Pattern Recognition Letters 84 (2016) 252–258

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Clustering with side information: Further efforts to improve efficiency

Ahmad Ali Abin*

Faculty of Computer Science and Engineering, Shahid Beheshti University, Velenjak, Tehran, P.O. Box 19395/4716, Iran

ARTICLE INFO

Article history:
Received 21 March 2016
Available online 18 October 2016

MSC:
41A05
41A10
65D05
65D17

Keywords:
Constrained clustering
Constraints selection
Fuzzy clustering
Pairwise constraints

ABSTRACT

This paper examines the issues of constrained clustering and active selection of clustering constraints in a unified approach. A fuzzy clustering method specially crafted to deal with non-spherical clusters and explicit pairwise constraints is proposed in this paper as a core clustering method. An active method for constraints selection is embedded into the core clustering method for querying beneficial constraints during clustering. The proposed approach has two major advantages relative to traditional methods. First, it considers the dependency of constraints effectiveness on the clustering algorithm by unifying both clustering and constraints selection in a uniform, principled framework. Second, a constraints selection method is embedded into the core clustering method based on the fact that constraints will be more useful if they are selected according to the current state of clustering. Experiments conducted on synthetic and real-world datasets show the effectiveness of the proposed method.

1. Introduction

Data clustering is undoubtedly the most significant problem in unsupervised learning. It is done by organizing objects into clusters whose members are similar in some way [16]. Data clustering is used in various applications in the real world. Such as business and marketing, data/text/web mining, image processing, social science, and so on [3].

Recently, there has been a growing interest in constrained clustering. In a constrained clustering setting, the focus is on clustering data in the presence of background knowledge in the form of constraints. These constraints are usually given in the form of pairwise must-link (ML)/cannot-link (CL) constraints which specify that two objects must be in the same cluster (must-link) or different clusters (cannot-link). Although constrained clustering has been emerged to improve the efficiency of clustering, it introduced some new issues in data clustering. Among them: utilizing the information of constraints requires new techniques to be introduced; Benefit from the existing robust clustering algorithms requires the methods to be adapted to use constraints [7]; the effectiveness of constraints is highly dependent on the clustering algorithms [9].

This paper proposes a unified approach for constrained clustering and active selection of clustering constraints, which, relative to traditional methods, has two major advantages. First, it considers the dependency of constraints effectiveness on the clustering algorithm by unifying the clustering and constraints selection in a uniform, principled framework. Second, a constraints selection method is embedded into the clustering method based on the fact that constraints will be more effective if they are selected according to the current state of clustering.

1.1. Related work

Clustering algorithms proposed in the literature of constrained clustering can be classified into two constraint-based and distance-based categories [7]. Algorithms in the constraint-based category incorporate pairwise constraints on cluster membership. In this category, the constraints state whether two instances should be grouped into the same cluster or not, and the clustering algorithm is adapted so that the available constraints bias the search for a suitable clustering of data. Some techniques in this category include constrained K-means clustering [25], probabilistic framework for constrained clustering [5], using constraints in agglomerative hierarchical clustering [11], constrained clustering based on integer linear programming [4], constrained fuzzy clustering [2], and constrained spectral clustering [10], to mention a few.

Algorithms in a distance-based category are first trained to learn a proper distance measure satisfying the given constraints and then use that measure for data clustering. Recent techniques in this category include integrating both clustering algorithm and learning the underlying similarity metric in a uniform framework [7], joint clustering and distance metric learning [28], metric learning with topology preserving [22], Kernel approaches for metric learning [17], margin-based clustering using boosting [13], learning

* Tel./Fax: +98 21 29904106.
E-mail addresses: a_abin@sbu.ac.ir, ali_abin@yahoo.com

http://dx.doi.org/10.1016/j.patrec.2016.10.013
0167-8655/

Mahalanobis distances metric [26], learning distances metric based on similarity information [20], and learning a distance metric that is globally linear but locally non-linear [8], to mention a few.

1.2. Contributions of this work

Most techniques in constrained clustering are fed by constraints that are preselected independent of the clustering algorithms. Querying constraints independent of clustering algorithm results in different degrees of usefulness for various clustering algorithms [9]. Constraints will more useful if they are asked during clustering by considering the current state of clustering, which requires to new techniques to be introduced in constrained clustering.

This paper proposes a unified approach for clustering and constraints selection, which, relative to traditional methods, has two major advantages. First, it considers the dependency of constraints effectiveness on the clustering algorithm by unifying both clustering and constraints selection in a uniform, principled framework. Second, a new constraints selection method is embedded into the clustering method based on the fact that constraints will be more useful if they are selected according to the current state of clustering. To this end, a fuzzy clustering method specially crafted to deal with non-spherical clusters and explicit pairwise constraints is proposed in this paper as a core clustering method. The arbitrary shape of cluster is accounted for by including in the objective function a set of weights for a set of predefined kernels. These weights are calculated to minimize the objective function and immune the core clustering method to inefficient kernels. Querying constraints in the proposed method is done by using an embedded method in which queries beneficial constraints during clustering by making assumptions on the stability of center and boundary of clusters.

The core clustering method proposed in this paper is built upon CMKIPCM, the authors previous work on constrained clustering [2], by applying two extensions to improve the objective function. First, the objective function of CMKIPCM is extended to consider the effect of constraints length when penalizes violation of constraints. Second, the objective function of CMKIPCM is enhanced such that different degrees for constraints certainty are considered during clustering. The optimality proof along with the necessary conditions for the extended objective function to attain its minimum is studied in this paper by introducing new theorems.

1.3. Organization of this paper

In Section 2, we review fuzzy clustering. In Section 3, we present our general formulation for constrained clustering and active selection of clustering constraints. Experimental results on real data in comparison with other methods are presented in Section 4. Finally, the paper concludes with conclusions and future works in Section 5.

2. Fuzzy clustering: an overview

There is a large body of work studying hard clustering of data [18,23]. Techniques based on hard clustering provide poor results when clusters overlap [14,21]. Considering the overlaps among clusters, Fuzzy C-Means (FCM) assigns cluster membership to each data point [6]. Given dataset $X = \{x_1, \ldots, x_N\}$, where $x_i \in \mathbb{R}^l$, FCM is built upon 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 cluster centers, d_{ci}^2 is the distance of feature vector x_i to cluster center v_c, i.e. $||x_i - v_c||^2$, N is the total number of feature vectors, C is the number of partitions, $u_{ci} \in [0, 1]$ is the fuzzy membership of x_i in partition c constrained by $\sum_{i=1}^{N} u_{ci} > 0 \; \forall c$ and $\sum_{c=1}^{C} u_{ci} = 1 \; \forall i$, m controls the clustering fuzziness, and $U \equiv [u_{ci}]$ is a $C \times N$ matrix called fuzzy partition matrix. The constraint that the memberships of a data point across all partitions must sum to one makes FCM sensitive to outliers or noise. Possibilistic C-Means (PCM) was proposed to improve FCM by using a possibilistic type of membership function [19] which is formulated as follows:

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

where $t_{ci} \in [0, 1]$ is the possibilistic membership of x_i in cluster c constrained by $\sum_{i=1}^{N} t_{ci} > 0 \; \forall c$, $T \equiv [t_{ci}]$ is a $C \times N$ matrix called possibilistic partition matrix, p is a weighting exponent for the possibilistic membership, and η_c are suitable positive numbers. PCM is more robust in the presence of noise, in finding valid clusters, and in giving a robust estimate of the centers [19]. However, PCM suffers from high dependency of its performance on good initialization and undesirable tendency to produce coincident clusters.

Improved possibilistic C-Means (IPCM) has been proposed in [29] by integrating FCM into PCM with strong robustness and fast convergence rate. IPCM can determine proper clusters via the fuzzy approach while it can achieve robustness via the possibilistic approach. The objective function for IPCM is as follows:

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

3. The proposed method for constrained clustering

Given a set of data points X, the focus of constrained clustering is to come up the problem of partitioning X into C clusters in the presence of λ_{total} pairwise constraints. In this paper, a fuzzy clustering method is built upon Improved Possibilistic C-Means (as an efficient approach to fuzzy clustering that deals efficiently with partially overlapping clusters and noisy datasets) to deal with non-spherical clusters and explicit pairwise constraints. The arbitrary shape of the cluster is accounted for by including a set of weights for a set of predefined kernels in the objective function. These weights for the kernels are calculated to immune the clustering method to inefficient kernels. To avoid querying inefficient or redundant constraints, an active query selection method is embedded into the proposed method based on the stability of clusters.

3.1. Formulation

Minimizing the following objective function is proposed to extend IPCM by two extra terms for considering existing pairwise constraints into the clustering. Dealing with non-spherical clusters is addressed in this function by applying multiple kernel setting. Because it is not easy to find the right combination of the similarity kernels, kernel weights are adjusted automatically to immune the function to ineffective kernels and make the choice of kernels less crucial.

$$J_{FE}(\mathbf{w}, T, U, V) = \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci}^{m} t_{ci}^{p} D_{ci}^{2} + \sum_{c=1}^{C} \eta_c \sum_{i=1}^{N} u_{ci}^{m} (1 - t_{ci})^{p} + \ldots$$
$$\alpha \left[\sum_{(i,j) \in \mathcal{M}} P_{ij} D_{ij}^{2} \sum_{c=1}^{C} \sum_{\substack{l=1 \\ l \neq c}}^{C} u_{ci}^{m} u_{lj}^{m} t_{ci}^{p} t_{lj}^{p} + \sum_{(i,j) \in \mathcal{C}} P_{ij} D_{ij}^{2} \sum_{c=1}^{C} u_{ci}^{m} u_{cj}^{m} t_{ci}^{p} t_{cj}^{p} \right] \tag{4}$$

$\mathcal{M}$ and $\mathcal{C}$ denotes the sets of must-link and cannot-link constraints, respectively. $U \equiv [u_{ci}]_{C \times N}$ and $T \equiv [t_{ci}]_{C \times N}$ are the fuzzy and

possibilistic membership matrices whose elements are the fuzzy and possibilistic memberships u_{ci} and t_{ci}, respectively. $v_c \in \mathbb{R}^L$ is the center of cth 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. $D_{ci}^2 = (\psi(x_i) - v_c)^T(\psi(x_i) - v_c)$ denotes the distance between data point x_i and cluster center v_c in the feature space, where $\psi(x) = \omega_1\psi_1(x) + \omega_2\psi_2(x) + \cdots + \omega_M\psi_M(x)$ is a non-negative combination of M base kernels mapping in Ψ constrained by $\sum_{k=1}^{M}\omega_k = 1$. $\psi(x)$ maps each data point to an implicit feature space by using kernel weights vector $\mathbf{w} = (\omega_1, \omega_2, \ldots, \omega_M)^T$. $D_{ij}^2 = (\psi(x_i) - \psi(x_j))^T(\psi(x_i) - \psi(x_j))$ denotes the distance between x_i and x_j in the feature space. η_c is a scale parameter and P_{ij} is degree of certainty for constraint (i, j).

Two first terms in Eq. (4) empower IPCM with multiple kernel setting to support compactness of clusters in the feature space. Setting up the objective function (4) with multiple kernels is given in Appendix 2 (submitted as supplementary material). The first term tries to minimize the sum of squared distances to the center of clusters in the feature space, which is weighted by two fuzzy and possibilistic memberships and the second term forces the possibilistic memberships to be as large as possible to stay away from a trivial solution. The last term defines two costs for violating pairwise constraints, which is weighted by degree of supervision α. The first one measures the cost of violating each must-link constraint $(i, j) \in \mathcal{M}$ by penalizing the presence of two data points x_i and x_j in different clusters by their corresponding membership values multiplied by D_{ij}^2 and P_{ij} (degree of constraint certainty provided by user). On the other hand, the second term measures the cost of violating each cannot-link constraint $(i, j) \in \mathcal{C}$ by penalizing the presence of two data points x_i and x_j in same cluster by their corresponding membership values multiplied by D_{ij}^2 and P_{ij}.

3.2. Optimization

The following theorem studies the necessary conditions for the objective function (4) to attain its minimum. The proof of theorem will be given later in Appendix 1 (submitted as supplementary material).

Theorem 1. *J_{FE} attains its local minima If $U \equiv [u_{ci}]_{C\times N}$, $T \equiv [t_{ci}]_{C\times N}$, and $\mathbf{w} \equiv [\omega_k]_{M\times 1}$ are calculated as follows:*

$$t_{ci} = \frac{1}{1 + \left(\frac{D_{ci}^2 + \alpha\left(\Gamma_{ci}^{\mathcal{M}} + \Gamma_{ci}^{\mathcal{C}}\right)}{\eta_c}\right)^{\frac{1}{p-1}}} \tag{5}$$

$$u_{ci} = \frac{1}{\sum_{k=1}^{C}\left(\frac{t_{ci}^{p-1}\left(D_{ci}^2 + \alpha\left(\Gamma_{ci}^{\mathcal{M}} + \Gamma_{ci}^{\mathcal{C}}\right)\right)}{t_{ki}^{p-1}\left(D_{ki}^2 + \alpha\left(\Gamma_{ki}^{\mathcal{M}} + \Gamma_{ki}^{\mathcal{C}}\right)\right)}\right)^{\frac{1}{m-1}}} \tag{6}$$

$$\omega_k = \frac{\frac{1}{\mathcal{F}_k}}{\frac{1}{\mathcal{F}_1} + \frac{1}{\mathcal{F}_2} + \cdots + \frac{1}{\mathcal{F}_M}} \tag{7}$$

where D_{ci}^2 denotes the distance between data point x_i and cluster center v_c in the feature space. Because the cluster centers v_c are in the feature space which might have an infinite dimensionality, it may be impossible to calculate D_{ci}^2 directly. Fortunately, D_{ci}^2 can be calculated by using $D_{ci}^2 = \sum_{k=1}^{M}\omega_k^2\mathcal{Q}_{ci}^k$, which eliminates the center of clusters from the calculation. $\mathcal{Q}_{ci}^k$ is calculated by using Eq. (8). D_{ij}^2 denotes the distance between data points x_i and x_j in the feature space and is calculated by using $D_{ij}^2 = \sum_{k=1}^{M}\omega_k^2\mathcal{R}_{ij}^k$, where $\mathcal{R}_{ij}^k = \kappa_k(x_i, x_i) - 2\kappa_k(x_i, x_j) + \kappa_k(x_j, x_j)$. The weights for M base kernels $\{\kappa\}_{k=1}^{M}$ are given by $\omega_1, \omega_2, \ldots, \omega_M$. $\Gamma_{ci}^{\mathcal{M}}$, $\Gamma_{ci}^{\mathcal{C}}$, and $\mathcal{F}_k$ are used as abbreviations to shorten long expressions in Eqs. (5)–(7).

$$\mathcal{Q}_{ci}^k = \kappa_k(x_i, x_i) - \frac{2\sum_{j=1}^{N}u_{cj}^m t_{cj}^p\kappa_k(x_i, x_j)}{\sum_{j=1}^{N}u_{cj}^m t_{cj}^p} + \frac{\sum_{r=1}^{N}\sum_{s=1}^{N}u_{cr}^m t_{cr}^p u_{cs}^m t_{cs}^p\kappa_k(x_r, x_s)}{\left(\sum_{r=1}^{N}u_{cr}^m t_{cr}^p\right)\left(\sum_{s=1}^{N}u_{cs}^m t_{cs}^p\right)}, \tag{8}$$

$$\Gamma_{ci}^{\mathcal{M}} = \sum_{(i,j)\in\mathcal{M}} P_{ij}D_{ij}^2\sum_{\substack{l=1\\ l\neq c}}^{C}u_{lj}^m t_{lj}^p \qquad \Gamma_{ci}^{\mathcal{C}} = \sum_{(i,j)\in\mathcal{C}}u_{cj}^m t_{cj}^p \tag{9}$$

$$\mathcal{F}_k = \sum_{c=1}^{C}\sum_{i=1}^{N}u_{ci}^m t_{ci}^p\mathcal{Q}_{ci}^k + \ldots$$

$$\alpha\left(\sum_{(i,j)\in\mathcal{M}}P_{ij}\mathcal{R}_{ij}^k\sum_{c=1}^{C}\sum_{\substack{l=1\\ l\neq c}}^{C}u_{cc}^m u_{lj}^m t_{ci}^p t_{lj}^p + \sum_{(i,j)\in\mathcal{C}}P_{ij}\mathcal{R}_{ij}^k\sum_{c=1}^{C}u_{ci}^m u_{cj}^m t_{ci}^p t_{cj}^p\right) \tag{10}$$

3.3. Algorithm description

The proposed method for constrained clustering starts by initializing the possibilistic and fuzzy membership matrices T^0 and U^0 at iteration $\mathcal{T} = 0$. At any subsequent iteration, the proposed method continues clustering by querying constraints according to the current state of clustering and then updating the weight of kernels, the possibilistic memberships, and the fuzzy memberships, respectively. The procedure **SelectConstraints(.)** is invoked at each iteration to return λ informative constraints along with their certainty degrees. (Details on this procedure will be given later in Section 3.4). Both constraints selection and optimization will continue until a specified criterion for convergence is satisfied. Algorithm 1 summarizes the proposed method for constrained clustering.

3.4. Active selection of clustering constraints

A large body of works on constrained clustering has reported the accuracy of clustering averaged on random constraints. Random constraints do not always improve the quality of results [9]. Active selection of constraints is an alternative to get the most beneficial constraints for the least effort. Querying constraints from sparse data regions is the most used assumption in existing techniques [1,12,24,27]. Although, this is a good assumption but cannot be effective when clusters overlap or have complex shapes (see Fig. 1). A useful set of constraints should guide the algorithm in whole clustering stages. For example, in center-based data clustering, constraints should be informative enough to fix the center of clusters at first and the boundary of clusters in the next steps (see Fig. 2). From the above, it can be concluded that it is better if clustering algorithms choose constraints actively to resolve any question about clustering.

Taking the above-mentioned issues into account, an active constraints selection method is proposed that is embedded into the clustering method and empowers it to query constraints according the current state of clustering (see Algorithm 2). The proposed method for active constraints selection is done as the following. At each iteration $\mathcal{T}$ of clustering, the entropy of fuzzy memberships for each data point $x_i \in X$ is calculated as $I(x_i) = -\sum_{c=1}^{C}u_{ci}\log(u_{ci})$. Then, X is divided into three subsets 1) $X_{R_1}^{\mathcal{T}} = \{x_i \mid I(x_i) \le \theta_{R_1}\}$, 2) $X_{R_2}^{\mathcal{T}} = \{x_i \mid \theta_{R_2} - \epsilon_{R_2} < I(x_i) \le \theta_{R_2}\}$, and 3) $X_{R_3}^{\mathcal{T}} = \{x_i \mid I(x_i) > \theta_{R_2}\}$ as shown in Fig. 3a. After that, the proposed method queries constraints at each iteration according to three following steps:

Algorithm 1 Proposed method for clustering with side information. 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}, input threshold vector $\tilde{\theta} = [\theta_{R_1}, \theta_{R_2}, thr_{R_1}, thr_{R_2}, \epsilon_{R_2}, \theta_{noise}]^T$ where θ_{noise} is a user defined threshold for noise and outliers and θ_{R_1}, θ_{R_2}, thr_{R_1}, thr_{R_2} and ϵ_{R_2} are used in active constraints selection function, **output** the fuzzy membership matrix $U \equiv [u_{ci}]_{C \times N}$.

procedure ClusterWithSideInfo($X, C, \{\kappa\}_{k=1}^{M}, \lambda, \lambda_{total}, \tilde{\theta}$)
  Initialize U^0 and T^0. ▷ Memberships initialization
  $\mathcal{T} \leftarrow 0$
  $\mathcal{M}^{(\mathcal{T})} \leftarrow \mathcal{C}^{(\mathcal{T})} \leftarrow \emptyset$
  for c=1,..,C **do**
    $\eta_c \leftarrow K \frac{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p D_{ci}^2}{\sum_{i=1}^{N} u_{ci}^m t_{ci}^p}$
  repeat ▷ Main loop
    $\mathcal{T} \leftarrow \mathcal{T} + 1$
    if $|\{\mathcal{M}^{(\mathcal{T}-1)}, \mathcal{C}^{(\mathcal{T}-1)}\}| < \lambda_{total}$ **then**
      $\{\mathcal{M}^{(\mathcal{T})}, \mathcal{C}^{(\mathcal{T})}\} \leftarrow \{\mathcal{M}^{(\mathcal{T}-1)}, \mathcal{C}^{(\mathcal{T}-1)}\} \cup$ **Select Constraints** $(X, U^{\mathcal{T}}, T^{\mathcal{T}}, C, \tilde{\theta}, \lambda)$
    else
      $\{\mathcal{M}^{(\mathcal{T})}, \mathcal{C}^{(\mathcal{T})}\} \leftarrow \{\mathcal{M}^{(\mathcal{T}-1)}, \mathcal{C}^{(\mathcal{T}-1)}\}$
    $\alpha^{(\mathcal{T})} \leftarrow \frac{N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci}^m t_{ci}^p D_{ci}^2}{(|\mathcal{M}^{(\mathcal{T})}| + |\mathcal{C}^{(\mathcal{T})}|) \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci}^m t_{ci}^p}$
    for k=1,..,M **do** ▷ Update kernel weights
      $\omega_k^{(\mathcal{T})} \leftarrow \frac{\frac{1}{\mathcal{F}_k}}{\frac{1}{\mathcal{F}_1} + \frac{1}{\mathcal{F}_2} + \cdots + \frac{1}{\mathcal{F}_M}}$
    for c=1,..,C **do** ▷ Update t_{ci}
      for i=1,..,N **do**
        $t_{ci}^{(\mathcal{T})} \leftarrow \frac{1}{1 + \left(\frac{D_{ci}^2 + \alpha^{(\mathcal{T})} \left(\Gamma_{ci}^{\mathcal{M}} + \Gamma_{ci}^{\mathcal{C}} \right)}{\eta_c} \right)^{\frac{1}{p-1}}}$
    for c=1,..,C **do** ▷ Update u_{ci}
      for i=1,..,N **do**
        $u_{ci}^{(\mathcal{T})} \leftarrow \frac{1}{\sum_{k=1}^{C} \left(\frac{t_{ci}^{p-1} \left(D_{ci}^2 + \alpha^{(\mathcal{T})} \left(\Gamma_{ci}^{\mathcal{M}} + \Gamma_{ci}^{\mathcal{C}} \right) \right)}{t_{ki}^{p-1} \left(D_{ki}^2 + \alpha^{(\mathcal{T})} \left(\Gamma_{ki}^{\mathcal{M}} + \Gamma_{ki}^{\mathcal{C}} \right) \right)} \right)^{\frac{1}{m-1}}}$
  until $\|U^{\mathcal{T}} - U^{\mathcal{T}-1}\| < \epsilon$
  return $U^{\mathcal{T}}$ ▷ Return fuzzy membership matrix $U^{\mathcal{T}}$

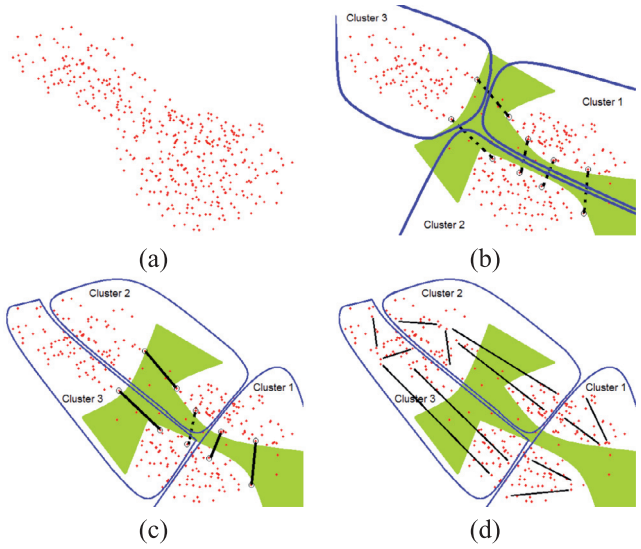

Fig. 1. (a) Three overlapped clusters. There is no clear boundary between clusters to be used for constraints selection (CL constraints in dashed black and ML constraints in solid black color), (b) Three clusters with sparse region among clusters in shaded green and boundary of clusters in solid blue color. Constraints queried from the sparse regions can be informative, (c) When sparse regions do not happen between clusters, constraints queried from these regions are not informative enough to fix the boundary of clusters, and (d) Some useful queries to fix the boundary of clusters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

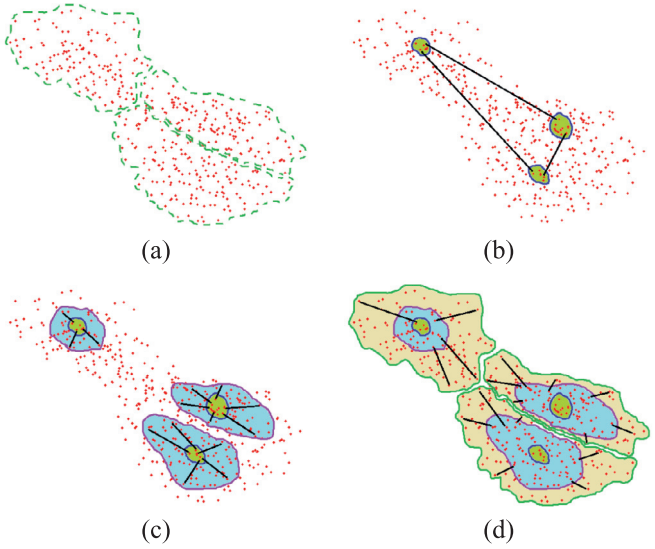

Fig. 2. (a) Three overlapped clusters with true boundary of clusters in dashed green, (b), (c), and (d) Querying constraints to fix the center, the strong boundary and the precise boundary of clusters, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Step 1: Fixing center of clusters by checking the stability of $X_{R_1}^{\mathcal{T}-1}$ (X_{R_1} at iteration $\mathcal{T}-1$) and choosing λ points using a min-max approach if this condition fails. This step is done to stabilize center of clusters. The data point $x_i \in X_{R_1}^{\mathcal{T}}$ with the smallest entropy $I(x_i)$ is selected as the first point and is added to empty set $\mathcal{S}_{R_1}$. Selection of the $\lambda - 1$ remaining points is continued by choosing data point $x_q = \arg\max_i \min_{x_j \in \mathcal{S}_{R_1}} D_{ij}$ (x_q whose smallest distance to points in $\mathcal{S}_{R_1}$ is the largest (see Fig. 3b)).

Step 2: Making the strong boundary of clusters stable by examining the stability condition for strong boundary of clusters if the first condition is satisfied and choosing λ points from $X_{R_2}^{\mathcal{T}}$ using a min-max approach. The data point $x_i \in X_{R_2}^{\mathcal{T}}$ with the largest entropy $I(x_i)$ is selected as the first point and is added to empty set $\mathcal{S}_{R_2}$. Selection of the $\lambda - 1$ remaining points is continued by choosing data point $x_q = \arg\max_i \min_{x_j \in \mathcal{S}_{R_2}} D_{ij}$. After choosing x_q, the type of constraint between x_q and nearest point in $X_{R_1}^{\mathcal{T}}$ is queried from the user (see Fig. 3c).

Step 3: Discovering the precise boundary of clusters by choosing λ points from $X_{R_3}^{\mathcal{T}}$ using a min-max approach if two previous conditions are satisfied. Data point $x_i \in X_{R_3}^{\mathcal{T}}$ with the largest entropy $I(x_i)$ is selected as the first point and is added to empty set $\mathcal{S}_{R_3}$. Selection of the $\lambda - 1$ remaining points is continued by choosing data point $x_q = \arg\max_i \min_{x_j \in \mathcal{S}_{R_3}} D_{ij}$. For each x_q, the type of constraint between x_q and nearest point in $X_{R_2}^{\mathcal{T}}$ is queried from the user (Fig. 3d).

Algorithm 2 summarizes the proposed method for active selection of clustering constraints.

Algorithm 2 Proposed method for 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 $T \equiv [t_{ci}]_{C \times N}$, input threshold vector $\bar{\theta} = [\theta_{R_1}, \theta_{R_2}, thr_{R_1}, thr_{R_2}, \epsilon_{R_2}, \theta_{noise}]^T$ where θ_{noise} is a user defined threshold for noise and outliers, **output** set of must-links constraints $\mathcal{M}$ and cannot-link constraints $\mathcal{C}$.

procedure SELECTCONSTRAINTS($X, U, T, C, \bar{\theta}, \lambda$)
  $\mathcal{M} \leftarrow \mathcal{C} \leftarrow \emptyset$
  ▷ Arbitrary step to eliminate noise and outliers
  $X \leftarrow \{x_i \in X : \max(t_{ci}) \geq \theta_{noise}, \forall c : 1, \ldots, C\}$
  Partition X into three subsets $X_{R_1}^{\mathcal{T}}$, $X_{R_2}^{\mathcal{T}}$ and $X_{R_3}^{\mathcal{T}}$
  ▷ Step 1 Making the center of clusters stable
  if $\frac{|X_{R_1}^{\mathcal{T}} \cap X_{R_1}^{\mathcal{T}-1}|}{|X_{R_1}^{\mathcal{T}-1}|} < thr_{R_1}$ **OR** $\mathcal{T} = 0$ **then**
    $\mathcal{S}_{R_1} \leftarrow \emptyset$
    $q \leftarrow 1$
    $x_q \leftarrow \arg\min_{x_i \in X_{R_1}^{\mathcal{T}}} I(x_i)$
    $\mathcal{S}_{R_1} \leftarrow \mathcal{S}_{R_1} \cup x_q$
    for q= $2, \ldots, \lambda$ **do**
      $x_q \leftarrow \arg\max_i \min_{x_j \in \mathcal{S}_{R_1}} D_{ij}$
      Query for $(x_q, x_j \in \mathcal{S}_{R_1}) \in \mathcal{M}$ or $\mathcal{C}$?
      $\mathcal{S}_{R_1} \leftarrow \mathcal{S}_{R_1} \cup x_q$
      ▷ Step 2 Making the strong boundary stable
  else if $\frac{|X_{R_2}^{\mathcal{T}} \cap X_{R_2}^{\mathcal{T}-1}|}{|X_{R_2}^{\mathcal{T}-1}|} < thr_{R_2}$ **then**
    for c= $1, \ldots, C$ **do**
      $X_{R_2^c}^{\mathcal{T}} \leftarrow \{x_i \in X_{R_2}^{\mathcal{T}} : u_{ci} \geq u_{c'i}, \forall\ 1 \leq c' \leq C, c \neq c'\}$
      $\mathcal{S}_{R_2} \leftarrow \emptyset$
      $q \leftarrow 1$
      $x_q \leftarrow \arg\min_{x_i \in X_{R_2^c}^{\mathcal{T}}} I(x_i)$
      $\mathcal{S}_{R_2} \leftarrow \mathcal{S}_{R_2} \cup x_q$
      for q= $2, \ldots, \frac{\lambda}{C}$ **do**
        Query for $(x_q, x_j \in \mathcal{S}_{R_1}) \in \mathcal{M}$ or $\mathcal{C}$?
        $x_q \leftarrow \arg\max_i \min_{x_j \in \mathcal{S}_{R_2}} D_{ij}$
        $\mathcal{S}_{R_2} \leftarrow \mathcal{S}_{R_2} \cup x_q$
      ▷ Step 3 Making the precise boundary stable
  else
    $\mathcal{S}_{R_3} \leftarrow \emptyset$
    $q \leftarrow 1$
    $x_q \leftarrow \arg\max_{x_i \in X_{R_3}^{\mathcal{T}}} I(x_i)$
    $\mathcal{S}_{R_3} \leftarrow \mathcal{S}_{R_3} \cup x_q$
    for q= $2, \ldots, \lambda$ **do**
      Query for $(x_q, x_j \in \mathcal{S}_{R_2}) \in \mathcal{M}$ or $\mathcal{C}$?
      $x_q \leftarrow \arg\max_i \min_{x_j \in \mathcal{S}_{R_3}} D_{ij}$
      $\mathcal{S}_{R_3} \leftarrow \mathcal{S}_{R_3} \cup x_q$
  return $\{\mathcal{M}, \mathcal{C}\}$ ▷ Return λ selected constraints

4. Experiments

Experiments are conducted to evaluate the proposed method in comparison with some well-known algorithms in constrained clustering. All experiments are carried out in Matlab 7.0 environment on a Pentium 2.4 GHz processor and 4 GB RAM. Details on experimental setup, compared methods, datasets, and evaluation measure are given in the following.

Experimental setup: Parameters m, p, α, and η_c in Eq. (4) are set to their suggested values [2,29]. Degree of constraint certainty for each constraint is set to 1 in the proposed method. Setting base kernels is done similar to CMKIPCM by using $\left\{\kappa_1^{\mathbf{v}}, \kappa_2^{\mathbf{v}}, \ldots, \kappa_5^{\mathbf{v}}, \kappa_1^{\mathbf{g}}, \kappa_2^{\mathbf{g}}, \ldots, \kappa_7^{\mathbf{g}}\right\}$ as the base kernels where $\kappa_k^{\mathbf{v}} = \mathbf{v}_k^T \mathbf{v}_k$ ($\mathbf{v}_k$ is the k^{th} eigenvectors of linear kernel $X^T X$ and $\kappa_k^{\mathbf{g}}(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^T (x_i - x_j)}{2^k \sigma_X}\right)$ (σ_X is the standard deviation of all $\binom{N}{2}$ pairwise distances among points in dataset X). The threshold vector $\bar{\theta} = [\theta_{R_1}, \theta_{R_2}, thr_{R_1}, thr_{R_2}, \epsilon_{R_2}, \theta_{noise}]^T$ is set to $[0.1log(C), 0.6\log(C), 0.5, 0.5, 0.1log(C), 0.1]^T$. ϵ is set to 10^{-3} and λ is set to $3C$ throughout the experiments.

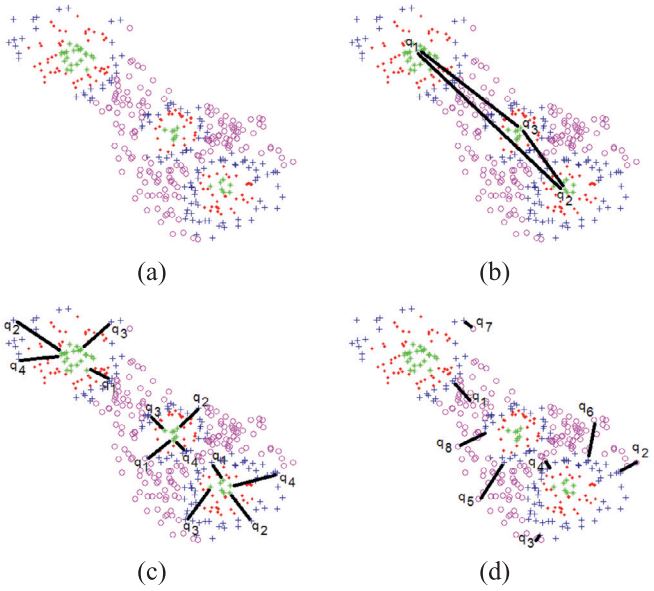

Fig. 3. (a) $X_{R_1}^{\mathcal{T}}$ (green stars), $X_{R_2}^{\mathcal{T}}$ (blue pluses), and $X_{R_3}^{\mathcal{T}}$ (cyan circles) in an arbitrary iteration $\mathcal{T}$ of clustering with $\theta_{R_1} = 0.15$, $\theta_{R_2} = 0.7$ and $\epsilon_{R_2} = 0.2$. (b) Querying constraints from $X_{R_1}^{\mathcal{T}}$, (c) Querying constraints between points in $X_{R_2}^{\mathcal{T}}$ and their closest point in $X_{R_1}^{\mathcal{T}}$, and (d) Querying constraints between points in $X_{R_3}^{\mathcal{T}}$ and their closest point in $X_{R_2}^{\mathcal{T}}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Comparison of the results using a paired t-test.

Dataset	(N, l, C)	Paired t-test	λ_{total}
Iris	(150,4,3)	A < M < K < X ∼ C < P	45
Balance	(625,4,3)	K < X < A < M < C < P	120
Ecoli	(336,7,8)	K ∼ X ∼ M < A < P ∼ C	120
Breast	(569,30,2)	A < K < M < X ∼ C < P	120
Ionosphere	(354,31,2)	A ∼ M < K ∼ X < C ∼ P	120
Heart	(270,13,2)	A < K < X ∼ C ∼ M < P	120
Glass	(214,9,6)	A < M < X ∼ K < C ∼ P	120
Soybean	(47,34,4)	K < A < X < M ∼ C ∼ P	50
Sonar	(208,60,2)	A ∼ K < X ∼ M < C < P	120
Wine	(178,13,3)	A < K < C ∼ X < M ∼ P	120
Letter(A,B)	(1555,16,2)	A < K < M < X < C < P	120

K ≡ **K**-Means **A** ≡ **A**HC **M** ≡ **M**PCKMeans
X ≡ **X**iang **C** ≡ **C**MKIPCM **P** ≡ **P**roposed

Compared methods: In all experiments, the quality of results for the proposed method is compared against the classical K-Means and Agglomerative Hierarchical Clustering (AHC) as two entirely unsupervised algorithms in data clustering and MPCK-Means [7] and Xiang's method [26] as two well-known algorithms in constrained clustering. In addition, the proposed method is compared against CMKIPCM [2]. In all experiments, MPCKMeans, Xiang's method, and CMKIPCM are configured by default parameters. For both proposed and CMKIPCM, their performances are averaged over 50 runs at each experiment.

Datasets: Some datasets from UCI machine learning repository are used to conduct experiments. Two first columns in Table 1

describe each by its name, number of objects (N), number of attributes (l), and number of clusters (C).

Evaluation measure: The well-known *Adjusted Rand Index* (ARI) [15] is used to evaluate the accuracy of clustering. To convert the fuzzy memberships $U \equiv [u_{ci}]_{C \times N}$ to hard clustering of data, each data is assigned to the cluster with the highest membership degree.

4.1. Performance analysis

The results are given in Fig. 4. As this figure shows, the proposed method generally outperforms the compared clustering algorithms. The superiority of the proposed method is considerably observable for Balance, Heart, Ionosphere, and Sonar datasets.

As the results show, the proposed method makes a great improvement in comparison with K-Means, whereas AHC, Xiang, and MPCKMeans algorithms drop into accuracy level lower than K-Means in some datasets (e.g. Ionosphere and Glass). When the proposed method is compared against AHC, it should be mentioned that the proposed method provides more dominant results than AHC in almost all experiments. It can be concluded from the results that AHC is not efficient enough to cluster complex datasets specially when high dimensional data exist.

In comparison with MPCKMeans, the proposed method makes better results than MPCKMeans especially in Balance and Sonar datasets. Although MPCKMeans tries to learn an appropriate distance function during clustering, the experiments show that the distance function has no significant improvement when the number of queries exceeds a threshold in some datasets (e.g., when the number of queries exceeds 90 in Balance dataset). On the other hand, the proposed method results in a nearly smooth increase in accuracy, which implies the usefulness of the newly added constraints.

Experiments show that the Xiang's method has kept the accuracy at a level greater than K-Means and AHC. In comparison with the proposed method, the Xiang's method outperforms the proposed method in early stages of clustering some data (e.g. Wine and Heart datasets) but the proposed method surpasses it when the number of queries increases. Like MPCKMeans, it suffers from improving the right metric when the number of constraints exceeds a threshold. This issue is especially noticeable in Heart, Soybean, and Letter (A,B) datasets.

In comparison between the proposed method and CMKIPCM, it can be observed that the proposed method has no significant improvement in early stages of clustering Breast, Heart, Glass, and Sonar datasets. This is because that the proposed method loses some constraints to make the center of clusters stable in early stages of clustering. When the number of queries increases, the proposed method outperforms CMKIPCM in almost all datasets.

A paired t-test with significance level 0.05 was conducted to significantly evaluate the results given in Fig. 4 for specified number of λ_{total} constraints on each dataset (see Table 1). We use the relation $x < y$ to state that the ARI values of the 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, the proposed method outperforms the compared methods with a 95% confidence level.

4.2. Complexity analysis

In this section, the computational complexity of the proposed method is studied against the compared approaches when they are implemented in primary mode (See Table 2). Let $\mathcal{T}_{\max}$ be the number of iterations an iterative algorithm needs to converge and λ_{total} be the total number of constraints.

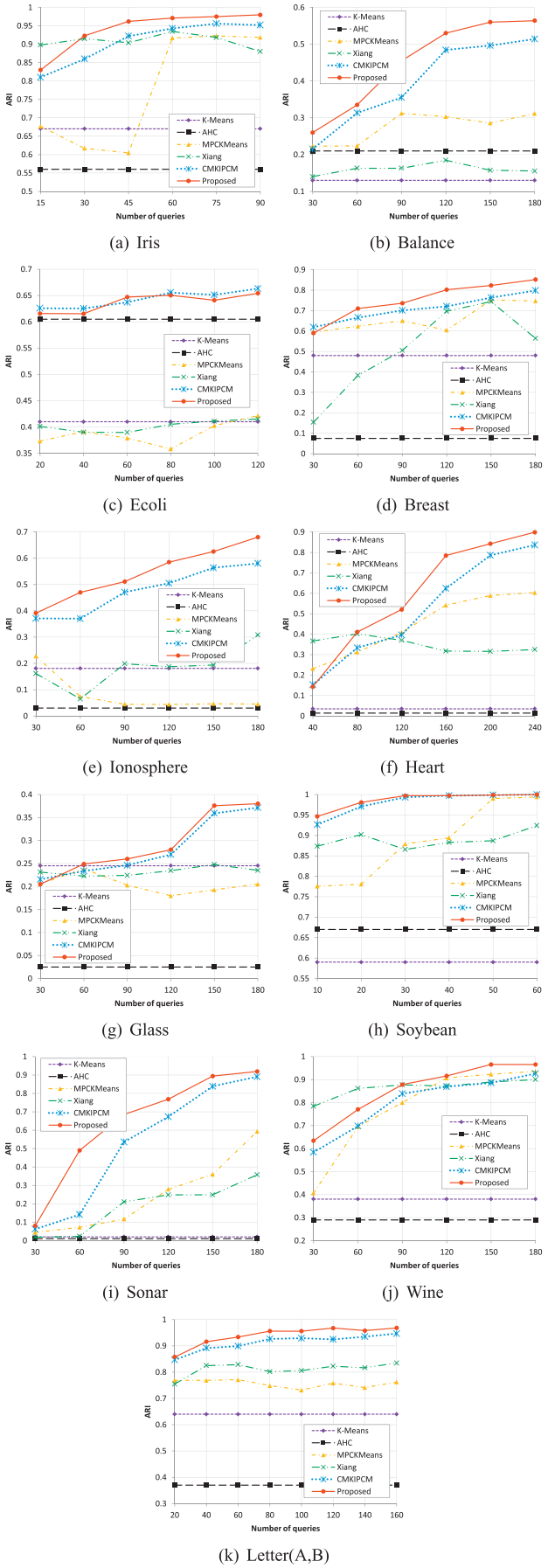

Fig. 4. Comparison of different clustering algorithms on UCI data.

Table 2
Complexity analysis of different clustering algorithms.

Method	Computational complexity
K-Means	$O(NCl\mathcal{T}_{\max})$
AHC	$O(N^3)$
MPCKMeans	$O\big((NCl^3+\lambda_{total}l^3)\mathcal{T}_{\max}\big)$
Xiang's method	$O\big((\lambda_{total}l^2+l^3)\mathcal{T}_{\max}+Nl^2\big)$
CMKIPCM	$O\big((N^2CM+NC^2\lambda_{total})\mathcal{T}_{\max}\big)$
Proposed	$O\big((N^2CM+N^2M\lambda_{total}+NC^2\lambda_{total})\mathcal{T}_{\max}\big)$

The time complexity of K-Means is $O(NCl\mathcal{T}_{\max})$. In the general case, the complexity of agglomerative hierarchical clustering (AHC) is $O(N^3)$. 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 $O\big((NCl^3+\lambda_{total}l^3)\mathcal{T}_{\max}\big)$ [7].

Xiang's method solves an Eigen decomposition problem in an iterative manner. It takes $O(\lambda_{total}l^2)$ to compute the covariance matrix of data points involved in constraints and $O(l^3)$ to solve an Eigen decomposition problem at each iteration. Finally, it takes $O(Nl^2)$ to apply the optimal transformation matrix to the original data points. So, the time complexity of Xiang's method is $O\big((\lambda_{total}l^2+l^3)\mathcal{T}_{\max}+Nl^2\big)$ [26].

Both CMKIPCM and the proposed method take $O(N^2CM)$ 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. CMKIPCM takes $O(NC^2+\lambda_{total}N)$ at each iteration to choose constraints actively, whereas it is $O(N^2M\lambda_{total}+NC)$ for the proposed method. So, the time complexity for CMKIPCM is $O\big((N^2CM+NC^2\lambda_{total})\mathcal{T}_{\max}\big)$ and for the proposed method is $O\big((N^2CM+N^2M\lambda_{total}+NC^2\lambda_{total})\mathcal{T}_{\max}\big)$.

5. Conclusion

A unified framework for constrained clustering and active selection of constraints was addressed in this paper. The proposed method adopts an alternating approach for constrained clustering, which empowers the idea of fuzzy clustering by learning based on multiple kernel and the information of constraints. The constraints are selected based on the fact that it will be better if the clustering algorithm queries constraints actively during clustering. Considering this assumption, an active constraint selection method has been embedded into the proposed method to query constraints according to the current state of clustering. Automatically setting parameters based on the structure of input records can be considered as a promising direction for the future work.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patrec.2016.10.013

References

[1] A.A. Abin, H. Beigy, Active selection of clustering constraints: a sequential approach, Pattern Recognit. 47 (3) (2014) 1443–1458.
[2] A.A. Abin, H. Beigy, Active constrained fuzzy clustering: a multiple kernels learning approach, Pattern Recognit. 48 (3) (2015) 953–967.
[3] A. Albarelli, E. Rodolà, A. Torsello, Loosely distinctive features for robust surface alignment, in: Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part V, 2010, pp. 519–532.
[4] B. Babaki, T. Guns, S. Nijssen, Constrained clustering using column generation, in: Integration of AI and OR Techniques in Constraint Programming - 11th International Conference, CPAIOR 2014, Cork, Ireland, May 19–23, 2014. Proceedings, 2014, pp. 438–454.
[5] S. Basu, M. Bilenko, R.J. Mooney, A probabilistic framework for semi-supervised clustering, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22–25, 2004, 2004, pp. 59–68.
[6] J. Bezdek, R. Ehrlich, W. Full, FCM: the fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.
[7] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4–8, 2004, 2004.
[8] H. Chang, D. Yeung, Locally linear metric adaptation for semi-supervised clustering, in: Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4–8, 2004, 2004.
[9] I. Davidson, K. Wagstaff, S. Basu, Measuring constraint-set utility for partitional clustering algorithms, in: Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18–22, 2006, Proceedings, 2006, pp. 115–126.
[10] S. Ding, H. Jia, L. Zhang, F. Jin, Research of semi-supervised spectral clustering algorithm based on pairwise constraints, Neural Comput. Appl. 24 (1) (2014) 211–219.
[11] J.M.M. Duarte, A.L.N. Fred, F.J.F. Duarte, Evidence accumulation clustering using pairwise constraints, in: KDIR 2012 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Barcelona, Spain, 4 - 7 October, 2012, 2012, pp. 293–299.
[12] N. Grira, M. Crucianu, N. Boujemaa, Active semi-supervised fuzzy clustering, Pattern Recognit. 41 (5) (2008) 1834–1844.
[13] T. Hertz, A. Bar-Hillel, D. Weinshall, Boosting margin based distance functions for clustering, in: Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4–8, 2004, 2004, pp. 393–400.
[14] H.C. Huang, Y.Y. Chuang, C.S. Chen, Multiple kernel fuzzy clustering, IEEE. Trans. Fuzzy Syst. 20 (1) (2012) 120–134.
[15] L. Hubert, P. Arabie, Comparing partitions, J.Classif. 2 (1) (1985) 193–218.
[16] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[17] 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.
[18] 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: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18–22, 2013, 2013, pp. 144–149.
[19] R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, IEEE. Trans. Fuzzy Syst. 1 (2) (1993) 98–110.
[20] 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 - Workshops, Milan, Italy, 15–18 September 2009, 2009, pp. 30–33.
[21] C. Qiu, J. Xiao, L. Han, M.N. Iqbal, Enhanced interval type-2 fuzzy c-means algorithm with improved initial center, Pattern Recognit. Lett. 38 (2014) 86–92.
[22] M. Soleymani, S. Bagheri, Non-linear metric learning using pairwise similarity and dissimilarity constraints and the geometrical structure of data, Pattern Recognit. 43 (2010) 2982–2992.
[23] R. Vidal, P. Favaro, Low rank subspace clustering (LRSC), Pattern Recognit. Lett. 43 (2014) 47–61.
[24] V.-V. Vu, N. Labroche, B. Bouchon-Meunier, Improving constrained clustering with active query selection, Pattern Recognit. 45 (4) (2012) 1749–1758.
[25] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, Constrained k-means clustering with background knowledge, in: Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28, - July 1, 2001, 2001, pp. 577–584.
[26] S. Xiang, F. Nie, C. Zhang, Learning a mahalanobis distance metric for data clustering and classification, Pattern Recognit. 41 (12) (2008) 3600–3612.
[27] Q. Xu, M. desJardins, K. Wagstaff, Active constrained clustering by examining spectral eigenvectors, in: Discovery Science, 8th International Conference, DS 2005, Singapore, October 8–11, 2005, Proceedings, 2005, pp. 294–307.
[28] X. Yin, S. Chen, E. Hu, D. Zhang, Semi-supervised clustering with metric learning: an adaptive kernel method, Pattern Recognit. 43 (4) (2010) 1320–1333.
[29] J.-S. Zhang, Y.-W. Leung, Improved possibilistic c-means clustering algorithms, IEEE. Trans. Fuzzy Syst. 12 (2) (2004) 209–217.