original Φ as given below.

$$\psi_1(x) = \begin{bmatrix} \phi_1(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \psi_2(x) = \begin{bmatrix} 0 \\ \phi_2(x) \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad \psi_M(x) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \phi_M(x) \end{bmatrix} \in \mathbb{R}^L \tag{10}$$

Each ψ converts x to a L-dimensional vector, where $L = \sum_{k=1}^{M} L_k$. Constructing new mappings in this way ensures that the feature spaces of these mappings have the same dimensionality and their linear combination is well defined. Independent mappings $\Psi = \{\psi_1, \psi_2, \ldots, \psi_M\}$ form a new set of orthogonal bases as given below.

$$\psi_k(x_i)^T\psi_{k'}(x_j) = \begin{cases} \kappa_k(x_i,x_j) & k = k' \\ 0 & k \neq k' \end{cases} \tag{11}$$

The objective function given in Eq. (4) tries to find a non-negative linear combination $\psi(x) = \sum_{k=1}^{M}\omega_k\psi_k(x)$ of M bases in Ψ to map the input data into an implicit feature space.

3.2. Description of the algorithm

CMKIPCM is fully summarized in Algorithm 1. It starts by initializing a possibilistic and fuzzy membership matrices T^0 and U^0 using IPCM at time $\mathcal{T} = 0$. Instead of feeding CMKIPCM with a preselected set of constraints, it chooses the most informative constraints considering the current state $\mathcal{T}$. Therefore, at each iteration $\mathcal{T}$, $T^{\mathcal{T}}$ and $U^{\mathcal{T}}$ are fed to Procedure **GetConstraints**(.) to return λ informative constraints. If the total number of constraints exceeds the allowed number of queries λ_{total}, it returns an empty set of constraints (details will be given in Section 4). At each iteration, the optimal weights are calculated by fixing the fuzzy and possibilistic memberships. The optimal possibilistic and fuzzy memberships are then updated assuming fixed weights. The process is repeated until a specified convergence criterion is satisfied.

Algorithm 1. Constrained multiple kernels improved possibilistic c-means (CMKIPCM). Given a set of N data points $X = \{x_i\}_{i=1}^{N}$, the desired number of clusters C, the set of base kernels $\{\kappa\}_{k=1}^{M}$, the number of constraints to be queried at each iteration λ, the total number of allowed constraints λ_{total}, and the outliers and noise threshold θ, **output** the fuzzy membership matrix $U \equiv [u_{ci}]_{C\times N}$.

```
procedure CMKIPCM $X, C, \{\kappa\}_{k=1}^{M}, \lambda, \lambda_{total}, \theta$
    Initialize the fuzzy and possibilistic membership matrices $U^0$ and $T^0$.
    $\mathcal{T} \leftarrow 0$
    $\mathcal{M}^{(\mathcal{T})} \leftarrow \varnothing$    ▹ Set of must-link constraints at iteration $\mathcal{T}$
    $\mathcal{C}^{(\mathcal{T})} \leftarrow \varnothing$    ▹ Set of cannot-link constraints at iteration $\mathcal{T}$
    for c=1,..,C do    ▹ Estimate $\eta_1, \eta_2, \ldots, \eta_C$
        $\eta_c \leftarrow K\frac{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p D_{ci}^2}{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p}$
    end for
    repeat
        $\mathcal{T} \leftarrow \mathcal{T}+1$
        if $|\{\mathcal{M}^{(\mathcal{T}-1)}, \mathcal{C}^{(\mathcal{T}-1)}\}| < \lambda_{total}$ then    ▹ Choose $\lambda$ constraints at iteration $\mathcal{T}$
            $\{\mathcal{M}^{(\mathcal{T})}, \mathcal{C}^{(\mathcal{T})}\} \leftarrow \{\mathcal{M}^{(\mathcal{T}-1)}, \mathcal{C}^{(\mathcal{T}-1)}\} \cup$ GetConstraints $(X, U^{\mathcal{T}}, T^{\mathcal{T}}, C, \theta, \lambda)$
    else
        $\{\mathcal{M}^{(\mathcal{T})}, \mathcal{C}^{(\mathcal{T})}\} \leftarrow \{\mathcal{M}^{(\mathcal{T}-1)}, \mathcal{C}^{(\mathcal{T}-1)}\}$
    end if
    $\alpha^{(\mathcal{T})} \leftarrow \frac{N\sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^{m} t_{ci}^{p} D_{ci}^{2}}{(|\mathcal{M}^{(\mathcal{T})}| + |\mathcal{C}^{(\mathcal{T})}|)\sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^{m} t_{ci}^{p}}$
        ▹ Update the importance degree of supervision
    for k=1,..,M do    ▹ Update kernel weights
        $\omega_k^{(\mathcal{T})} \leftarrow \frac{\frac{1}{\mathcal{Y}_k}}{\frac{1}{\mathcal{Y}_1} + \frac{1}{\mathcal{Y}_2} + \cdots + \frac{1}{\mathcal{Y}_M}}$
    end for    ▹ Use Eq. (37) to compute $\mathcal{Y}_k$
    for c=1,..,C do    ▹ Update possibilitstic memberships
        for i=1,..,N do
            $t_{ci}^{(\mathcal{T})} \leftarrow \frac{1}{1 + \left(\frac{D_{ci}^2 + \alpha^{(\mathcal{T})}(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}})}{\eta_c}\right)^{1/(p-1)}}$
        end for    ▹ Use Eq. (18) to compute $S_{ci}^{\mathcal{M}}$ and $S_{ci}^{\mathcal{C}}$ and
    Eq. (35) to compute $D_{ci}^2$
    end for
    for c=1,..,C    ▹ Update fuzzy memberships
        for i=1,..,N do
            $u_{ci}^{(\mathcal{T})} \leftarrow \frac{1}{\sum_{k=1}^{C}\left(\frac{t_{ci}^{p-1}\left(D_{ci}^2 + \alpha^{(\mathcal{T})}\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right)}{t_{ki}^{p-1}\left(D_{ki}^2 + \alpha^{(\mathcal{T})}\left(S_{ki}^{\mathcal{M}} + S_{ki}^{\mathcal{C}}\right)\right)}\right)^{1/(m-1)}}$
        end for
    end for
  until $\|U^{\mathcal{T}-1} - U^{\mathcal{T}}\| < \epsilon$
  return $U^{\mathcal{T}}$    ▹ return fuzzy membership matrix $U^{\mathcal{T}}$
end procedure
```

4. Active selection of clustering constraints

Although CMKIPCM can be fed by a preselected set of clustering constraints, it will be better if it chooses non-redundant informative constraints considering its current state. As we know, CMKIPCM updates the possibilistic memberships $T \equiv [t_{ci}]_{C\times N}$ at each iteration $\mathcal{T}$, which can be used to avoid querying unhelpful constraints (noise and outliers). To this aim, data points x_i with $\max(t_{ci}) < \theta$, (for $c = 1, \dots, C$) are ignored for the future constraint selection. θ is a user defined threshold for noise and outliers.

Also CMKIPCM utilizes the measurement of Type-II mistake in clustering to select non-redundant informative constraints. Type-II mistake is the mistake of classifying data from a same class into different clusters and is used to measure the overlap degree for each cluster. Given a fuzzy partitioning of data, the idea is to find the most overlapped cluster (cluster with the most value of Type-II mistake) and query the pairwise relation of its ambiguous members to the most certain member of other clusters. Let $\tilde{F} = \{\tilde{F}_1, \tilde{F}_2, \dots, \tilde{F}_C\}$ be C fuzzy partitions of a dataset and $F_c = \{x_i \in X_{t_{ci} \geq \theta} : u_{ci} \geq u_{c'i}, \forall\ 1 \leq c' \leq C, c \neq c'\}$ be the corresponding crisp set of fuzzy cluster $\tilde{F}_c$, where $X_{t_{ci} \geq \theta} = \{x_i \in X : max(t_{ci}) \geq \theta, \forall c : 1, \dots, C\}$ (see Fig. 1) . The overlap degree between two fuzzy partitions $\tilde{F}_c$ and $\tilde{F}_{c'}$ is defined as

$$O(\tilde{F}_c, \tilde{F}_{c'}) = \frac{\sum_{x_i \in F_c \cup F_{c'}} \min(u_{ci}, u_{c'i})^2}{\sum_{x_i \in F_c \cup F_{c'}} \min(u_{ci}, u_{c'i})}. \tag{12}$$

Smaller value of $O(\tilde{F}_c, \tilde{F}_{c'})$ indicates smaller possibility of existing Type-II mistakes. By averaging the overlap degree of the cluster pairs, we define the inter-cluster overlap for each fuzzy partition $\tilde{F}_c$ as

$$O(\tilde{F}_c) = \frac{1}{C-1} \sum_{\substack{c'=1 \\ c' \neq c}}^{C} O(\tilde{F}_c, \tilde{F}_{c'}). \tag{13}$$

Let $\tilde{F}_m$ be the fuzzy partition with the most inter-cluster overlap $O(\tilde{F}_m)$ and λ be the maximum allowed queries that CMKIPCM is allowed to ask at each iteration. The proposed method distributes λ constraints between the most overlapped cluster $\tilde{F}_m$ and other clusters $\tilde{F}_c$ (for $c = 1, 2, \dots, C$ & $c \neq m$) according to the normalized ambiguity degree between them that is

$$q_c = \lambda \frac{O(\tilde{F}_c, \tilde{F}_m)}{\sum_{\substack{c'=1 \\ c' \neq c}}^{C} O(\tilde{F}_{c'}, \tilde{F}_m)}. \tag{14}$$

To choose q_c constraints between partitions $\tilde{F}_m$ and $\tilde{F}_c$, the data point $x_a \in F_m \cup F_c$ with the maximum ambiguity $\min(u_{ma}, u_{ca})$ is determined and is queried against $x_m \in F_m$ with the highest u_{mm} and $x_c \in F_c$ with the highest u_{cc}. This process is repeated until λ constraints are selected between F_m and F_c. Fig. 1(e) shows the process of constraint selection for four typical clusters $\{F_1, \dots, F_4\}$ with $F_m = 4$ as the most overlapped cluster. The resulting active constraint selection method is summarized as Algorithm 2.

Algorithm 2. Active selection of clustering constraints. Given a set of N data points $X = \{x_i\}_{i=1}^{N}$, the desired number of clusters C, the number of constraints λ, the fuzzy membership matrix $U \equiv [u_{ci}]_{C\times N}$, the possibilistic membership matrix $T \equiv [t_{ci}]_{C\times N}$, and the outliers and noise threshold θ, **output** set of must-links constraints $\mathcal{M}$ and cannot-link constraints $\mathcal{C}$.

```
procedureGetConstraints $X, U, T, C, \theta, \lambda$
    $\mathcal{M} \leftarrow \varnothing$
    $\mathcal{C} \leftarrow \varnothing$
    $X_{t_{ci} \geq \theta} \leftarrow \{x_i \in X : max(t_{ci}) \geq \theta, \forall c : 1, \dots, C\}$    ▹ Eliminate
  noise and outliers
    for c=1,…,$C$ do    ▹ Determine the crip set for each
  fuzzy partition $\tilde{F}_c$
        $F_c \leftarrow \{x_i \in X_{t_{ci} \geq \theta} : u_{ci} \geq u_{c'i}, \forall\ 1 \leq c' \leq C, c \neq c'\}$
    end for
    for $c = 1, \dots, C$ do    ▹ Compute the inter-cluster overlap
  for each fuzzy partition
        $O(\tilde{F}_c) = \frac{1}{C-1}\sum_{\substack{c'=1 \\ c' \neq c}}^{C} O(\tilde{F}_c, \tilde{F}_{c'})$
    end for
    $\tilde{F}_m \leftarrow \arg\max_{\tilde{F}_c:1,\dots,\tilde{F}_C} O(\tilde{F}_c)$    ▹ Determine the most
  overlapped cluster
    for c = 1, …, $C, c \neq m$ do    ▹ Compute the number of
  candidate queries between $\tilde{F}_m$ and $\tilde{F}_c \neq \tilde{F}_m$
        $q_c = \lambda \frac{O(\tilde{F}_c, \tilde{F}_m)}{\sum_{\substack{c'=1 \\ c' \neq c}}^{C} O(\tilde{F}_{c'}, \tilde{F}_m)}$
    end for
    for c = 1, …, $C, c \neq m$ do    ▹ Query constraint between
  $\tilde{F}_m$ and $\tilde{F}_c \neq \tilde{F}_m$
        $x_m \leftarrow \arg\max_{x_j \in F_m} u_{mj}$
```

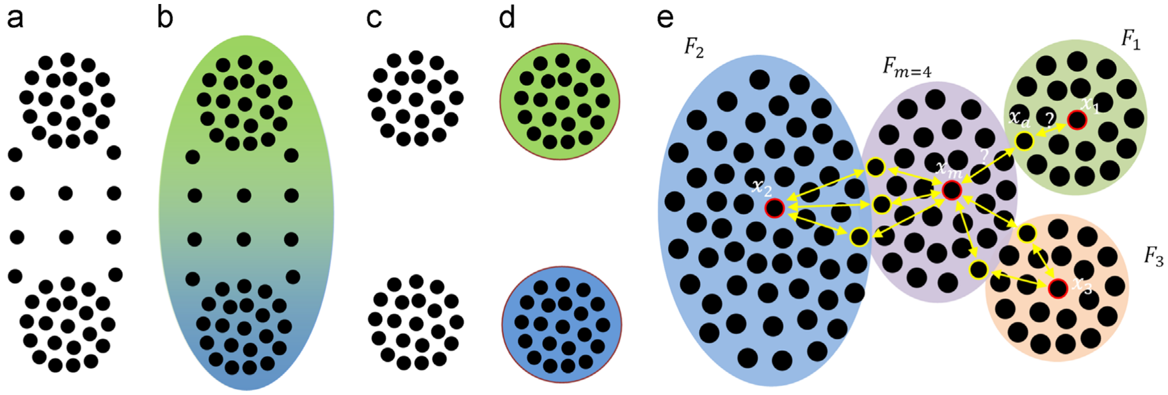

Fig. 1. (a) Synthetic data set X in $\mathbb{R}^2$, (b) fuzzy partitioning $\{\tilde{F}_1, \tilde{F}_2\}$ of X, (c) resultant $X_{t_{ci} \geq \theta}$ by ignoring data points $x_i \in X$ with $max(t_{ci}) < \theta, \forall c : 1, 2$, (d) $\{F_1, F_2\}$ as corresponding crisp set of fuzzy clusters $\{\tilde{F}_1, \tilde{F}_2\}$, and (e) Constraint selection for four typical clusters $\{F_1, \ldots, F_4\}$ with $F_{m=4}$ as the most overlapped cluster. The red outlined points indicate the least ambiguous member for each cluster and the yellow outlined points indicate the most ambiguous members between cluster pairs $(F_{m=4}, F_i) : i = 1, 2, 3$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

```
    $x_c \leftarrow \arg\max_{x_j \in F_c} u_{cj}$
    $\mathcal{A} \leftarrow \varnothing$
    for i = $1, \ldots, q_c$ do
      $x_a \leftarrow \arg\max_{x_j \in F_m \cup F_c, x_j \notin \mathcal{A}} min(u_{mj}, u_{cj})$
      $\mathcal{A} \leftarrow \mathcal{A} \cup x_a$
      Query the user about the $Label_{am}$ of $(x_a, x_m)$?
      if $Label_{am}$ is ML then
        $\mathcal{M} \leftarrow \mathcal{M} \cup (x_a, x_m)$
      else
        $\mathcal{C} \leftarrow \mathcal{C} \cup (x_a, x_m)$
      end if
      Query the user about the $Label_{ac}$ of $(x_a, x_c)$?
      if $Label_{ic}$ is ML then
          $\mathcal{M} \leftarrow \mathcal{M} \cup (x_a, x_c)$
      else
        $\mathcal{C} \leftarrow \mathcal{C} \cup (x_a, x_c)$
      end if
    end for
  end for
  return $\{\mathcal{M}, \mathcal{C}\}$      ▹ return $\lambda$selected constraints
end procedure
```

5. Experiments

An extensive set of experiments are designed to address the following questions in the proposed method. How do different types of kernels affect the efficiency of clustering? Whether our proposed method improves the accuracy compared to existing unsupervised clustering algorithms? How does our model perform in comparison with the other semi-supervised clustering algorithms? How much efficient is our proposed active constraint selection model? Whether constraints selected by the proposed active constraint selection model are informative for other semi-supervised clustering algorithms or not? Next section answers these questions by conducting experiments on some real-world datasets.

5.1. Experimental setup

In this section, we describe datasets, evaluation metric, base kernels, and comparative experiments. The weighting exponent p and m for the possibilistic and fuzzy membership are set to 2 and the convergence threshold ϵ is set to 10^{-3} for all experiments. The number of clusters C is set as the ground truth for all datasets. Two input parameters θ as the noise and outlier threshold and λ as the number of selected constraints in each iteration are set to 0.1 and 5, respectively. These values are considered as default values for CMKIPCM unless specified. The input parameter α in Eq. (4) should also be assigned to ensure that the impact of the constraints is not ignored at each iteration. Hence, it is chosen so that the supervision term (the third term in Eq. (4)) be of the same order of magnitude as two first additive terms in Eq. (4). Thus, the value of α at each iteration $\mathcal{T}$ is approximated as

$$\alpha^{(\mathcal{T})} = \frac{N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci}^{m} t_{ci}^{p} D_{ci}^{2}}{\left(|\mathcal{M}^{(\mathcal{T})}| + |\mathcal{C}^{(\mathcal{T})}|\right) \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci}^{m} t_{ci}^{p}}, \qquad (15)$$

where $|\mathcal{M}^{(\mathcal{T})}|$ and $|\mathcal{C}^{(\mathcal{T})}|$ are the cardinality of must-link and cannot-link sets at iteration $\mathcal{T}$.

5.1.1. Data collection

In order to show the efficiency of the proposed method on real datasets, experiments are conducted on some datasets from UCI Machine Learning Repository[1] (each with the following number of objects / attributes/and clusters): Iris (150/4/3), Glass Identification (214/9/6), Breast Cancer Wisconsin (Diagnostic) (569/30/2), Soybean (47/34/4), Ionosphere (354/31/2), Ecoli (336/7/8), Wine (178/13/3), Sonar (208/60/2), Heart (270/13/2), Balance Scale (625/4/3), Letter Recognition (A,B) (1555/16/2), Libras Movement (360/90/15) and Pima Indians Diabetes (768/8/2). These datasets were chosen because they have already been used in constrained clustering papers.

5.1.2. Evaluation metric

There are many clustering measures for evaluating results of clustering. The well-known *Adjusted Rand Index* (ARI) [28] is used in all experiments to evaluate the agreement between the theoretical partition of each dataset and the output partition of the evaluated algorithms. As we know, CMKIPCM results in fuzzy membership matrix $U \equiv [u_{ci}]_{C \times N}$ as a membership degrees of data points to the clusters. These membership degrees must be converted to hard assignments to be used by ARI. To this aim, we assign each data item to the cluster with the highest membership degree. The interested reader is referred to [29] for a more detailed review of cluster validity indices.

5.1.3. Base kernels selection

Different families of kernel functions with different numbers can be used to construct the base kernel set. In this paper, three different categories of functions are used as base kernels: (1) Kernel functions constructed by the spectral information of

[1] http://archive.ics.uci.edu/ml/

data, (2) Gaussian kernel functions, and (3) polynomial kernels. Let $X=[x_1,x_2,\ldots,x_N]_{l\times N}$ be a $l\times N$ matrix, where each column of X contains a feature vector in $\mathbb{R}^l$ and let $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_N]_{N\times N}$ be eigenvectors of linear kernel X^TX. In the first category of kernel function, $M_{\mathbf{v}}$ kernel matrices $\{\kappa_1^{\mathbf{v}},\kappa_2^{\mathbf{v}},\ldots,\kappa_{M_{\mathbf{v}}}^{\mathbf{v}}\}$ are constructed by setting $\kappa_k^{\mathbf{v}}=\mathbf{v}_k^T\mathbf{v}_k$ for $k=1,2,\ldots,M_{\mathbf{v}}$.

In the second one, $M_{\mathbf{g}}$ kernel matrices $\{\kappa_1^{\mathbf{g}},\kappa_2^{\mathbf{g}},\ldots,\kappa_{M_{\mathbf{g}}}^{\mathbf{g}}\}$ are constructed by mapping data with Gaussian kernels $\kappa_k^{\mathbf{g}}(x_i,x_j)=\exp(-(x_i-x_j)^T(x_i-x_j)/2^{(M_{\mathbf{g}}-k)}\sigma_X)$ for $k=1,2,\ldots,M_{\mathbf{g}}$, where σ_X is the standard deviation of all $\binom{N}{2}$ pairwise distances among points in dataset $X=\{x_i\}_{i=1}^N$. The coefficient $2^{(M_{\mathbf{g}}-k)}$ in the denominator gives different widths to these kernel functions. Polynomial functions are used for the last category of kernel mapping that is $\kappa_k^{\mathbf{p}}(x_i,x_j)=(x_i^Tx_j+c)^k$ for $k=1,2,\ldots,M_{\mathbf{p}}$. c is set to 0 in all experiments. So, we have $M=M_{\mathbf{v}}+M_{\mathbf{g}}+M_{\mathbf{p}}$ base kernels $\{\kappa_1^{\mathbf{v}},\kappa_2^{\mathbf{v}},\ldots,\kappa_{M_{\mathbf{v}}}^{\mathbf{v}},\kappa_1^{\mathbf{g}},\kappa_2^{\mathbf{g}},\ldots,\kappa_{M_{\mathbf{g}}}^{\mathbf{g}},\kappa_1^{\mathbf{p}},\kappa_2^{\mathbf{p}},\ldots,\kappa_{M_{\mathbf{p}}}^{\mathbf{p}}\}$ as the base kernel set.

5.2. Comparative experiments

Three experiments are conducted to study the performance of the proposed method. The first experiment compares CMKIPCM with two unsupervised clustering algorithms: (1) agglomerative hierarchical clustering (AHC) and (2) classical K-Means in terms of improvement of the results. In the second one, the proposed method is compared to MPCKMeans [3] and Xiang's method [8] as two well-known constrained clustering algorithms. Because of the sensitivity of K-Means, MPCKMeans and CMKIPCM to the initialization, their performances are averaged over 50 runs for each experiment. Also, the random selection of constraints is repeated 50 times and the performance is reported on the average. In order to see what the proposed active constraint selection heuristic brings in terms of improvement of the results, experiments are conducted to report the average performance of CMKIPCM when the constraints are selected actively (CMKIPCM-Active). Three MPCKMeans, Xiang and CMKIPCM algorithms are fed with a same set of constraints in all experiments. For MPCKMeans and Xiang methods, the maximum accuracy achieved by using actively or randomly selected constraints are reported in all experiments. The kernel set $\{\kappa_1^{\mathbf{v}},\kappa_2^{\mathbf{v}},\ldots,\kappa_5^{\mathbf{v}},\kappa_1^{\mathbf{g}},\kappa_2^{\mathbf{g}},\ldots,\kappa_7^{\mathbf{g}}\}$ is used in all experiments as the base kernel set. Figs. 2 and 3 show the clustering accuracy for some UCI datasets. For each dataset and number of query, the average and standard deviation of ARI are plotted in these figures. Fig. 3(f) plots the mean and standard deviation of the optimal weights corresponding to each kernel for all datasets. As shown in these figures, different kernels take different weights for each dataset. Discussion about the effect of different set of kernels will be given in Section 6.

It can be observed from Figs. 2 and 3 that CMKIPCM generally outperforms other clustering algorithms (K-Means,, AHC, Xiang, and MPC-KMeans). As these figures show, the proposed method not only makes better results when it chooses constraints actively, but also outperforms when it uses random constraints. This shows the reliability of CMKIPCM even if the constraints are selected at random. The superiority of the proposed method is considerably observable for Balance, Ecoli, Ionosphere, Soybean, Sonar, and Letter (A,B) datasets.

It is also interesting to notice that CMKIPCM makes a great improvement in comparison with K-Means without considering how the constraints are queried, while AHC, Xiang, and MPCKMeans algorithms drop into accuracy level lower than K-Means in some datasets (Ionosphere, Glass, Letter(A,B)). This is remarkable for AHC in Iris, Balance, Ionosphere, Heart, Glass, and Wine datasets. For MPCKmeans, drops in efficiency are in Iris, Balance, Ionosphere, and Glass datasets and for Xiang method, this happened in Balance, Ionosphere, and Glass datasets. This can be explained by the non-linear property of CMKIPCM utilized by multiple kernel learning, which can discover complex relationships among data points, while K-Means algorithm suffers from lack of this property.

It should also be mentioned in comparison between CMKIPCM and AHC that the proposed method provides more dominant results than AHC in all experiments. It can be concluded from the results that AHC is not efficient enough to cluster complex datasets specially high dimensional datasets like Sonar (208/60/2) and Breast (569/30/2). It is because AHC does not utilize any cues to consider non-linearity in datasets.

In comparison with MPCKMeans that generally outperforms both K-Means and AHC algorithms, the proposed method obtains better results especially in Balance, Ecoli, Ionosphere, Glass, Sonar, and Letter (A,B) datasets. For higher dimensional datasets such as Ionosphere and Sonar this improvement is remarkable. Although MPCKmeans learns metrics during clustering, the experiments show that MPCKMeans does not progress in correct metrics learning by increasing the number of constraints. This is evident in Glass, Heart, Balance, Ecoli, Ionosphere, and Letter (A,B) datasets. As shown in these figures, MPCKmeans results in a stable non increasing level of accuracy when the side information exceeds a threshold (e.g when the number of constraints is greater than 90 in Balance dataset). On the other hand, CMKIPCM results in a nearly smooth increase in accuracy, which implies the usefulness of the newly added constraints along with the high learning capability of CMKIPCM.

The proposed method significantly outperforms Xiang method except in Wine and Heart datasets with small number of constraints. On the other hand, CMKIPCM considerably surpasses Xiang method when the number of queries increases (e.g when the number of constraints is greater than 100 in Heart dataset). Although Xiang algorithm has kept the accuracy in a level greater than both K-Means and AHC algorithms, it has failed to improve the learnt metric and consequently the clustering accuracy by increasing the number of constraints. This is specially noticeable in Wine, Glass, Ecoli, Ionosphere, Soybean, Sonar, and Letter (A,B) datasets.

Local decrease in accuracy with increasing the number of queries is a well-known issue in constrained clustering [12]. This indicates the inefficiency of clustering algorithm in maximum utilization of available side information or there is an inconsistency among the selected constraints. This issue is considerably observable in Iris, Ecoli, Breast, Ionosphere, Glass, and Letter (A,B) datasets for MPCKMeans algorithm and Breast, Ionosphere, Heart, Sonar, and Letter (A,B) datasets for Xiang algorithm. On the other hand, the proposed method does not encounter this issue when the constraints are queried actively or chosen in a random manner. In all datasets, the proposed method results in a smooth increase in accuracy. It means that CMKIPCM utilizes the constraint information more than the other clustering algorithms without considering how the constraints are selected.

Applicability of the proposed method is another important issue, which should be considered. Although MPCKmeans and Xiang methods sometimes increase the accuracy of clustering algorithms compared with K-Means and AHC, they are inefficient when the number of constraints increases because they cannot utilize all information of existing side information. Also, considerable local drops in accuracy is another issue that affect the applicability of these algorithms. However, the proposed method can be a better option for constrained clustering in real-world applications because it demonstrated more improvement than the other methods.

A paired t-test with significance level 0.05 was conducted to significantly evaluate the results shown in Figs. 2 and 3 for a specified number of constraints, $\lambda_{total}=150$, on each dataset. Table 1 summarizes the comparison of the results using a paired t-test. We use the relation $x<y$ to state that the ARI values of the

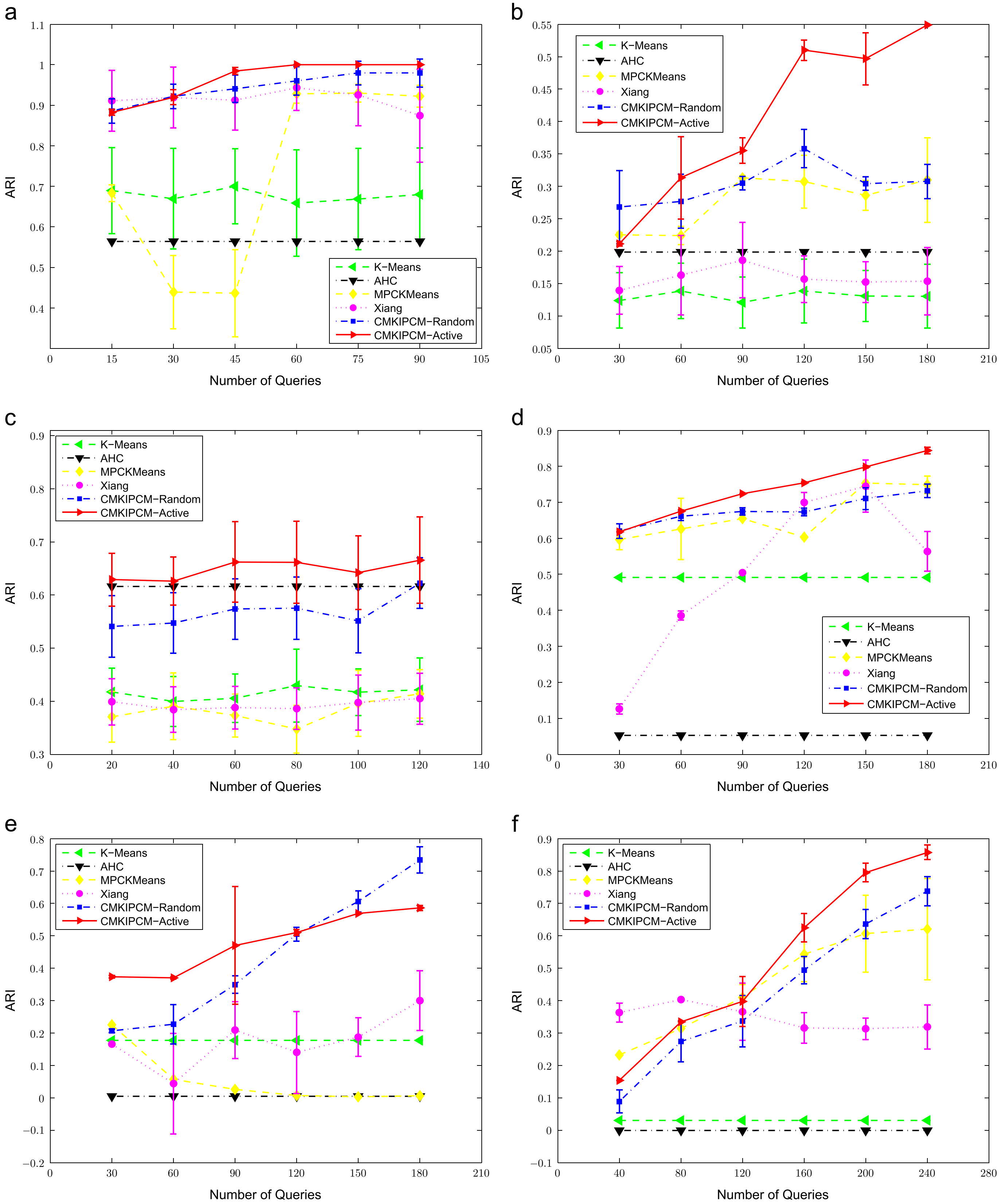

Fig. 2. Performance comparison of CMKIPCM in conjunction with two actively and randomly selected constraints with K-Means, AHC, Xiang [8], and MPCKMeans [3] clustering algorithms. (a) Iris. (b) Balance. (c) Ecoli. (d) Breast. (e) Ionosphere (f) Heart.

latter method are significantly higher than those of the former one. Similarly, $x \sim y$ denotes that the results of two methods are not significantly different for the given confidence level. As the paired t-test shows, CMKIPCM-Active generally outperforms all the other methods with a 95% confidence level. Moreover, CMKIPCM-Random is the second best method in most cases.

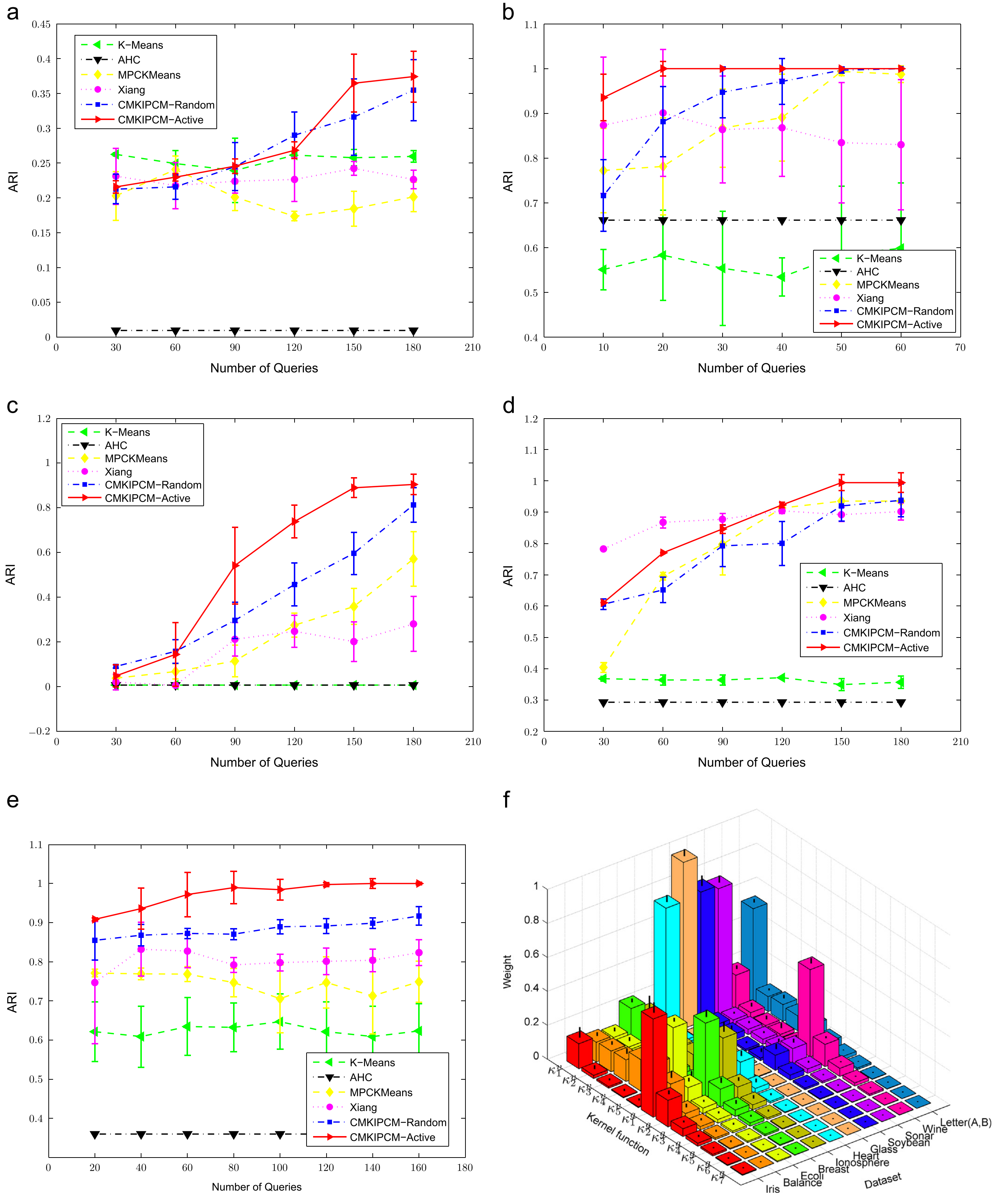

Fig. 3. Performance comparison of CMKIPCM in conjunction with two actively and randomly selected constraints with K-Means, AHC, Xiang [8], and MPCKMeans [3] clustering algorithms. (a) Glass. (b) Soybean. (c) Sonar. (d) Wine. (e) Letter(A,B). (f) Weight plot.

5.3. *Impact of the actively selected constraints*

This experiment demonstrates the efficiency of the proposed active query selection heuristic on informative query selection. As discussed in Section 5.2, CMKIPCM efficiently groups data when it chooses constraints randomly. It provides better results when the constraints are selected by the proposed active constraint selection heuristic. This is remarkable in Balance, Ecoli, Heart, Glass, Sonar, and Letter (A,B) datasets. These actively selected constraints not only increase the performance of CMKIPCM but also make MPCKMeans and Xiang methods more efficient than when the random constraints are used. Investigating this issue, experiment was conducted to compare the performance of MPCKMeans and Xiang algorithms in conjunction with 100 constraints selected by the proposed active query selection and random heuristics. Fig. 4 illustrates the result of this experiment.

A remarkable point about the efficiency of the actively selected constraints is that for $\lambda_{total} > 120$, they could not provide information more than random constraints in Ionosphere dataset (see Fig. 2(e)). This is because when the number of actively chosen constraints exceeds 120, CMKIPCM converges to a local minima and any extra constraints will be similar to already selected ones and give no more information about the true clustering of data. In contrast, when the constraints are chosen randomly, extra constraints help CMKIPCM to escape from a local minima. For more results and experiments, refer to [30].

5.4. Time complexity analysis

In this section, the computational complexity of the proposed method is studied. Also, the same is done for some state of the art and all compared approaches when they are implemented in basic mode by assigning default values to parameters. Let N be the number of data points, l be the dimension of data in input space, C be the number of cluster, and $\mathcal{T}_{max}$ be the number of iterations needs for an iterative algorithm to converge, λ_{total} be the number of allowed constraints, and M be the number of kernel matrices used in CMKIPCM.

The time complexity of K-Means is $O(NCl\mathcal{T}_{max})$ and the time complexity of FCM is $O(NC^2\, l\mathcal{T}_{max})$ [31]. K-harmonic means algorithm (KHM) that uses the harmonic mean of distance from each data point to all centers has the computational cost $O(NCl\mathcal{T}_{max}))$ [32]. In the general case, the complexity of agglomerative hierarchical clustering (AHC) is $O(N^3)$. However, for some special cases, optimal efficient agglomerative methods of complexity $O(N^2)$ are known.

MPCKMeans takes $O(NCl^3 + \lambda_{total} l^3)$ to assign each data point to its closest cluster, $O(Nl)$ to calculate the mean of C clusters, and $O(Nl + \lambda_{total} l)$ to update distance metric at each iteration. Hence, the time complexity of MPCKMeans is $O((NCl^3 + \lambda_{total} l^3)\mathcal{T}_{max})$ [3]. RCA takes $O(\lambda_{total} l^2)$ to compute the covariance matrix of data-points in chunklets, $O(l^3)$ to compute the whitening transformation matrix, and $O(Nl^2)$ to apply the whitening transformation matrix to the original data points. So, the time complexity of RCA is $O(\lambda_{total} l^2 + l^3 + Nl^2)$ [33].

Xiang's method solves an Eigen decomposition problem in an iterative manner until a specified convergence criterion is satisfied. It takes $O(\lambda_{total} l^2)$ to compute the covariance matrix of data points involved in constraints and $O(l^3)$ to solve the Eigen decomposition problem at each iteration. Finally it takes $O(Nl^2)$ to apply the optimal

Table 1
Comparison of the clustering results shown in Figs. 2 and 3 using a paired t-test.

Data set	Paired t-test										
Iris	AHC	<	K-Means	<	Xiang	<	MPCKmeans	<	CMKIPCM-Random	~	CMKIPCM-Active
Balance	K-Means	<	Xiang	<	AHC	<	MPCKmeans	~	CMKIPCM-Random	<	CMKIPCM-Active
Ecoli	MPCKmeans	~	Xiang	~	K-Means	<	AHC	~	CMKIPCM-Random	<	CMKIPCM-Active
Breast	AHC	<	K-Means	<	Xiang	~	MPCKmeans	~	CMKIPCM-Random	<	CMKIPCM-Active
Ionosphere	AHC	~	MPCKmeans	<	K-Means	~	Xiang	<	CMKIPCM-Random	~	CMKIPCM-Active
Heart	AHC	<	K-Means	<	Xiang	<	CMKIPCM-Random	~	MPCKmeans	<	CMKIPCM-Active
Glass	AHC	<	MPCKmeans	<	K-Means	~	Xiang	<	CMKIPCM-Random	<	CMKIPCM-Active
Soybean	K-Means	<	AHC	<	Xiang	<	MPCKmeans	~	CMKIPCM-Random	~	CMKIPCM-Active
Sonar	K-Means	~	AHC	<	Xiang	<	MPCKmeans	<	CMKIPCM-Random	<	CMKIPCM-Active
Wine	AHC	<	K-Means	<	Xiang	~	CMKIPCM-Random	~	MPCKmeans	<	CMKIPCM-Active
Letter(A,B)	AHC	<	K-Means	<	MPCKmeans	<	Xiang	<	CMKIPCM-Random	<	CMKIPCM-Active

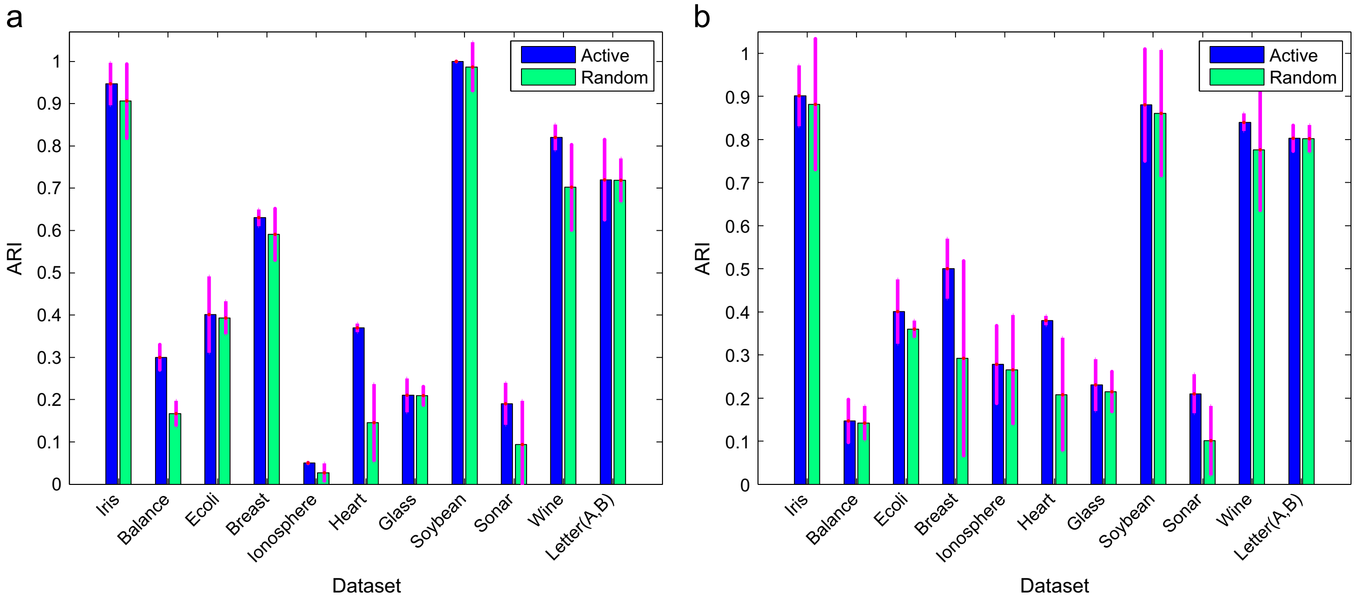

Fig. 4. Performance comparison of MPCKMeans and Xiang clustering methods in conjunction with 100 constraints selected by the proposed active selection and random heuristics. (a) MPCKMeans. (b) Xiang.

transformation matrix to the original data points. So, the time complexity of Xiang's method is $O(i_\epsilon(\lambda_{total}l^2+l^3)+Nl^2)$ where i_ϵ is the number of iterations to achieve the specified error bound [8]. Soleymani's method extends Xinag's method by preserving the topological structure of data during metric learning using the idea of locally linear embedding (LLE). This adds Xiang's method the complexity $O(lN^2)$ of computing nearest neighbors, $O(lNK^3)$ of solving a constrained least-squares problem to compute reconstruction weights, and $O(dN^2)$ of computing Eigen vectors (K and d are the number of nearest neighbors for each data point and dimension of embedding, respectively). Hence, the complexity of Soleymani's method is $O(i_\epsilon(\lambda_{total}l^2+l^3)+Nl^2+lN^2+lNK^3+dN^2)$ where i_ϵ is the number of iterations to achieve the specified error bound [5].

CMKIPCM takes $O(N^2CM)$ to compute distance between N data points and the center of C clusters, $O(NC^2\lambda_{total})$ to update the possibilitstic and fuzzy memberships, and $O(N^2CM)$ to update the weights at each iteration. So, the time complexity of CMKIPCM with a preselected set of constraints is $O((N^2CM+NC^2\lambda_{total})\mathcal{T}_{max})$. When CMKIPCM chooses constraints actively, λ constraints are selected at each iteration until the total number of selected constraints does not exceed the allowed number of queries λ_{total}. Selection of λ constraints takes $O(NC^2+\lambda N)$ at each iteration. So, the time complexity of CMKIPCM-Active will be $O((\mathcal{T}_{max}-\lambda_{total}/\lambda)(N^2CM+NC^2\lambda_{total})+\lambda_{total}/\lambda(N^2CM+NC^2\lambda_{total})(NC^2+\lambda N))$. Table 2 summarizes the computational complexity of the above mentioned methods. Fig. 5 shows the computation time (second) of CMKIPCM and compared approaches for Sonar and Wine datasets. In this figure, CMKIPCM is compared with the other methods when different numbers of queries λ_{total} are selected and plotted in horizontal axis. As this figure shows, the computational cost of CMKIPCM increases with more slope rather than the other methods. This is because that the computational cost of CMKIPCM increase in near polynomial degree of λ_{total}. Also, it can be observed that CMKIPCM takes much time when the constraints are selected actively because of time overhead coming from active constraints selection.

6. Discussion

Given a dataset, we do not know in advance which kernel set will perform better for it, and there is no common set of suitable kernels for all datasets. If we use a common set of kernels in all experiments, it will have better performance for some datasets but perform worse for some others. All experiments given in Section 5.2 use $\{\kappa_1^{\mathbf{v}},\kappa_2^{\mathbf{v}},\dots,\kappa_5^{\mathbf{v}},\kappa_1^{\mathbf{g}},\kappa_2^{\mathbf{g}},\dots,\kappa_7^{\mathbf{g}}\}$ as a common set of kernels. Although this kernel set performs efficiently in all experiments, it makes poor results for two Libras and Diabetes datasets. On the other hand, using $\{\kappa_1^{\mathbf{v}},\kappa_2^{\mathbf{v}},\dots,\kappa_5^{\mathbf{v}},\kappa_1^{\mathbf{g}},\kappa_2^{\mathbf{g}},\ \dots,\kappa_7^{\mathbf{g}},\kappa_1^{\mathbf{p}},\dots,\kappa_3^{\mathbf{p}}\}$ as second kernel set (by including polynomial kernels to first kernel set), CMKIPCM outperforms other clustering algorithms in both

Table 2
The computational complexity of different clustering algorithms.

Method	Computational complexity
K-Means	$O(NCl\mathcal{T}_{max})$
FCM [26]	$O(NC^2l\mathcal{T}_{max})$
K-harmonic means [32]	$O(NCl\mathcal{T}_{max})$
AHC [34]	$O(N^3)$
MPCKMeans [3]	$O\Big((NCl^3+\lambda_{total}l^3)\mathcal{T}_{max}\Big)$
RCA [33]	$O(\lambda_{total}l^2+l^3+Nl^2)$
Xiang's method [8]	$O\Big(i_\epsilon(\lambda_{total}l^2+l^3)+Nl^2\Big)$
Soleymani's method [5]	$O\Big(i_\epsilon(\lambda_{total}l^2+l^3)+Nl^2+lN^2+lNK^3+dN^2\Big)$
CMKIPCM	$O\Big((N^2CM+NC^2\lambda_{total})\mathcal{T}_{max}\Big)$
CMKIPCM- Active	$O\Big((\mathcal{T}_{max}-\frac{\lambda_{total}}{\lambda})(N^2CM+NC^2\lambda_{total})+\frac{\lambda_{total}}{\lambda}(N^2CM+NC^2\lambda_{total})(NC^2+\lambda N)\Big)$

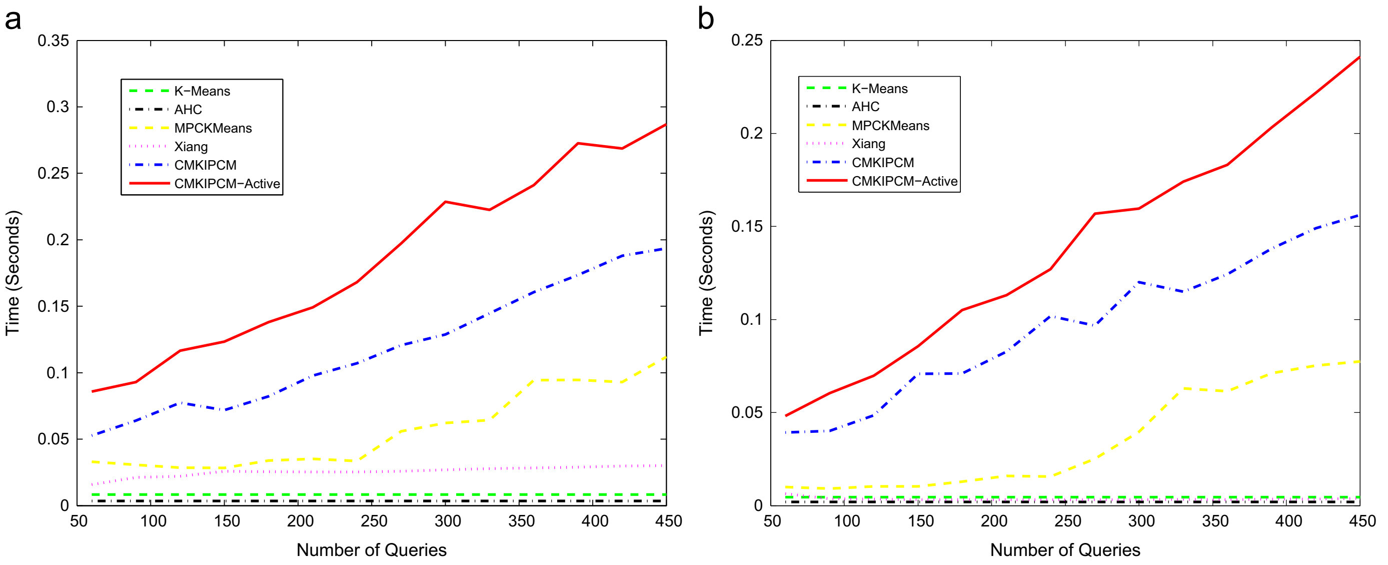

Fig. 5. Computation time (second) of CMKIPCM and compared approaches on a PC with 2.7 GHz CPU and 2 GB RAM, using Matlab 6.5. (a) Sonar. (b) Wine.

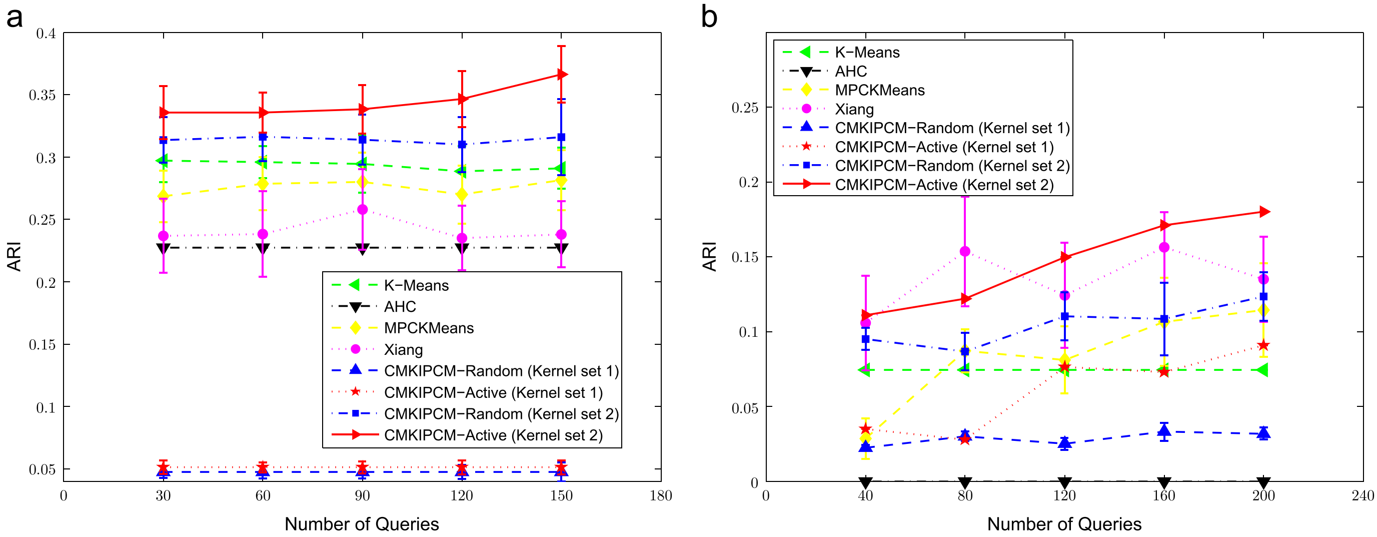

Fig. 6. The effectiveness of CMKIPCM given two different kernel sets $\{\kappa_1^{\mathbf{v}}, \kappa_2^{\mathbf{v}}, \ldots, \kappa_5^{\mathbf{v}}, \kappa_1^{\mathbf{g}}, \kappa_2^{\mathbf{g}}, \ldots, \kappa_7^{\mathbf{g}}\}$ and $\{\kappa_1^{\mathbf{v}}, \kappa_2^{\mathbf{v}}, \ldots, \kappa_5^{\mathbf{v}}, \kappa_1^{\mathbf{g}}, \kappa_2^{\mathbf{g}}, \ldots, \kappa_7^{\mathbf{g}}, \kappa_1^{\mathbf{p}}, \ldots, \kappa_3^{\mathbf{p}}\}$ on Libras and Diabetes datasets. (a) Libras. (b) Diabetes.

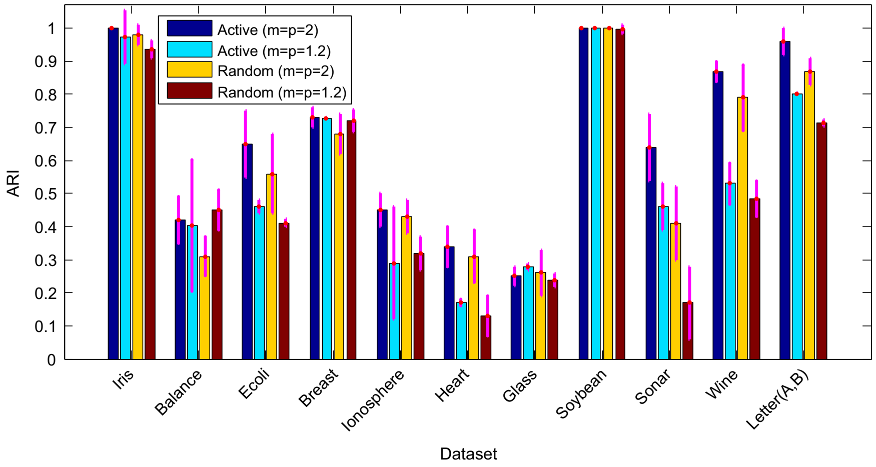

Fig. 7. Performance evaluation of CMKIPCM on 100 constraints selected by two active (proposed embedded active) and random query selection heuristics given two different weighting exponents $m=p=2$ and $m=p=1.2$.

datasets. Fig. 6 shows the effect of two kernel sets 1 and 2 on Libras and Diabetes datasets. As shown in this figure, including Polynomial kernels to the previous set significantly improves the accuracy of CMKIPCM. As a result, for real world applications we have no cues in advance to choose suitable kernel set for the given problem and different sets of kernels make results in different levels of accuracy. The only cues is to use known effective kernels based on the given problem.

Determining the best fuzzy and possibilistic weighting exponents m and p remains as an open issue in fuzzy and possibilistic clusterings. Graves et al. [35] have concluded that the choice of the weighting exponents highly depends on both application and clustering algorithm. Fig. 7 demonstrates the effectiveness of CMKIPCM on 100 randomly and actively selected constraints given two different weighting exponents $m=p=2$ and $m=p=1.2$. We can see that making the clustering more fuzzy and possibilistic provides better performance in some datasets. This is evident for Iris, Ecoli, Ionosphere, Heart, Sonar, Wine, and Letter (A,B) Datasets. For Balance, Breast, and Glass datasets, making the clustering less fuzzy and possibilistic provides better performance. This experiment justifies the dependency of m and p on both application and clustering algorithm.

7. Conclusion

The problem of joint constrained clustering and active constraint selection was addressed in this paper. To this aim, the improved possibilistic c-means was extended to consider pairwise constraints in a multiple kernels learning setting. This extension not only makes CMKIPCM immune to inefficient kernels or irrelevant features but only robust it against noise and outliers. Also, the multiple kernels trick caused CMKIPCM to address the non-linearity of data in clustering. In order to avoid querying inefficient or redundant constraints, an active query selection heuristic was embedded into CMKIPCM based on the measurement of clustering mistake. Comprehensive experiments were conducted to evaluate the efficiency of the proposed method from the reliability, efficiency and sensitivity perspectives. Experiments carried out on real datasets show that the proposed method improves the clustering accuracy by effectively incorporating multiple kernels. A promising future direction is to extend CMKIPCM from point prototypes to hyper-volumes whose size is determined automatically from the data being clustered. These prototypes are shown to be less sensitive to bias in the distribution of the data. Automatically setting the fuzzy and possibilistic weighting exponents and choosing the base kernels can also be considered as other directions to the future works.

Conflict of interest

None declared.

Appendix A

A.1. Optimality proof of Theorem 1

The goal of CMKIPCM is to simultaneously find the combination weights $\mathbf{w} \equiv [\omega_k]_{M\times 1}$, the possibilistic memberships $T \equiv [t_{ci}]_{C\times N}$, the fuzzy memberships $U \equiv [u_{ci}]_{C\times N}$, and the cluster centers $V \equiv [v_c]_{L\times C}$ that minimize the objective function given in Eq. (4). CMKIPCM adopts an alternating optimization approach to minimize J_{CMKIPCM}. With constraints $\sum_{c=1}^{C} u_{ci} = 1$ and $\sum_{k=1}^{M} \omega_k = 1$, the minimum of the objective function is calculated by forming an energy function with Lagrange multipliers γ for constraint $\sum_{c=1}^{C} u_{ci} = 1$ and with a Lagrange multiplier β for constraint $\sum_{k=1}^{M} \omega_k = 1$. For briefness, we use D_{ci} to denote the distance between data x_i and cluster center v_c, i.e., $D_{ci}^2 = (\psi(x_i) - v_c)^T(\psi(x_i) - v_c)$. Thus, the following Lagrange function is obtained.

$$\begin{aligned} J_{\text{CMKIPCM}}(\mathbf{w}, T, U, V, \gamma, \beta) = & \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^m t_{ci}^p D_{ci}^2 + \sum_{c=1}^{C} \eta_c \sum_{i=1}^{N} u_{ci}^m (1-t_{ci})^p \\ & + \alpha\left(\sum_{(i,j)\in\mathcal{M}} \sum_{c=1}^{C} \sum_{\substack{l=1 \\ l\neq c}}^{C} u_{ci}^m u_{lj}^m t_{ci}^p t_{lj}^p \right. \\ & \left. + \sum_{(i,j)\in\mathcal{C}} \sum_{c=1}^{C} u_{ci}^m u_{cj}^m t_{ci}^p t_{cj}^p \right) \\ & + \sum_{i=1}^{N} \gamma_i \left(\sum_{c=1}^{C} u_{ci} - 1\right) + 2\beta\left(\sum_{k=1}^{M} \omega_k - 1\right) \end{aligned} \tag{16}$$

Optimizing the possibilistic memberships $T \equiv [t_{ci}]_{C\times N}$, the fuzzy memberships $U \equiv [u_{ci}]_{C\times N}$, and the weights $\mathbf{w} \equiv [\omega_k]_{M\times 1}$ is described by three following lemmas.

Lemma 1. *When the weights $\mathbf{w} \equiv [\omega_k]_{M\times 1}$, the fuzzy memberships $U \equiv [u_{ci}]_{C\times N}$, and the cluster centers $V \equiv [v_c]_{L\times C}$ are fixed, the optimal values of the possibilistic memberships $T \equiv [t_{ci}]_{C\times N}$ equal to:*

$$t_{ci} = \frac{1}{1 + \left(\dfrac{D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)}{\eta_c}\right)^{1/(p-1)}} \quad \forall c, i. \tag{17}$$

Proof. To find the optimal possibilistic memberships T, the weights $\mathbf{w}$, the fuzzy memberships U, and the cluster centers V are fixed at first. When the weights, the fuzzy memberships, and the cluster centers are fixed, the distances are also constants. Taking derivatives of the Lagrange function given in Eq. (16) with respect to the possibilistic memberships and setting them to zero; for each possibilistic membership t_{ci}, we obtain

$$\begin{aligned} \frac{\partial J(\mathbf{w}, T, U, V, \gamma, \beta)}{\partial t_{ci}} = & \, p u_{ci}^m t_{ci}^{p-1} D_{ci}^2 - \eta_c p u_{ci}^m (1-t_{ci})^{p-1} \\ & + \alpha\left(p u_{ci}^m t_{ci}^{p-1}\left(\underbrace{\sum_{(i,j)\in\mathcal{M}} \sum_{\substack{l=1 \\ l\neq c}}^{C} u_{lj}^m t_{lj}^p}_{S_{ci}^{\mathcal{M}}} + \underbrace{\sum_{(i,j)\in\mathcal{C}} u_{cj}^m t_{cj}^p}_{S_{ci}^{\mathcal{C}}}\right)\right) = 0. \end{aligned} \tag{18}$$

Using Eq. (18), we obtain

$$\eta_c (1-t_{ci})^{p-1} = t_{ci}^{p-1}\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right). \tag{19}$$

So that

$$\left(\frac{1-t_{ci}}{t_{ci}}\right)^{p-1} = \frac{D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)}{\eta_c} \Longrightarrow \frac{1}{t_{ci}} = 1 + \left(\frac{D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)}{\eta_c}\right)^{1/(p-1)}. \tag{20}$$

Thus, the solution for t_{ci} is

$$t_{ci} = \frac{1}{1 + \left(\dfrac{D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)}{\eta_c}\right)^{1/(p-1)}}. \tag{21}$$

This completes the proof of this lemma. □

Lemma 2. *When the weights $\mathbf{w} \equiv [\omega_k]_{M\times 1}$, the possibilistic memberships $T \equiv [t_{ci}]_{C\times N}$, and the cluster centers $V \equiv [v_c]_{L\times C}$ are fixed, the optimal values of the fuzzy memberships $U \equiv [u_{ci}]_{C\times N}$ equal to*

$$u_{ci} = \frac{1}{\sum_{k=1}^{C}\left(\dfrac{t_{ci}^{p-1}\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right)}{t_{ki}^{p-1}\left(D_{ki}^2 + \alpha\left(S_{ki}^{\mathcal{M}} + S_{ki}^{\mathcal{C}}\right)\right)}\right)^{1/(m-1)}} \quad \forall c, i. \tag{22}$$

Proof. To find the optimal fuzzy memberships U, we first fix the weights $\mathbf{w}$, the possibilistic memberships T, and the cluster centers V. Since the weights, the possibilistic memberships and the cluster centers are fixed, the distances are also constants. Taking derivatives of energy function given in Eq. (16) with respect to the fuzzy memberships and setting them to zero; for each fuzzy membership u_{ci}, we obtain

$$\begin{aligned} \frac{\partial J(\mathbf{w}, T, U, V, \gamma, \beta)}{\partial u_{ci}} = & \, m u_{ci}^{m-1} t_{ci}^p D_{ci}^2 + \eta_c m u_{ci}^{m-1} (1-t_{ci})^p \\ & + \alpha\left(m u_{ci}^{m-1} t_{ci}^p\left(\underbrace{\sum_{(i,j)\in\mathcal{M}} \sum_{\substack{l=1 \\ l\neq c}}^{C} u_{lj}^m t_{lj}^p}_{S_{ci}^{\mathcal{M}}} + \underbrace{\sum_{(i,j)\in\mathcal{C}} u_{cj}^m t_{cj}^p}_{S_{ci}^{\mathcal{C}}}\right)\right) \\ & - \gamma_i = 0. \end{aligned} \tag{23}$$

By some algebraic simplifications in Eq. (23), we obtain

$$m u_{ci}^{m-1}\left(t_{ci}^p\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right) + \eta_c (1-t_{ci})^p\right) = \gamma_i \tag{24}$$

$$\Longrightarrow u_{ci} = \left(\frac{\gamma_i}{m\left(t_{ci}^p\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right) + \eta_c (1-t_{ci})^p\right)}\right)^{1/(m-1)}. \tag{25}$$

From Eq. (19), we have

$$\eta_c (1-t_{ci})^p = t_{ci}^{p-1}(1-t_{ci})\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right). \tag{26}$$

By using Eq. (26) in Eq. (25), this equation equals to

$$u_{ci} = \left(\frac{\gamma_i}{m\left(t_{ci}^p\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right) + t_{ci}^{p-1}(1-t_{ci})\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right)\right)}\right)^{1/(m-1)}. \tag{27}$$

By some algebraic simplifications in Eq. (27), the solution for u_{ci} is

$$u_{ci} = \left(\frac{\gamma_i}{m\left(t_{ci}^{p-1}\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right)\right)}\right)^{1/(m-1)}. \tag{28}$$

Because of the constraint $\sum_{k=1}^{C} u_{ki} = 1$, the Lagrange multiplier γ is eliminated as

$$\sum_{k=1}^{C} u_{ki} = \sum_{k=1}^{C} \left(\frac{\gamma_i}{m\left(t_{ki}^{p-1}\left(D_{ki}^2 + \alpha\left(S_{ki}^{\mathcal{M}} + S_{ki}^{\mathcal{C}}\right)\right)\right)} \right)^{1/(m-1)} = 1 \tag{29}$$

$$\Longrightarrow \left(\frac{\gamma_i}{m}\right)^{1/m-1} = \frac{1}{\sum_{k=1}^{C} \left(\frac{1}{t_{ki}^{p-1}\left(D_{ki}^2 + \alpha\left(S_{ki}^{\mathcal{M}} + S_{ki}^{\mathcal{C}}\right)\right)} \right)^{1/(m-1)}}. \tag{30}$$

By using Eq. (30) in Eq. (28), the closed-form solution for the optimal memberships is obtained as

$$u_{ci} = \frac{1}{\sum_{k=1}^{C} \left(\frac{t_{ci}^{p-1}\left(D_{ci}^2 + \alpha\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right)}{t_{ki}^{p-1}\left(D_{ki}^2 + \alpha\left(S_{ki}^{\mathcal{M}} + S_{ki}^{\mathcal{C}}\right)\right)} \right)^{1/(m-1)}}. \tag{31}$$

This completes the proof of this lemma. □

From Lemmas 1 and 2, it can be seen that the optimal possibilistic memberships T and fuzzy memberships U can be obtained when the weights $\mathbf{w}$ and cluster centers V are fixed. The following lemma is used to derive the optimal weights to combine the kernels.

Lemma 3. *When the possibilistic memberships $T \equiv [t_{ci}]_{C \times N}$ and the fuzzy memberships $U \equiv [u_{ci}]_{C \times N}$ are fixed, the optimal values of the weights $\mathbf{w} \equiv [\omega_k]_{M \times 1}$ equal to*

$$\omega_k = \frac{\frac{1}{\mathcal{Y}_k}}{\frac{1}{\mathcal{Y}_1} + \frac{1}{\mathcal{Y}_2} + \cdots + \frac{1}{\mathcal{Y}_M}} \quad \forall k. \tag{32}$$

Proof. To derive the optimal centers and weights to combine the kernels, we assume that both fuzzy and possibilistic memberships are fixed. Taking the derivative of $J_{\text{CMKIPCM}}(\mathbf{w}, T, U, V, \gamma, \beta)$ in Eq. (16) with respect to v_c and setting it to zero leads to the following equation.

$$\frac{\partial J_{\text{CMKIPCM}}(\mathbf{w}, T, U, V, \gamma, \beta)}{\partial v_c} = -2 \sum_{i=1}^{N} u_{ci}^m t_{ci}^p (\psi(x_i) - v_c) = 0 \tag{33}$$

Given T and U, the optimal v_c has the following closed-form solution represented by the combination weights:

$$v_c = \frac{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p \psi(x_i)}{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p} \tag{34}$$

Because these cluster centers are in the feature space which might have an infinite dimensionality, it may be impossible to evaluate these centers directly. Fortunately, for optimizing $J_{\text{CMKIPCM}}(\mathbf{w}, T, U, V, \gamma, \beta)$, it is possible to obtain the possibilistic memberships, the fuzzy memberships and the weights without implicitly evaluating cluster centers. This possibility is shown later in this paper. Thus, we find the optimal weights for fixed fuzzy and possibilistic memberships considering the closed-form optimal solution for the cluster centers. Hence, we try to eliminate cluster centers v_c from the evaluation of the energy function $J_{\text{CMKIPCM}}(\mathbf{w}, T, U, V, \gamma, \beta)$. As mentioned above, the distance between data x_i and cluster center v_c in feature space is calculated as

$$\begin{aligned}
D_{ci}^2 &= (\psi(x_i) - v_c)^T(\psi(x_i) - v_c) = \psi(x_i)^T\psi(x_i) - 2\psi(x_i)^T v_c + v_c^T v_c \\
&= \psi(x_i)^T\psi(x_i) - 2\psi(x_i)^T \left(\frac{\sum_{j=1}^{N} u_{cj}^m t_{cj}^p \psi(x_j)}{\sum_{j=1}^{N} u_{cj}^m t_{cj}^p} \right) \\
&\quad + \left(\frac{\sum_{r=1}^{N} u_{cr}^m t_{cr}^p \psi(x_r)}{\sum_{r=1}^{N} u_{cr}^m t_{cr}^p} \right)^T \left(\frac{\sum_{s=1}^{N} u_{cs}^m t_{cs}^p \psi(x_s)}{\sum_{s=1}^{N} u_{cs}^m t_{cs}^p} \right) \\
&= \sum_{k=1}^{M} \omega_k^2 \kappa_k(x_i, x_i) - \frac{2\sum_{j=1}^{N} \sum_{k=1}^{M} \omega_k^2 u_{cj}^m t_{cj}^p \kappa_k(x_i, x_j)}{\sum_{j=1}^{N} u_{cj}^m t_{cj}^p} \\
&\quad + \frac{\sum_{r=1}^{N} \sum_{s=1}^{N} \sum_{k=1}^{M} \omega_k^2 u_{cr}^m t_{cr}^p u_{cs}^m t_{cs}^p \kappa_k(x_r, x_s)}{\left(\sum_{r=1}^{N} u_{cr}^m t_{cr}^p\right)\left(\sum_{s=1}^{N} u_{cs}^m t_{cs}^p\right)} \\
&= \sum_{k=1}^{M} \omega_k^2 \underbrace{\left(\kappa_k(x_i, x_i) - \frac{2\sum_{j=1}^{N} u_{cj}^m t_{cj}^p \kappa_k(x_i, x_j)}{\sum_{j=1}^{N} u_{cj}^m t_{cj}^p} + \frac{\sum_{r=1}^{N} \sum_{s=1}^{N} u_{cr}^m t_{cr}^p u_{cs}^m t_{cs}^p \kappa_k(x_r, x_s)}{\left(\sum_{r=1}^{N} u_{cr}^m t_{cr}^p\right)\left(\sum_{s=1}^{N} u_{cs}^m t_{cs}^p\right)} \right)}_{\mathcal{Q}_{ci}^k} \\
&= \sum_{k=1}^{M} \omega_k^2 \mathcal{Q}_{ci}^k.
\end{aligned} \tag{35}$$

Eq. (35) eliminates the cluster centers from the evaluation of D_{ci}. Thus, the energy function in Eq. (16) becomes

$$\begin{aligned}
J_{\text{CMKIPCM}}(\mathbf{w}, T, U, V, \gamma, \beta) = {} & \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci}^m t_{ci}^p \sum_{k=1}^{M} \omega_k^2 \mathcal{Q}_{ci}^k + \sum_{c=1}^{C} \eta_c \sum_{i=1}^{N} u_{ci}^m (1 - t_{ci})^p \\
& + \alpha \left(\sum_{(i,j) \in \mathcal{M}} \sum_{c=1}^{C} \sum_{\substack{l=1 \\ l \neq c}}^{C} u_{cc}^m u_{lj}^m t_{ci}^p t_{lj}^p \right. \\
& \left. + \sum_{(i,j) \in \mathcal{C}} \sum_{c=1}^{C} u_{ci}^m u_{cj}^m t_{ci}^p t_{cj}^p \right) + \sum_{i=1}^{N} \gamma_i \left(\sum_{c=1}^{C} u_{ci} - 1 \right) \\
& + 2\beta \left(\sum_{k=1}^{M} \omega_k - 1 \right).
\end{aligned} \tag{36}$$

When the possibilistic and the fuzzy memberships are fixed, by taking the partial derivatives with respect to ω_k and setting them to zero, we have

$$\frac{\partial J(\mathbf{w}, T, U, V, \gamma, \beta)}{\partial \omega_k} = 2 \left(\underbrace{\sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci}^m t_{ci}^p \mathcal{Q}_{ci}^k}_{\mathcal{Y}_k} \right) \omega_k - 2\beta = 0 \Longrightarrow \omega_k = \frac{\beta}{\mathcal{Y}_k}. \tag{37}$$

Since $\sum_{k=1}^{M} \omega_k = 1$, we obtain

$$\sum_{k=1}^{M} \omega_k = \beta \left(\frac{1}{\mathcal{Y}_1} + \frac{1}{\mathcal{Y}_2} + \cdots + \frac{1}{\mathcal{Y}_M} \right) = 1 \Longrightarrow \beta = \frac{1}{\frac{1}{\mathcal{Y}_1} + \frac{1}{\mathcal{Y}_2} + \cdots + \frac{1}{\mathcal{Y}_M}} \tag{38}$$

By substituting Eq. (38) into Eq. (37), we can find the optimum weight as the harmonic mean given below.

$$\omega_k = \frac{\frac{1}{\mathcal{Y}_k}}{\frac{1}{\mathcal{Y}_1} + \frac{1}{\mathcal{Y}_2} + \cdots + \frac{1}{\mathcal{Y}_M}}. \tag{39}$$

This completes the proof of this lemma. □

Using Lemmas 1–3, the convergence of CMKIPCM is concluded from three following lemmas.

Lemma 4. *Let $\mathcal{J}(T) = J_{\text{CMKIPCM}}$, where $T \equiv [t_{ci}]_{C \times N}$, $U \equiv [u_{ci}]_{C \times N}$ are fixed and satisfies the constraints conditions $\sum_{c=1}^{C} u_{ci} = 1$ (for $i = 1, 2, \ldots, N$), $\mathbf{w} \equiv [\omega_k]_{M \times 1}$ are fixed, for all $1 \le c \le C$ and $1 \le i \le N$ we have $D_{ci}^2 > 0$, $m > 1$, and $p > 1$, then T is a local optimum of $\mathcal{J}(T)$, if and only if t_{ci} (for $c = 1, 2, \ldots, C$ and $i = 1, 2, \ldots, N$) are calculated by Eq. (6).*

Proof. The necessity has been proven by Lemma 1. To prove its sufficiency, the Hessian matrix $H(\mathcal{J}(T))$ of $\mathcal{J}(T)$ is obtained using the Lagrange function given in Eq. (16) as the following.

$$\begin{aligned}
h_{fg,ci}(T) &= \frac{\partial}{\partial t_{fg}} \left[\frac{\partial \mathcal{J}(T)}{\partial t_{ci}} \right] \\
&= \begin{cases} p(p-1) u_{ci}^m \left(t_{ci}^{p-2} D_{ci}^2 + \eta_c (1 - t_{ci})^{p-2} + \alpha t_{ci}^{p-2} \left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}} \right) \right) & \text{If } f = c,\ g = i \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{40}$$

According to Eq. (40), $h_{fg,ci}(T)$ is a diagonal matrix. For all $1 \le c \le C$ and $1 \le i \le N$, t_{ci} and u_{ci} are calculated using Eqs. (6) and (7), respectively and $0 < t_{ci} < 1$, $u_{ci} > 0$, $m > 1$, $D_{ci}^2 > 0$, $\eta_c > 0$,

$\alpha > 0$, The above Hessian matrix is a positive definite matrix. So (6) gives the sufficient condition to minimize $\mathcal{J}(T)$. □

Lemma 5. *Let $\mathcal{J}(U) = J_{\text{CMKIPCM}}$, where $U \equiv [u_{ci}]_{C\times N}$ satisfies the constraints conditions $\sum_{c=1}^{C} u_{ci} = 1$ (for $i = 1, 2, \ldots, N$), $T \equiv [t_{ci}]_{C\times N}$ are fixed, $\mathbf{w} \equiv [\omega_k]_{M\times 1}$ are fixed, for all $1 \le c \le C$ and $1 \le i \le N$ we have $D_{ci}^2 > 0$, $m > 1$, and $p > 1$, then U is a local optimum of $\mathcal{J}(U)$, if and only if u_{ci} (for $c = 1, 2, \ldots, C$ and $i = 1, 2, \ldots, N$) are calculated by Eq. (7).*

Proof. The necessity has been proven by Lemma 2. The sufficiency proof is the same as Lemma 4, the Hessian matrix $H(\mathcal{J}(U))$ of $\mathcal{J}(U)$ is obtained using the Lagrange function given in Eq. (16) as the following.

$$h_{fg,ci}(U) = \frac{\partial}{\partial u_{fg}}\left[\frac{\partial \mathcal{J}(U)}{\partial u_{ci}}\right] = \begin{cases} m(m-1)u_{ci}^{m-2}\left(t_{ci}^{p} D_{ci}^{2} + \eta_c (1 - t_{ci})^{p} + \alpha t_{ci}^{p}\left(S_{ci}^{\mathcal{M}} + S_{ci}^{\mathcal{C}}\right)\right) & \text{If } f = c, g = i \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

According to Eq. (41), $h_{fg,ci}(U)$ is a diagonal matrix. For all $1 \le c \le C$ and $1 \le i \le N$, t_{ci} and u_{ci} are separately calculated by Eqs. (6) and (7), $0 < t_{ci} < 1$, $u_{ci} > 0$, $m > 1$, $D_{ci}^2 > 0$, $\eta_c > 0$, $\alpha > 0$, The above Hessian matrix is a positive definite matrix. So (7) gives the sufficient condition to minimize $\mathcal{J}(U)$. □

Lemma 6. *Let $\mathcal{J}(\mathbf{w}) = J_{\text{CMKIPCM}}$, where $\mathbf{w} \equiv [\omega_k]_{M\times 1}$ satisfies the condition $\sum_{k=1}^{M} \omega_k = 1$, $U \equiv [u_{ci}]_{C\times N}$ are fixed and satisfies the constraints conditions $\sum_{c=1}^{C} u_{ci} = 1$ (for $i = 1, 2, \ldots, N$), $T \equiv [t_{ci}]_{C\times N}$ are fixed, for all $1 \le c \le C$ and $1 \le i \le N$ we have $D_{ci}^2 > 0$, $m > 1$, and $p > 1$, then $\mathbf{w}$ is a local optimum of $\mathcal{J}(\mathbf{w})$, if and only if ω_k (for $k = 1, 2, \ldots, M$) are calculated by Eq. (8).*

Proof. The necessity has been proven by Lemma 3. To prove its sufficiency, the Hessian matrix $H(\mathcal{J}(\mathbf{w}))$ of $\mathcal{J}(\mathbf{w})$ is obtained using the Lagrange function given in Eq. (36), which eliminates the cluster centers from the evaluation of D_{ci} as the following.

$$h_{f,k}(\mathbf{w}) = \frac{\partial}{\partial \omega_f}\left[\frac{\partial \mathcal{J}(\mathbf{w})}{\partial \omega_k}\right] = \begin{cases} 2\mathcal{Y}_k & \text{If } f = k \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

According to Eq. (42), $h_{f,k}(\mathbf{w})$ is a diagonal matrix. For all $1 \le c \le C$ and $1 \le i \le N$, t_{ci} and u_{ci} are calculated by Eqs. (6) and (7), respectively, $0 < t_{ci} < 1$, $u_{ci} > 0$, $m > 1$, $p > 1$, The above Hessian matrix is a positive definite matrix. So (6) gives the sufficient condition to minimize $\mathcal{J}(\mathbf{w})$. □

Proof of Theorem 1. The necessary conditions for objective function given in Eq. (4) to attain its minimum was proven in Lemmas 1–3. According to Lemmas 4–6, $J_{\text{CMKIPCM}}(U^{\mathcal{T}+1}, T^{\mathcal{T}+1}, \mathbf{w}^{\mathcal{T}+1}) \le J_{\text{CMKIPCM}}(U^{\mathcal{T}}, T^{\mathcal{T}}, \mathbf{w}^{\mathcal{T}})$ can be proved, therefore, CMKIPCM will converge. □

References

[1] I.A. Maraziotis, A semi-supervised fuzzy clustering algorithm applied to gene expression data, Pattern Recognit. 45 (1) (2012) 637–648.

[2] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, Constrained k-means clustering with background knowledge, in: Proceedings of the 18th International Conference on Machine Learning, ICML '01, 2001, pp. 577–584.

[3] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04, 2004, pp. 11–18.

[4] J. Ye, Z. Zhao, H. Liu, Adaptive distance metric learning for clustering, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 2007, pp. 1–7.

[5] M. Soleymani Baghshah, S. Bagheri Shouraki, Non-linear metric learning using pairwise similarity and dissimilarity constraints and the geometrical structure of data, Pattern Recognit. 43 (2010) 2982–2992.

[6] P. Jain, B. Kulis, J.V. Davis, I.S. Dhillon, Metric and kernel learning using a linear transformation, J. Mach. Learn. Res. 13 (2012) 519–547.

[7] T. Hertz, A. Bar-hillel, D. Weinshall, Boosting margin based distance functions for clustering, in: Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04, 2004, pp. 393–400.

[8] S. Xiang, F. Nie, C. Zhang, Learning a mahalanobis distance metric for data clustering and classification, Pattern Recognit. 41 (12) (2008) 3600–3612.

[9] A.A. Abin, H. beigy, Clustering at presence of side information via weighted constraints ratio gap maximization, in: Proceedings of the First International Workshop on Multi-view data, High Dimensionality, and External Knowledge: Striving for a Unified Approach to Clustering, 3Clust '12, 2012, pp. 27–38.

[10] M. Okabe, S. Yamada, Clustering with constrained similarity learning, in: Proceedings of the 2009 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology, 2009, pp. 30–33.

[11] H. Chang, D. yan Yeung, Locally linear metric adaptation for semi-supervised clustering, in: Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04, 2004, pp. 153–160.

[12] I. Davidson, K.L. Wagstaff, S. Basu, Measuring constraint-set utility for partitional clustering algorithms, in: Proceedings of the Tenth European Conference on Principle and Practice of Knowledge Discovery in Databases, PKDD '06, 2006, pp. 115–126.

[13] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in: Proceedings of the Fifth SIAM International Conference on Data Mining, ICDM '04, 2004, pp. 333–344.

[14] D. Klein, S.D. Kamvar, C.D. Manning, From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering, in: Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02, 2002, pp. 307–314.

[15] Q. Xu, M. desJardins, K.L. Wagstaff, Active constrained clustering by examining spectral eigenvectors, in: Proceedings of the Eigth International Conference on Discovery Science, DS '05, 2005, pp. 294–307.

[16] N. Grira, M. Crucianu, N. Boujemaa, Active semi-supervised fuzzy clustering, Pattern Recognit. 41 (5) (2008) 1834–1844.

[17] P.K. Mallapragada, R. Jin, A.K. Jain, Active query selection for semi-supervised clustering, in: Proceedings of the Nineteenth International Conference on Pattern Recognition, ICPR '08, 2008, pp. 1–4.

[18] V.-V. Vu, N. Labroche, B. Bouchon-Meunier, Improving constrained clustering with active query selection, Pattern Recognit. 45 (4) (2012) 1749–1758.

[19] A.A. Abin, H. Beigy, Active selection of clustering constraints: a sequential approach, Pattern Recognit. 47 (3) (2014) 1443–1458.

[20] J.-S. Zhang, Y.-W. Leung, Improved possibilistic c-means clustering algorithms, IEEE Trans. Fuzzy Syst. 12 (2) (2004) 209–217.

[21] F.deA.T. de Carvalho, Y. Lechevallier, F.M. de Melo, Partitioning hard clustering algorithms based on multiple dissimilarity matrices, Pattern Recognit. 45 (1) (2012) 447–464.

[22] Y. Lu, Y. Wan, Pha: a fast potential-based hierarchical agglomerative clustering method, Pattern Recognit. 46 (5) (2013) 1227–1239.

[23] J.Z. Lai, E.Y. Juan, F.J. Lai, Rough clustering using generalized fuzzy clustering algorithm, Pattern Recognit. 46 (9) (2013) 2538–2547.

[24] F. Khani, M.J. Hosseini, A.A. Abin, H. Beigy, An algorithm for discovering clusters of different densities or shapes in noisy data sets, in: Symposium on Applied Computing (SAC '13), 2013, pp. 144–149.

[25] A.H. Kashan, B. Rezaee, S. Karimiyan, An efficient approach for unsupervised fuzzy clustering based on grouping evolution strategies, Pattern Recognit. 46 (5) (2013) 1240–1254.

[26] J. Bezdek, R. Ehrlich, W. Full, FCM: the fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.

[27] R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (2) (1993) 98–110.

[28] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.

[29] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Pattern Recognit. 46 (1) (2013) 243–256.

[30] A.A. Abin, Active constrained clustering using instance-level constraint ranking (Ph.D. thesis), Sharif University of Technology, 2014.

[31] S.K.D. Soumi Ghosh, Comparative analysis of k-means and fuzzy c-means algorithms, International Journal of Advanced Computer Science and Applications(IJACSA) 4(4).

[32] B. Zhang, Generalized *k*-harmonic means - dynamic weighting of data in unsupervised learning, in: SDM, 2001, pp. 1–13.

[33] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning a mahalanobis metric from equivalence constraints, J. Mach. Learn. Res. 6 (2005) 937–965.

[34] S. Johnson, Hierarchical clustering schemes, Psychometrika 32 (3) (1967) 241–254.

[35] D. Grave, W. Pedrycz, Kernel-based fuzzy clustering and fuzzy clustering: a comparative study, Fuzzy Sets Syst. 161 (4) (2010) 522–543.

Ahmad Ali Abin received the B.Sc. degree in computer engineering from the Iran University of Science and Technology, Iran, in 2005. In September 2008, he completed the M.Sc. degree in computer engineering at the Sharif University of Technology, Iran. He is currently working toward the Ph.D. degree at the Department of Computer Engineering, Sharif University of Technology. His research interests include pattern recognition, machine learning, neural computing and image processing.

Hamid Beigy received the B.Sc. and M.Sc. degrees in computer engineering from the Shiraz University, Shiraz, Iran, in 1992 and 1995, respectively, and the Ph.D. degree in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2004. Since 2004, he joined the Department of Computer Engineering, Sharif University of Technology, Tehran. His current research interests include machine learning theory, scaling up learning algorithms, parallel algorithms, and bioinformatics.