Semi-Supervised Clustering with Constraints of Different Types from Multiple Information Sources

Liang Bai, Jiye Liang *Senior Member, IEEE*, Fuyuan Cao

Abstract—Semi-supervised clustering is one of important research topics in cluster analysis, which uses pre-given knowledge as constraints to improve the clustering performance. While clustering a data set, people often get prior constraints from different information sources, which may have different representations and contents, to guide clustering process. However, most of existing semi-supervised clustering algorithms are based on single-source constraints and rarely consider to integrate multi-source constraints to enhance the clustering quality. To solve the problem, we analyze the relations among different types of constraints and propose an uniform representation for them. Based it, we propose a new semi-supervised clustering algorithm to find out a clustering that has good cluster structure and high consensus of all the sources of constraints. In the algorithm, we construct an optimization objective model and its solution method to achieve the aim. This algorithm can integrate multi-source constraints well to reduce the effect of incorrect constraints from single sources and find out a high-quality clustering. By the experimental studies on several benchmark data sets, we illustrate the effectiveness of the proposed algorithm, compared to other semi-supervised clustering algorithms.

Index Terms—Cluster analysis, semi-supervised clustering, multi-source constraints, consensus of constraints.

1 Introduction

Clustering is an important problem in statistical multivariate analysis, data mining and machine learning [1]. The goal of clustering is to group a set of objects into clusters so that the objects in the same cluster are highly similar but remarkably dissimilar with objects in other clusters [2]. To tackle this problem, various types of clustering algorithms have been developed in the literature (e.g., [3] and references therein), including partitional, hierarchical, density-based and grid-based clustering and so on.

Since clustering is no need for class labels of data sets, it is also called "unsupervised learning". However, everything has two sides. Due to the lack of class labels, the clustering results may be different from users' expectation. To solve the problem, a number of *semi-supervised clustering* approaches have been proposed in the literature [4], [5], which are able to make use of prior knowledge, such as pairwise information (Must-Link and Cannot-Link constraints) or label information (Positive-Label and Negative-Label constraints), to guide the search process. Wagstaff et al. [6] early put forward the concept of constrained clustering which incorporates pairwise information into a traditional clustering algorithm. They proposed a k-means [7] algorithm with pairwise constraints, called COP-k-means [8], which attempts to satisfy all the constraints and assign each object to its nearest cluster. Several improved COP-k-means algorithms have been proposed in [9], [10]. Besides, many classical clustering algorithms have been extended to cluster data sets with pairwise constraints, such as semi-supervised mean-shift clustering [11], semi-supervised maximum margin clustering [12], semi-supervised gaussian mixture model [13], constrained spectral clustering [14], [15], [16], semi-supervised generative model [17], semi-supervised active clustering [18], [19]. For label information, some scholars [20] in early stage used them to find better initialization of cluster centers and employed the standard k-means to finish the clustering task. Liu et al. viewed the Positive-Label information as the constraints and proposed a k-means clustering with label constraints [21]. Wu et al. proposed the nonnegative matrix factorization (NMF) [22] clustering algorithm with the Positive-Label constraints [23]. Zhou et al. proposed a semi-supervised label propagation algorithm which can be seen as a spectral clustering [24], [25] with Positive-Label constraints [26]. Zoidi et al. further extended the algorithm and proposed a label propagation algorithm with Negative-Label constraints [27]. Besides, many scholars [28], [29], [30], [31], [32] explored distance metric learning in semi-supervised clustering which employs the constraints to learn a best distance metric for clustering. Currently, semi-supervised clustering has been widely applied in different areas, such as image processing [33], [34], natural language processing [35], bioinformatics [36] and social networks [37].

Most of existing semi-supervised clustering algorithms have an important limitation: They mainly deal with single-source constraints and rarely consider to integrate multi-source constraints to improve the effective-

• *L. Bai, J. Liang and F. Cao are with school of Computer and Information Technology, Shanxi University, Taiyuan, 030006, Shanxi, China. E-mail: {bailiang, ljy and cfy}@sxu.edu.cn*

ness of the clustering result. The performance of a clustering algorithm with single-source constraints depends on the quality of constraints. It is very easily affected by noisy or incorrect constraints. In real-world applications, we may get multi-source constraints on a data set. For example, there are a data set with three clusters and constraints from four experts (See Fig. 1). These experts have different prior information on clustering the data set. We can see that multi-source constraints can provide sufficient information about the cluster structure of the data set and reduce the effect of incorrect information, compared to single-source constraints. However, the constraints from different sources may have different types, which can be seen in Fig. 1. Currently, the representatives of constraint types include pairwise constraints (Must-Link and Cannot-Link) and label constraints (Positive-Label and Negative-Label) which are widely used in semi-supervised clustering. Besides, the constraints from different sources may conflict with each other (e.g., the relation between objects B and D in Fig. 1). Therefore, the multi-source constraints bring the opportunity as well as a great challenge to a semi-supervised clustering task. In order to integrate multi-source constraints well to guide the clustering process, we will discuss two important issues: (1) *how do we construct an uniform representation to save different types of constraints?* and (2) *how do we integrate multi-source constraints to guide clustering process?* The main contributions of this paper are highlighted as follows.

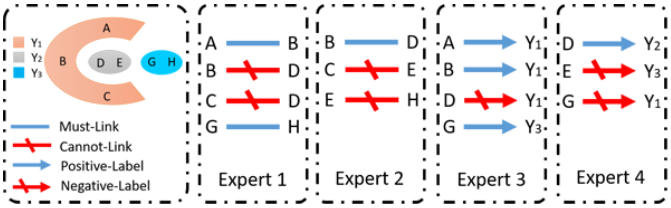


Fig. 1. Example of a data set with 3 clusters and constraints from 4 experts.

- For different types of constraints, we analyze their relations and propose an uniform representation for them, which converts each of them into a pairwise relation matrix.
- We propose an optimization model for semi-supervised clustering with multi-source constraints to find out a clustering with good cluster structure and high consensus of all the sources of constraints.
- The proposed algorithm can be used for not only multi-source constraints but also single-source constraints. Compared to existing semi-supervised clustering algorithms, the proposed algorithm is highly adaptable for different types of constraints.

The remainder of the paper is organized as follows. Section 2 proposes an uniform representation for different types of constraints. Section 3 presents a semi-supervised clustering algorithm with multi-source constraints. Section 4 evaluates the performance of the proposed algorithm using several benchmark data sets. Section 5 concludes the paper with some remarks.

2 An Uniform Representation for Different Types of Constraints

Let $\mathbb{X}$ be a $n \times d$ data matrix, where $\mathbf{x}_i$ is the ith row of $\mathbb{X}$ which represents the ith object with d features, and k be the number of clusters. Pairwise constraints and label constraints are introduced as follows.

(1) *Pairwise constraints* reflect the relations between objects on $\mathbb{X}$, which include Must-Link and Cannot-Link constraints. For objects $\mathbf{x}_i$ and $\mathbf{x}_j$, if they are thought to belong to the same cluster, their pairwise relation is seen as a Must-Link constraint. If they are thought to belong to different clusters, their pairwise relation is seen as a Cannot-Link constraint. The pairwise constraints can be formally described as follows. $M = \{< \mathbf{x}_i, \mathbf{x}_j >: \mathbf{x}_i \in \mathbf{c}_l, \mathbf{x}_j \in \mathbf{c}_h, l = h\}$ is a set of Must-Link constraints and $C = \{< \mathbf{x}_i, \mathbf{x}_j >: \mathbf{x}_i \in \mathbf{c}_l, \mathbf{x}_j \in \mathbf{c}_h, l \neq h\}$ is a set of Cannot-Link constraints, where $\mathbf{c}_l$ is the lth desired cluster, for $1 \leq l \leq k$. A set of pairwise constraints is used to being represented by a $n \times n$ relation matrix A. If $< \mathbf{x}_i, \mathbf{x}_j > \in M$, $A_{(ij)}$ is set to a positive value. If $< \mathbf{x}_i, \mathbf{x}_j > \in C$, $A_{(ij)}$ is set to a negative value. In other cases, $A_{(ij)}$ is set to 0.

(2) *Label constraints* reflect the relations between objects and "true" classes, which include Positive-Label and Negative-Label constraints. Let $\mathbf{y}_l$ be the lth class, for $1 \leq l \leq k$. It is assumed that the number of "true" classes is equal to the number of desired clusters. However, it is worth noting that the "true" classes maybe not consistent with the desired clusters, i.e., $\mathbf{y}_l$ is not necessarily equal to $\mathbf{c}_l$, for $1 \leq l \leq k$. For object $\mathbf{x}_i$ and class $\mathbf{y}_l$, if $\mathbf{x}_i$ is thought to belong to $\mathbf{y}_l$, their relation is seen as a Positive-Label constraint. If $\mathbf{x}_i$ is thought to not belong to $\mathbf{y}_l$, their relation is seen as a Negative-Label constraint. The label constraints can be formally described as follows. Let $P = \{< \mathbf{x}_i, \mathbf{y}_l >: \mathbf{x}_i \in \mathbf{y}_l\}$ be a set of Positive-Label constraints and $N = \{< \mathbf{x}_i, \mathbf{y}_l >: \mathbf{x}_i \notin \mathbf{y}_l\}$ be a set of Negative-Label constraints. A set of label constraints is used to being represented by a $n \times k$ membership matrix Q. If $< \mathbf{x}_i, \mathbf{y}_l > \in P$, $Q_{(il)}$ is set to a positive value. If $< \mathbf{x}_i, \mathbf{y}_l > \in N$, $Q_{(il)}$ is set to a negative value. In other cases, $Q_{(il)}$ is set to 0.

According to the above introductions, we observe that the representation of pairwise constraints is different from label constraints. In order to integrate them, we need to convert them into an uniform representation. We found that converting label constraints into a $n \times n$ pairwise matrix is very easier than converting pairwise constraints into a $n \times k$ membership matrix. Thus, we define a $n \times n$ pairwise matrix to represent and save the two types of constraints, which is formalized as follows. Let $G = [G_{(ij)}]$ is a $n \times n$ pairwise matrix, where $G_{(ij)}$ is a prior relation between $\mathbf{x}_i$ and $\mathbf{x}_j$. $G_{(ij)} = 0$ means no prior relation between them.

According to the definition of pairwise constraints, we can see that pairwise constraints can be directly saved

into G, i.e., $G = A$. Given a set of Must-Link constraints M and a set of Cannot-Link constraints C, the pairwise matrix G can be defined as

$$G_{(ij)} = \begin{cases} \lambda_+, & < \mathbf{x}_i, \mathbf{x}_j > \in M, \\ \lambda_-, & < \mathbf{x}_i, \mathbf{x}_j > \in C, \\ 0, & otherwise, \end{cases} \tag{1}$$

where $\lambda_+ (\geq 0)$ is a reward factor and $\lambda_- (\leq 0)$ is a penalty factor. They are used to stimulate a clustering algorithm to comply with the pairwise constraints.

For label constraints, we need to convert the membership matrix Q into a pairwise matrix G. In order to effectively solve this problem, we do two important works as follows.

(1) *Designing conversion rules*. For objects $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{X}$, we define the following conversion rules: (a) If they have the same positive labels, there should be a Must-Link constraint between them, i.e., $G_{(ij)} = \lambda_+$; (b) If they have different positive labels, there should be a Cannot-Link constraint between them, i.e., $G_{(ij)} = \lambda_-$; (c) If the positive label of $\mathbf{x}_i$ is the same as negative label of $\mathbf{x}_j$, there should also be a Cannot-Link constraint between them, i.e., $G_{(ij)} = \lambda_-$; (d) If they have the same $k-1$ negative labels, there should be a Must-Link constraint between them, i.e., $G_{(ij)} = \lambda_+$; (e) If both they have $k-1$ negative labels and the number of their common negative labels is less than $k-1$, there should be a Cannot-Link constraint between them, i.e., $G_{(ij)} = \lambda_-$; (f) In other cases, $G_{(ij)} = 0$.

According to these rules, we can see that there is no information loss in the process of converting Positive-Label constraints. However, if we use only the negative labels, we do not easily determine the pairwise constraints between objects, unless they all have $k-1$ negative labels. For example, if there are two Negative-Label constraints $< \mathbf{x}_i, \mathbf{y}_l >$, $< \mathbf{x}_j, \mathbf{y}_l > \in N$, we only can know that both $\mathbf{x}_i$ and $\mathbf{x}_j$ do not belong to Class $\mathbf{y}_l$. However, we can not determine whether or not they should belong to the same cluster. While implementing semi-supervised clustering, no constraint is often better than "uncertain" constraints. Therefore, although we lose some information in the process of converting Negative-Label constraints, setting $G_{(ij)} = 0$ in the uncertain cases can reduce the effect of the uncertainty of Negative-Label constraints on the semi-supervised clustering.

(2) *Sufficiently exploiting label constraints*. It is worth noting that a Positive-Label constraint $< \mathbf{x}_i, \mathbf{y}_l > \in P$ is a strong constraint. It contains not only a Positive-Label constraint but also $k-1$ Negative-Label constraints, i.e., $\{< \mathbf{x}_i, \mathbf{y}_h >: h \neq l\}$. We should add these Negative-Label constraints to N. A Negative-Label constraint $< \mathbf{x}_i, \mathbf{y}_h > \in N$ is a weak constraint. Object $\mathbf{x}_i$ is thought to belong to $\mathbf{y}_l$, only if there are $k-1$ Negative-Label constraints between $\mathbf{x}_i$ and $\mathbf{y}_h$ ($h \neq l$). Therefore, if $|\{< \mathbf{x}_i, \mathbf{y}_h > \in N, , h \neq l\}| = k-1$, $< \mathbf{x}_i, \mathbf{y}_l >$ should be added to P. In order to sufficiently consider the supervised information of the label constraints in the process of converting them, we define a Positive-Label constraint set P^+ and a Negative-Label constraint set N^+, instead of P and N, as follows. $P^+ = P \cup \{< \mathbf{x}_i, \mathbf{y}_l > | \forall h \neq l, < \mathbf{x}_i, \mathbf{y}_h > \in N\}$ and $N^+ = N \cup \{< \mathbf{x}_i, \mathbf{y}_{h \neq l} > | < \mathbf{x}_i, \mathbf{y}_l > \in P\}$.

Given a set of Positive-Label constraints P^+ and a set of Negative-Label constraints N^+, the pairwise matrix G can be defined as

$$G_{(ij)} = \begin{cases} \lambda_+, & \exists_l < \mathbf{x}_i, \mathbf{y}_l >, < \mathbf{x}_j, \mathbf{y}_l > \in P^+ \text{ or} \\ & |\{\mathbf{y}_l | < \mathbf{x}_i, \mathbf{y}_l >, < \mathbf{x}_j, \mathbf{y}_l > \in N^+\}| = k-1, \\ \lambda_-, & \exists_{l \neq h} < \mathbf{x}_i, \mathbf{y}_l >, < \mathbf{x}_j, \mathbf{y}_h > \in P^+ \text{ or} \\ & \exists_l < \mathbf{x}_i, \mathbf{y}_l > \in P^+, < \mathbf{x}_j, \mathbf{y}_l > \in N^+, \\ 0, & otherwise. \end{cases} \tag{2}$$

Based on Eqs. (1) and (2), an uniform representation method for pairwise and label constraints is provided. The advantages of the proposed uniform representation are shown as follows.

(1) *It can compare different types of constraints in an uniform pairwise space and reduce the uncertainty of Negative-Label constraints*. According to the proposed method, we can see that different types of constraints contain different quantities of supervision information. Among them, Positive-Label constraint is the strongest type of constraints. For example, p Positive-Label constraints can be converted into p^2 pairwise constraints. Negative-Label constraint is the most uncertain type of constraints. At worst, we can not get any pairwise constraints from the given Negative-Label constraints.

(2) *It can solve the misalignment between classes from different sources*. For example, there are two sets of label constraints from Expert 1 and 2, respectively. We assume Q_1 and Q_2 are the membership matrices of the constraints from Expert 1 and 2, respectively. G_1 and G_2 are the converted pairwise matrices of Q_1 and Q_2, respectively. Here, class $\mathbf{y}_l$ perceived by Expert 1 does not necessarily correspond to class $\mathbf{y}_l$ perceived by Expert 2, for $1 \leq l \leq k$. If we use the difference between their membership matrices, e.g., $||Q_1 - Q_2||_F^2$ to evaluate the consensus between their label constraints, the misalignment problem is ignored, which may affect the evaluating result. Using their pairwise relation matrices, e.g., $||G_1 - G_2||_F^2$, instead of $||Q_1 - Q_2||_F^2$, can effectively solve the misalignment.

3 Optimization Model for Clustering with Multi-Source Constraints

While clustering a data set with constraints from m different sources, we can save each source of constraints into a $n \times n$ pairwise matrix, according to the proposed representation method. Let $\mathbb{G} = \{G_t\}_{t=1}^m$ be a set of constraints from m sources, where $G_t = [G_{t(ij)}]$ is a $n \times n$ matrix which is used to save constraints from the tth source. Given a data set $\mathbb{X}$ and multi-source constraints $\mathbb{G}$, our clustering goal is to find out a clustering which has good cluster structure and high consensus of all

the sources of constraints. Let $U = [U_{(il)}]$ be a $n \times k$ membership matrix of $\mathbb{X}$, where $U_{(il)} \in \{0, 1\}$ is the membership degree of object $\mathbf{x}_i$ to the lth cluster. We use U to save the final clustering result. In order to achieve the clustering goal, the optimization problem of a semi-supervised clustering is defined as follows.

$$\min \Theta(U) = F(U) + \alpha E(U), \tag{3}$$

where

$$F(U) = \|\mathbb{X} - UZ\|_F^2, \tag{4}$$

and

$$E(U) = \frac{1}{2}\sum_{t=1}^{m}\left\|G_t - \hat{U}\hat{U}^T\right\|_F^2, \tag{5}$$

where Z is a $k \times d$ center matrix, $\hat{U} = UD^{-1/2}$ is the normalized matrix of U, $D = [D_l]$ is a $k \times k$ diagonal matrix with $D_l = \sum_{i=1}^{n} U_{(il)}$, and $\alpha(\geq 0)$ is a non-negative parameter used to balance the importance of F and E. According to the definition, we can see that the objective function (3) is made up of two terms. Next, we illustrate the roles of the terms in the clustering process.

The first term F is a validity function that is used to evaluate the internal quality (the within-cluster or between-cluster similarity) of a clustering. For the function F, we could use the objective functions of existing clustering algorithms, such as k-means [7], [38], spectral clustering [24], [25] and NMF [22] clustering, to define it. In this paper, the function F is defined based on the objective function of k-means. The less the function F value is, the more similar the objects within the same clusters are. Furthermore, we can obtain the following equation [39]

$$\min \|\mathbb{X} - UZ\|_F^2 = \min Tr(K) - Tr(\hat{U}^T K \hat{U}), \tag{6}$$

where $K = [K_{(ij)}]$ is a $n \times n$ dot product matrix and $K_{(ij)}$ is the dot product between $\mathbf{x}_i$ and $\mathbf{x}_j$. Since $Tr(K)$ is a constant, the minimization problem of the function F becomes

$$\max_{\hat{U}} Tr(\hat{U}^T K \hat{U}), \text{ s.t. } \hat{U}^T\hat{U} = I_k, \tag{7}$$

where I_k is a $k \times k$ identity matrix. This is the standard trace minimization problem. It is solved by the matrix $\hat{U}$ which contains the first k eigenvectors of K as rows. K can be computed by a kernel function. In this case, the function F becomes the objective function of kernel k-means [38]. If we replace K with the normalized kernel matrix $\hat{K} = \Lambda^{-1/2} K \Lambda^{-1/2}$, where $\Lambda = [\lambda_j]$ is a $n \times n$ diagonal matrix with $\lambda_j = \sum_{i=1}^{n} K_{(ij)}$, the function F is equivalent to the normalized spectral clustering algorithm, which had been proved in [40]. In this paper, we select Gaussian kernel function which is one of widely-used kernel functions in cluster analysis, such as kernel k-means, spectral clustering, and mean-shift algorithms, to compute the dot product matrix. Compared to directly compute them in the original feature space, the kernel function can map the data set into a high-dimensional space, where objects are easily linearly separable.

The second term E is a consensus function that is used to evaluate the differences between a clustering and all the sources of constraints. Since the constraints from each source is saved into a $n \times n$ relation matrix, we use $\hat{U}\hat{U}^T$, instead of U, to represent the pairwise relation of the clustering result U. We use Frobenius norm to measure the difference between constraints G_t and $\hat{U}\hat{U}^T$. We wish that the clustering result is close to each source of constraints. We try to minimize E to get the most consensus of all the sources of constraints. Furthermore, we can get the following equation

$$\begin{aligned}\left\|G_t - \hat{U}\hat{U}^T\right\|_F^2 &= Tr((G_t - \hat{U}\hat{U}^T)^T(G_t - \hat{U}\hat{U}^T)) \\ &= Tr(G_t^T G_t) + Tr(I_n) - 2Tr(\hat{U}^T G_t \hat{U}).\end{aligned} \tag{8}$$

Since $Tr(G_t^T G_t) + Tr(I_n)$ is a constant, the minimization problem of the function E becomes

$$\max_{\hat{U}} Tr\left(\hat{U}^T\left(\sum_{t=1}^{m} G_t\right)\hat{U}\right), \text{ s.t. } \hat{U}^T\hat{U} = I_k. \tag{9}$$

This is also the standard trace minimization problem. It is solved by the matrix $\hat{U}$ which contains the first k eigenvectors of $\sum_{t=1}^{m} G_t$ as rows.

In this paper, we use the eigenvalue decomposition (EVD) to solve the optimization problem $\Theta(U)$. Based on Eqs. (7) and (9), we can see that minimizing $\Theta(U)$ is equivalent to

$$\max_{\hat{U}} Tr\left(\hat{U}^T\left(K + \alpha\sum_{t=1}^{m} G_t\right)\hat{U}\right), \text{ s.t. } \hat{U}^T\hat{U} = I_k. \tag{10}$$

Thus, the optimization problem can be converted into an eigenvalue decomposition problem. Let $\Phi = K + \alpha\sum_{t=1}^{m} G_t$. $\hat{U}$ can be obtained by taking the top k eigenvectors of Φ. The lowest computational complexity of $\hat{U}$ is $O(n^2k)$. However, the worst computational complexity is $O(n^3)$. Furthermore, we treat each row of $\hat{U}$ as a vertex in $\mathbb{R}^k$ and cluster these vertices to get U via the ward-linkage algorithm [41] which is a widely-used hierarchical clustering algorithm.

Algorithm 1: SC-MPI

Input: K, $\mathbb{G}$, k, α
Output: U
Compute $K + \alpha\sum_{t=1}^{m} G_t$ and get its top k eigenvectors;
Construct the matrix $\hat{U} \in \mathbb{R}^{n\times k}$ from the eigenvectors;
Cluster $\hat{U}$ to get U via the ward-linkage algorithm;

The proposed algorithm is formalized in Algorithm 1, called SC-MPI. Its time complexity is $O(n^2 + n^2k + n^2\log n + \sum_{t=1}^{m} p_t^2)$, where $O(n^2)$, n^2k, $n^2\log n$ and $O(\sum_{t=1}^{m} p_t^2)$ are the time costs of getting the dot produce matrix K, fast obtaining the top k eigenvectors,

implementing the ward-linkage algorithm, and converting m sources of constraints into pairwise matrices $\mathbb{G}$, respectively. Here, p_t is the number of constraints in the tth source. Its space complexity is $O(nd+n^2+\sum_{t=1}^{m} p_t^2)$, where $O(nd)$, $O(n^2)$ and $O(\sum_{t=1}^{m} p_t^2)$ are the space costs of saving the data set $\mathbb{X}$, the dot product matrix K and the converted pairwise matrices $\mathbb{G}$, respectively.

Before implementing the proposed algorithm, we need to set the parameters α, λ_+ and λ_- which affect its performance. The parameter α is used to balance the importance of the terms F and E. We assume that the α value is in the interval $[0, +\infty)$. If $\alpha = 0$, the constraints does not work in the clustering process. However, if the α value is very large, the constraints may break the cluster structure on the data set. In this paper, we suggest to setting $\alpha = 1$ which means the first term F is the same important as the second term E.

The parameters λ_+ and λ_- are used to control the role of the constraints. If λ_+ and λ_- are equal to 0, the proposed algorithm becomes an unsupervised clustering algorithm. If λ_+ is very large or λ_- is very small, the connectivity of the similarity graph of objects on a data set may be broken. This may lead that the cluster structure in the similarity graph is not easily found. For example, there are three objects $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, and a Must-Link constraint between $\mathbf{x}_1$ and $\mathbf{x}_2$. We assume $\mathbf{x}_3$ is the nearest neighbor of $\mathbf{x}_1$. Based on the assumption, we know the similarity between $\mathbf{x}_1$ and $\mathbf{x}_3$ should be higher than that between $\mathbf{x}_1$ and $\mathbf{x}_2$. After adding the constraint to the clustering process, the similarity graph is changed. If λ_+ is set to a very big value, the similarity between $\mathbf{x}_1$ and $\mathbf{x}_3$ may be far lower than that between $\mathbf{x}_1$ and $\mathbf{x}_2$. In this case, the margin between $\mathbf{x}_1$ and $\mathbf{x}_3$ may be mistaken as the margin between clusters by the clustering algorithm. This may affect the clustering result. Furthermore, we take another example, where there are three objects $\mathbf{x}_4$, $\mathbf{x}_5$ and $\mathbf{x}_6$ and a Cannot-Link constraint between $\mathbf{x}_4$ and $\mathbf{x}_5$. We assume that the similarity between $\mathbf{x}_4$ and $\mathbf{x}_5$ is higher than that between $\mathbf{x}_4$ and $\mathbf{x}_6$. After adding the constraint to the clustering process, the similarity graph is also changed. If λ_- is set to a very small value, the similarity between $\mathbf{x}_4$ and $\mathbf{x}_5$ may be far higher than that between $\mathbf{x}_4$ and $\mathbf{x}_6$. This may lead that the possibility that $\mathbf{x}_4$ and $\mathbf{x}_5$ are partitioned into the same cluster overly increases.

Therefore, in this paper, we set different λ_+ and λ_- for each pair of objects. For objects $\mathbf{x}_i$ and $\mathbf{x}_j$, $K_{(ij)}$ reflects their similarity in a kernel-based feature space. $\max(K)$ is the maximum similarity between all the objects. $\min(K)$ is the minimum similarity between all the objects. We wish that the changed values of the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ brought by λ_+ and λ_- are no more than $\max(K) - K_{(ij)}$ and $K_{(ij)} - \min(K)$, respectively. Besides, in this paper, we consider constraints from multiple information sources. Thus, for $\mathbf{x}_i$ and $\mathbf{x}_j$, we use the average changed values to set λ_+ and λ_-. Therefore, we define $\lambda_+ = \frac{1}{\gamma_{(ij)}}\left(\max(K) - K_{(ij)}\right)$ and $\lambda_- = \frac{1}{\gamma_{(ij)}}\left(\min(K) - K_{(ij)}\right)$, where $\gamma_{(ij)}$ is the number of the constraints between $\mathbf{x}_i$ and $\mathbf{x}_j$ from m sources.

4 Experimental Analysis

In this section, we conduct the experiments on 12 benchmark data sets from [42] and [43] to compare the proposed algorithm with eight state-of-the-art semi-supervised clustering algorithms. These data sets include different types, such as face image data (Yale and ORL), banknote image data (Banknote), spoken letter recognition data (Isolet), satellite image data (Statlog), handwritten digit data (COIL, OpticalDigits, PenDigits, USPS and MNIST) and handwritten letter data (Letters). Table 1 shows the details of the tested data sets. We employ the external indices ARI [44] and NMI [45] to evaluate the clustering accuracy. All the experiments are carried out in Matlab 2016b on a PC with an Intel i7-4710MQ CPU@2.5Hz and 16GB of RAM.

TABLE 1
Description of data sets

Data Sets	#Ojbects	#Features	#Classes
Yale [43]	165	1024	15
ORL [43]	400	1024	40
Banknote [42]	1,372	4	2
COIL20 [43]	1,440	1024	20
Isolet [43]	1,560	617	26
OpticalDigits [42]	5,620	64	10
Statlog [42]	6,435	36	6
COIL100 [43]	7,200	1024	100
MNIST [43]	10,000	784	10
PenDigits [43]	10,992	16	10
USPS [43]	11,000	256	10
Letters [42]	20,000	16	26

In the experiment, we first compare the proposed algorithm with other semi-supervised clustering algorithms, given single-source constraints. The experiment is made up of three parts as follows.

Part 1: Given a set of pairwise constraints (Must-Link and Cannot-Link), we present the performance of the proposed algorithm, compared to existing pairwise-constrained clustering algorithms including COP [6], CVQE [9] and CPLP [15]. In this comparison, we set the size of the pairwise constraints, according to the number of objects n on each data set. On a data set, we set the number of pairwise constraints, where the number of Must-Link is equal to that of Cannot-Link, to 20%, 40%, 60%, 80% and 100% of n, respectively. According to each size, we randomly select the pairwise constraints on each data set.

Part 2: Given a set of Positive-Label constraints, we present the performance of the proposed algorithm, compared to existing positive-label-constrained clustering algorithms including LP [26], NLP [46], CNMF [23] and PLCC [21]. In this comparison, we set the size of the positive labels, according to the number of objects n on each data set. On a data set, we set the number of

positive labels to 10%, 20%, 30%, 40% and 50% of n, respectively. According to each size, we randomly select the positive labels on each data set.

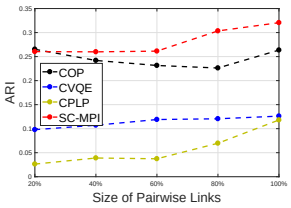
Part 3: Given a set of Positive-Label and Negative-Label constraints, we present the performance of the proposed algorithm, compared to the label-constrained clustering algorithm PNLP [27]. In this comparison, we set the size of the labels, according to the number of objects n on each data set. On a data set, we set the number of labels, where the number of positive labels is equal to that of negative labels, to 10%, 20%, 30%, 40% and 50% of n, respectively. According to each size, we randomly select the label constraints on each data set.

In the comparisons, for each algorithm, we set the number of clusters k is equal to its true number of classes on each of the given data sets. Furthermore, we select Gaussian kernel function, which is one of widely-used kernel functions in cluster analysis, to compute the dot product matrix. The kernel parameter is set to the variance of a data set. Besides, the compared algorithms need to set the parameters which are similar to α to regulate the role of the constraints in clustering a data set. In the experiment, for each of the compared algorithms, we assume the importance of constraints is equal to that of clustering validity. Thus, we set $\alpha = 1$ for the proposed algorithm. Besides, for the COP, CVQE, CNMF and PLCC algorithms, their clustering effectiveness depends on the selection of initial points. Thus, we run each of them 50 times and select the highest NMI and ARI values for them in the comparisons.

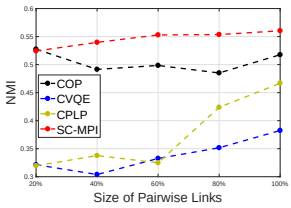
Fig. 2 shows the comparison of COP, CVQE, CPLP and the proposed algorithm with different sizes of pairwise constraints on the tested data sets. According to the these figures, we can see that the proposed algorithm outperforms other algorithms on these data sets. Fig. 3 shows the comparison of LP, NLP, CNMF, PLCC and the proposed algorithm with different sizes of positive labels on the tested data sets. According to the these figures, we can see that the performances of LP, NLP and the proposed algorithm are superior to CNMF and PLCC. We also can see that the clustering accuracies of the proposed algorithm are very close to LP and NLP on most of the data sets. On some data sets, the performance of the proposed algorithm is slightly better than LP and NLP. Fig. 4 shows the comparison of PNLP and the proposed algorithm with different sizes of positive and negative labels on the tested data sets. We can observe that the proposed algorithm is obviously better than PNLP. According to the above figures, we can see that the proposed algorithm has very good performance on clustering most of the data sets, compared to other algorithms. Besides, the proposed algorithm can deal with different types of constraints and provide good clustering results in the situation of single-source constraints. Here, we need to discuss why the proposed algorithm with single-source constraints has very good performance, compared to other algorithms. There are three main reasons: (a) While dealing with pairwise constraints, the proposed algorithm can obtain an approximately and globally optimal solution by the eigenvalue decomposition method. However, the compared algorithms with pairwise constraints (COP, CVQE and CPLP) do not easily obtain their optimal solutions. (b) For Positive-Label constraints, the proposed representation method can overcome a misalignment problem between true class labels and cluster labels. The compared algorithms with label constraints (LP, NLP, CNMF and PLCC) directly compare the membership matrices of label constraints and clustering result, which does not fully consider the misalignment problem. (c) For Negative-Label constraints, we delete their uncertain constraints which affects the performance of the PNLP algorithm.

Next, we test the performance of the proposed algorithm with multi-source constraints. In the experiment, we first randomly select a set of $50\%n$ pairwise constraints (including 50% Must-Link and 50% Cannot-Link) and a set of $50\%n$ label constraints including (50% Positive-Label and 50% Negative-Label) on each data set. Furthermore, we add a certain proportion of incorrect constraints to each of the given sets of pairwise or label constraints. For each data set, we assume the proportion of the incorrect constraints to the number of constraints is 10%, 20%, 30%, 40% and 50%, respectively. Given a proportion value, we make use of a set of pairwise or label constraints on a data set to produce 5 different sets of constraints, each of which is made up of the given set of constraints and a set of the randomly produced incorrect constraints. Therefore, for each data set, we produce 10 sets of constraints including 5 sets of pairwise constraints and 5 sets of label constraints, according to the fixed proportion of incorrect information. Fig. 5 presents the performance of the proposed algorithm with the 10 constraint sets, compared to the best performance of it with each of the pairwise constraint sets and each of the label constraint sets, in the cases of different sizes of incorrect information. In the figures, 'SLCS', 'SPCS' and 'MCS' denote the highest ARI and NMI values of the proposed algorithm with one of the five label constraint sets, the highest ARI and NMI values of the proposed algorithm with one of the five pairwise constraint sets, and the ARI and NMI values of the proposed algorithm with the 10 constraint sets, respectively. According to the figures, we see that the incorrect information can reduce the clustering effectiveness of the proposed algorithm. As the proportion of the incorrect information increases, the ARI and NMI values of the proposed algorithm decreases. We also observe that the performance of the proposed algorithm with multiple constraint sets are better than it with single constraint set. The experiment results tell us that the proposed algorithm can integrate these constraints to reduce the effect of incorrect information and enhance the robustness of the clustering results.

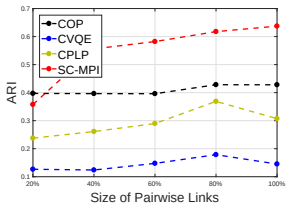
Furthermore, we analyze the effect of the parameter α on the performance of the proposed algorithm on 12 data



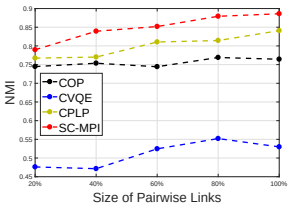
(a) ARI against size of pairwise links on data set Yale



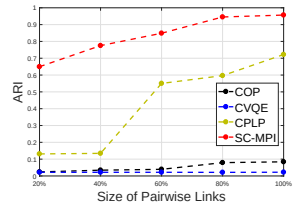
(b) NMI against size of pairwise links on data set Yale



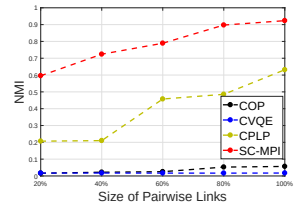
(c) ARI against size of pairwise links on data set ORL



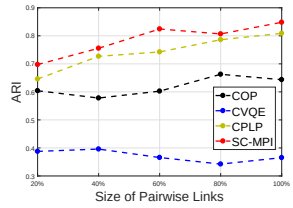
(d) NMI against size of pairwise links on data set ORL



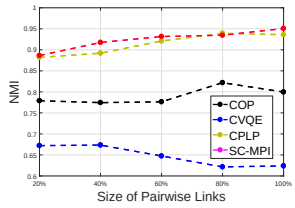
(e) ARI against size of pairwise links on data set Banknote



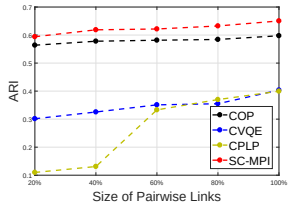
(f) NMI against size of pairwise links on data set Banknote



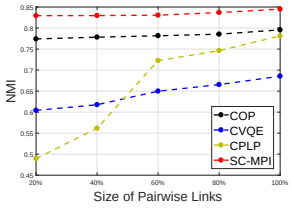
(g) ARI against size of pairwise links on data set COIL20



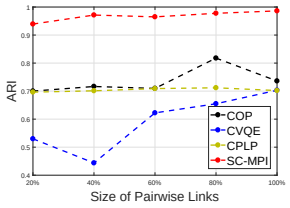
(h) NMI against size of pairwise links on data set COIL20



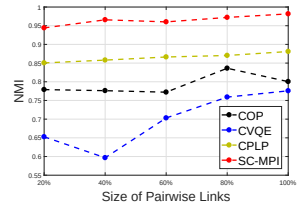
(i) ARI against size of pairwise links on data set Isolet



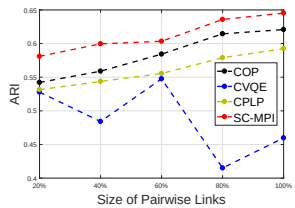
(j) NMI against size of pairwise links on data set Isolet



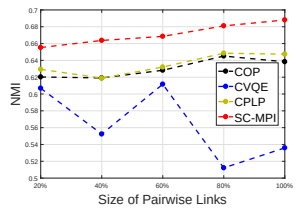
(k) ARI against size of pairwise links on data set OpticalDigits



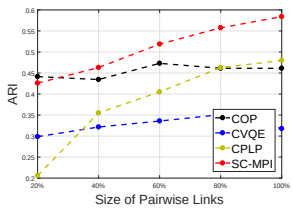
(l) NMI against size of pairwise links on data set OpticalDigits



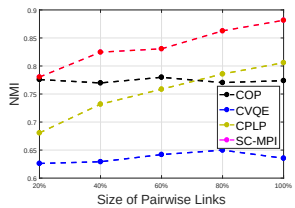
(m) ARI against size of pairwise links on data set Statlog



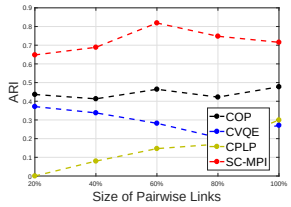
(n) NMI against size of pairwise links on data set Statlog



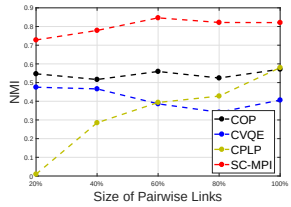
(o) ARI against size of pairwise links on data set COIL100



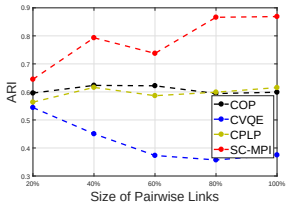
(p) NMI against size of pairwise links on data set COIL100



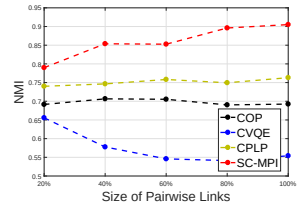
(q) ARI against size of pairwise links on data set MNIST



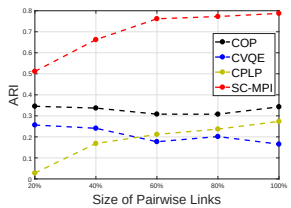
(r) NMI against size of pairwise links on data set MNIST



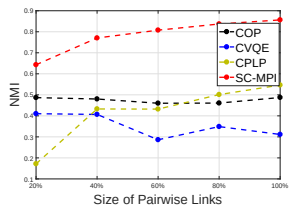
(s) ARI against size of pairwise links on data set PenDigits



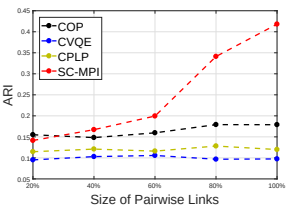
(t) NMI against size of pairwise links on data set PenDigits



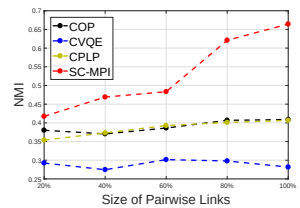
(u) ARI against size of pairwise links on data set USPS



(v) NMI against size of pairwise links on data set USPS

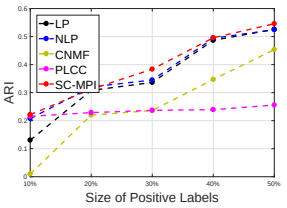


(w) ARI against size of pairwise links on data set Letters

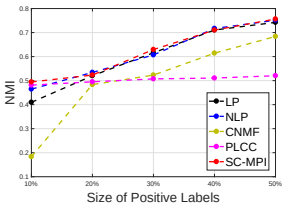


(x) NMI against size of pairwise links on data set Letters

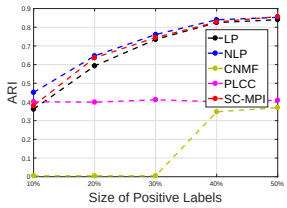
Fig. 2. Comparisons of different algorithms with pairwise constraints.



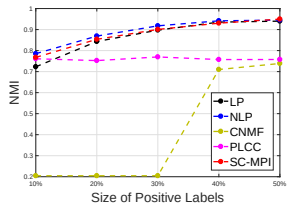
(a) ARI against size of positive labels on data set Yale



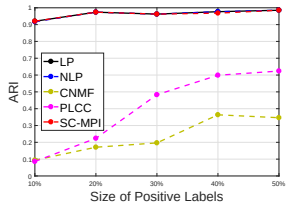
(b) NMI against size of positive labels on data set Yale



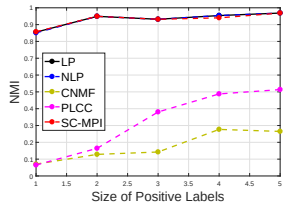
(c) ARI against size of positive labels on data set ORL



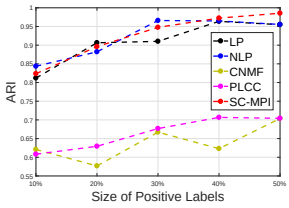
(d) NMI against size of positive labels on data set ORL



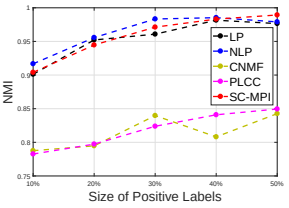
(e) ARI against size of positive labels on data set Banknote



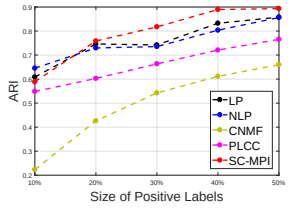
(f) NMI against size of positive labels on data set Banknote



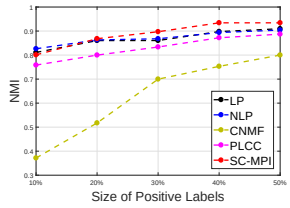
(g) ARI against size of positive labels on data set COIL20



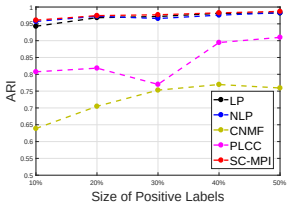
(h) NMI against size of positive labels on data set COIL20



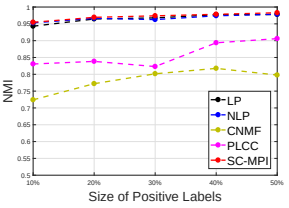
(i) ARI against size of positive labels on data set Isolet



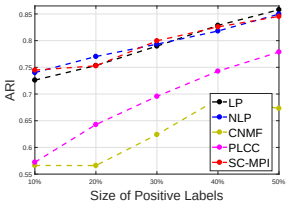
(j) NMI against size of positive labels on data set Isolet



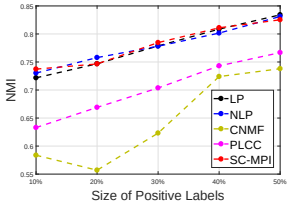
(k) ARI against size of positive labels on data set OpticalDigits



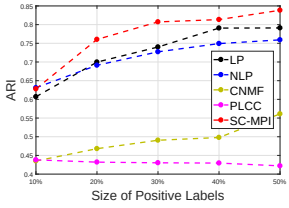
(l) NMI against size of positive labels on data set OpticalDigits



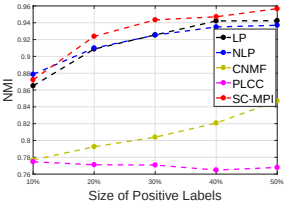
(m) ARI against size of positive labels on data set Statlog



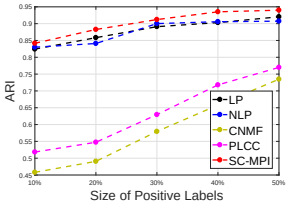
(n) NMI against size of positive labels on data set Statlog



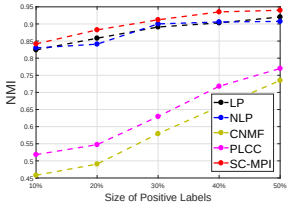
(o) ARI against size of positive labels on data set COIL100



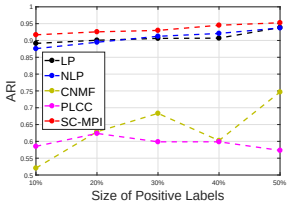
(p) NMI against size of positive labels on data set COIL100



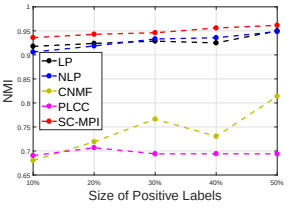
(q) ARI against size of positive labels on data set MNIST



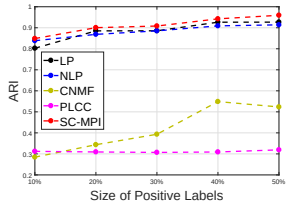
(r) NMI against size of positive labels on data set MNIST



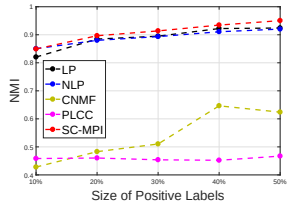
(s) ARI against size of positive labels on data set PenDigits



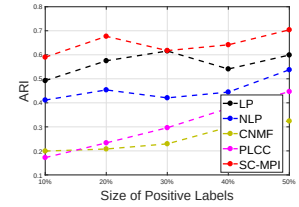
(t) NMI against size of positive labels on data set PenDigits



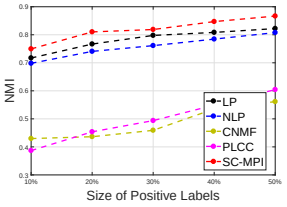
(u) ARI against size of positive labels on data set USPS



(v) NMI against size of positive labels on data set USPS



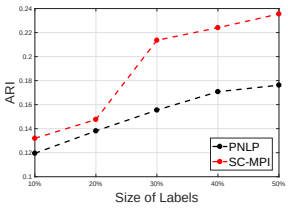
(w) ARI against size of positive labels on data set Letters



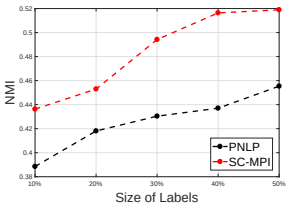
(x) NMI against size of positive labels on data set Letters

Fig. 3. Comparisons of different algorithms with positive labels.

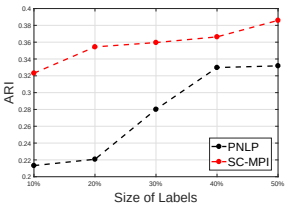
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2979699, IEEE Transactions on Pattern Analysis and Machine Intelligence



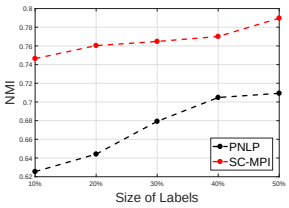
(a) ARI against size of labels on data set Yale



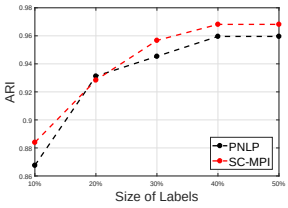
(b) NMI against size of labels on data set Yale



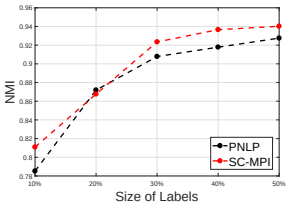
(c) ARI against size of labels on data set ORL



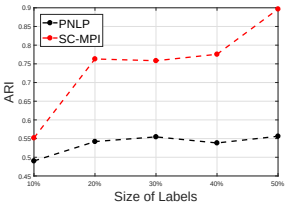
(d) NMI against size of labels on data set ORL



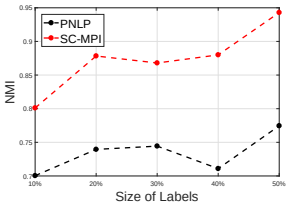
(e) ARI against size of labels on data set Banknote



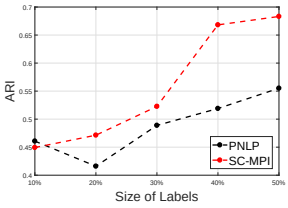
(f) NMI against size of labels on data set Banknote



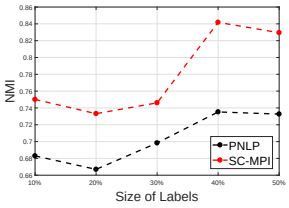
(g) ARI against size of labels on data set COIL20



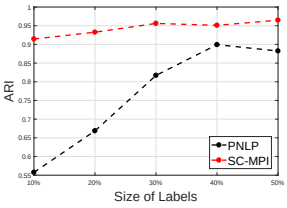
(h) NMI against size of labels on data set COIL20



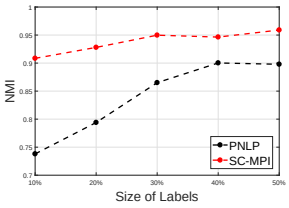
(i) ARI against size of labels on data set Isolet



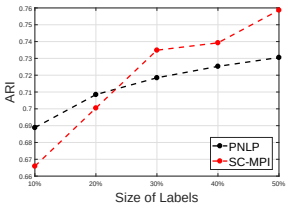
(j) NMI against size of labels on data set Isolet



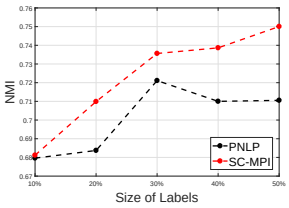
(k) ARI against size of labels on data set OpticalDigits



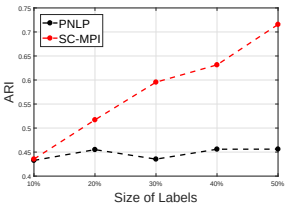
(l) NMI against size of labels on data set OpticalDigits



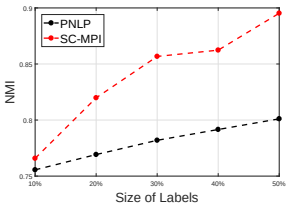
(m) ARI against size of labels on data set Statlog



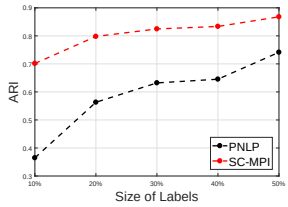
(n) NMI against size of labels on data set Statlog



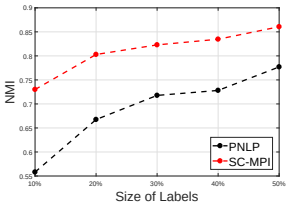
(o) ARI against size of labels on data set COIL100



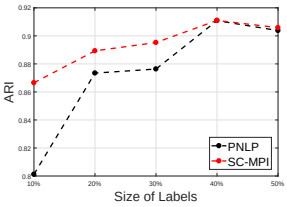
(p) NMI against size of labels on data set COIL100



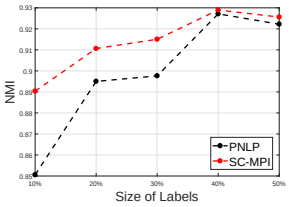
(q) ARI against size of labels on data set MNIST



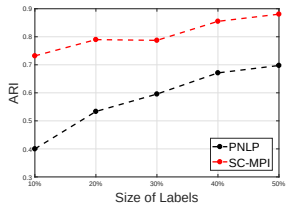
(r) NMI against size of labels on data set MNIST



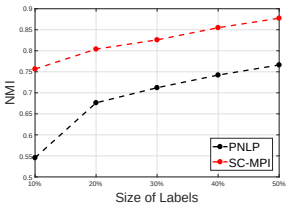
(s) ARI against size of labels on data set PenDigits



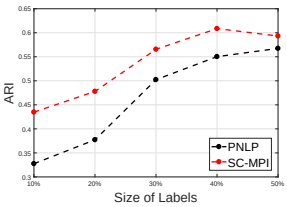
(t) NMI against size of labels on data set PenDigits



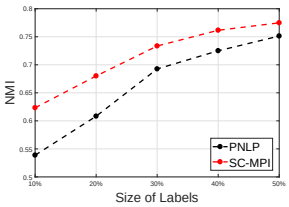
(u) ARI against size of labels on data set USPS



(v) NMI against size of labels on data set USPS

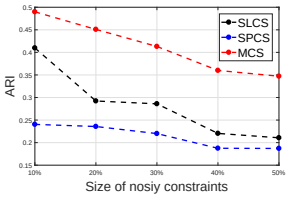


(w) ARI against size of labels on data set Letters

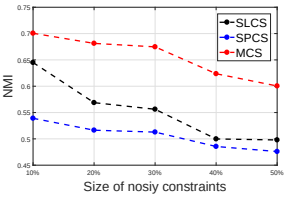


(x) NMI against size of labels on data set Letters

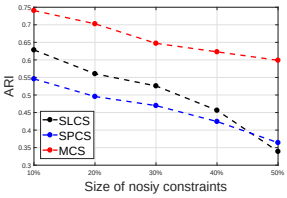
Fig. 4. Comparisons of different algorithms with positive and negative labels.



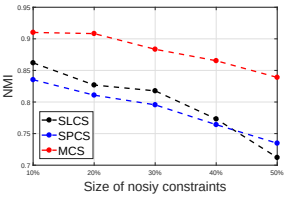
(a) ARI against size of noisy constraints on data set Yale



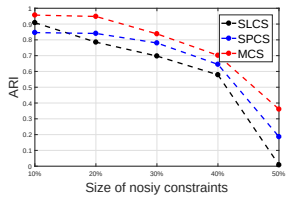
(b) NMI against size of noisy constraints on data set Yale



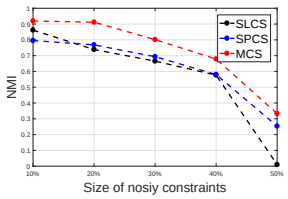
(c) ARI against size of noisy constraints on data set ORL



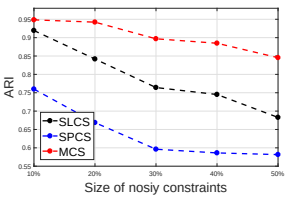
(d) NMI against size of noisy constraints on data set ORL



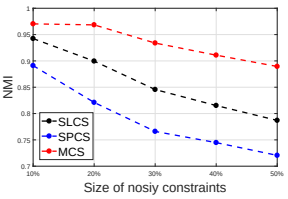
(e) ARI against size of noisy constraints on data set Banknote



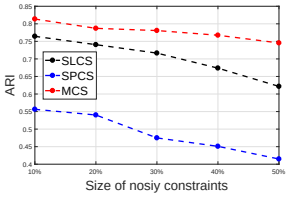
(f) NMI against size of noisy constraints on data set Banknote



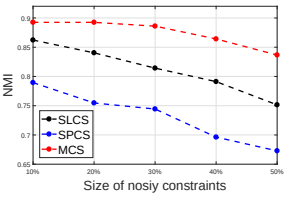
(g) ARI against size of noisy constraints on data set COIL20



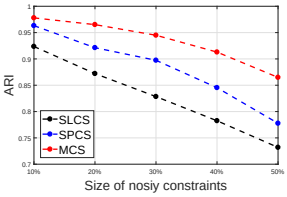
(h) NMI against size of noisy constraints on data set COIL20



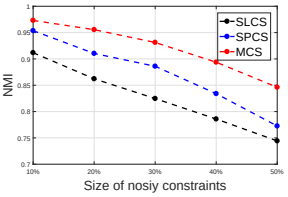
(i) ARI against size of noisy constraints on data set Isolet



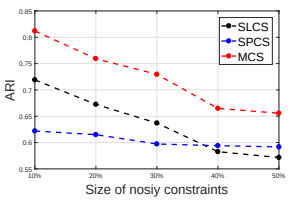
(j) NMI against size of noisy constraints on data set Isolet



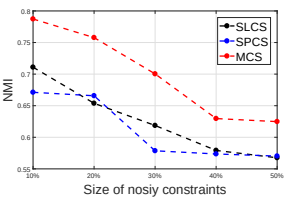
(k) ARI against size of noisy constraints on data set OpticalDigits



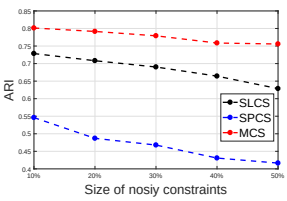
(l) NMI against size of noisy constraints on data set OpticalDigits



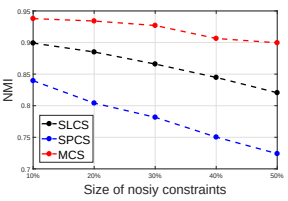
(m) ARI against size of noisy constraints on data set Statlog



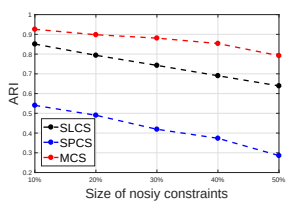
(n) NMI against size of noisy constraints on data set Statlog



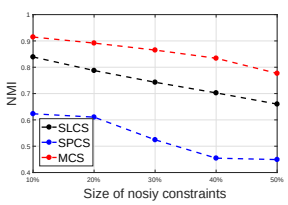
(o) ARI against size of noisy constraints on data set COIL100



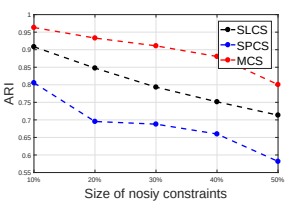
(p) NMI against size of noisy constraints on data set COIL100



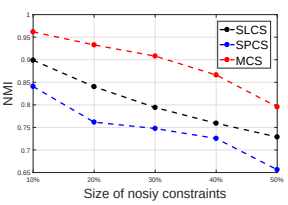
(q) ARI against size of noisy constraints on data set MNIST



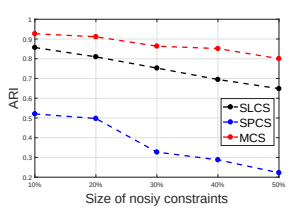
(r) NMI against size of noisy constraints on data set MNIST



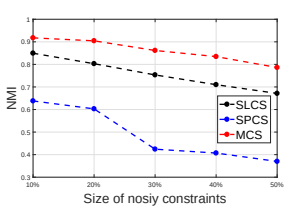
(s) ARI against size of noisy constraints on data set PenDigits



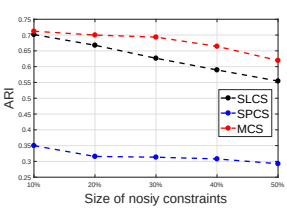
(t) NMI against size of noisy constraints on data set PenDigits



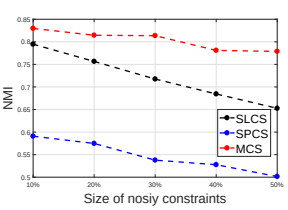
(u) ARI against size of noisy constraints on data set USPS



(v) NMI against size of noisy constraints on data set USPS



(w) ARI against size of noisy constraints on data set Letters



(x) NMI against size of noisy constraints on data set Letters

Fig. 5. Comparisons of the SC-MPI algorithm with different proportions of noisy constraints.

sets, as shown in Fig. 6. In the analysis, we consider the three cases, i.e., the proposed algorithm with pairwise constraints (including 50% Must-Link and 50% Cannot-Link), with label constraints (50% Positive-Label and 50% Negative-Label) and with mixed constraints (50% pairwise and 50% label constraints). We fix the overall number of constraints as $50\%n$ where n is the number of objects on a data set. We test the proposed algorithm with α in the interval $[0, 2]$ and the step length of 0.25. According to these figures, we observe that the effects of the parameter α are different on these data sets. This indicates that it is difficult to select an appropriate α for the proposed algorithm on each data set. In order to further analyze the effect, we show the means of ARI and NMI for the proposed algorithm on all the tested data sets for each α in Fig. 6. According to the mean lines, we see that if $\alpha > 0$, the average performance of the proposed algorithm is relatively stable. Thus, we can see that setting $\alpha = 1$ is not a bad choice and has very good interpretability.

5 Conclusion

In this paper, we propose a new semi-supervised clustering algorithm which can integrate multi-source constraints to guide clustering process. In the algorithm, we first present an uniform representation for different types of constraints, which converts constraints from each source into a pairwise relation matrix. Furthermore, we define an objective function which includes two terms: evaluating the clustering validity and the consensus of all the sources of constraints. We provide its optimization solving method to minimize it. Extensive experiments demonstrate the proposed algorithm is very adaptable for different types of constraints, compared to other semi-supervised clustering algorithms.

Acknowledgement

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 61773247, 61876103, 61976128, 61902227), the Technology Research Development Projects of Shanxi (No. 201901D211192) and the 1331 Engineering Project of Shanxi Province, China.

References

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[2] C. Aggarwal and C. Reddy, Eds., *Data Clustering: Algorithms and Applications*. CRC Press, 2014.

[3] A. Jain, "Data clustering: 50 years beyond k-means," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2008.

[4] S. Basu, M. Bilenko, R. Mooney, and M. Bilenko, "A probabilistic framework for semi-supervised clustering," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 59–68.

[5] X. Zhu and A. Goldberg, Eds., *Introduction to Semi-Supervised Learning*. CA, USA: Morgan Claypool, 2009.

[6] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *Proceedings of the 7th International Conference on Machine Learning*, 2000, pp. 1103–1110.

[7] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[8] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proceedings of International Conference on Machine Learning*, 2001, pp. 577–584.

[9] I. Davidson and S. S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proceedings of SIAM International Conference on Data Mining*, 2005, pp. 201–211.

[10] D. Pelleg and D. Baras, "K-means with large and noisy constraint sets," in *Proceedings of European Conference on Machine Learning*, 2007, pp. 674–682.

[11] S. Anand, S. Mittal, O. Tuzel, and P. Meer, "Semi-supervised kernel mean shift clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1201–1215, 2014.

[12] H. Zeng and Y. Cheung, "Semi-supervised maximum margin clustering with pairwise constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, pp. 926–939, 2012.

[13] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing gaussian mixture models with em using equivalence constraints," in *Advances in Neural Information Processing Systems*, 2004, pp. 465–472.

[14] Z. Li, J. Liu, and X. Tang, "Constrained clustering via spectral regularization," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2009, pp. 421–428.

[15] Z. Lu and Y. Peng, "Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications," *International Journal of Computer Vision*, vol. 103, no. 3, pp. 306–325, 2013.

[16] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 1–30, 2014.

[17] Y. Luo, T. Tian, J. Shi, J. Zhu, and B. Zhang, "Semi-crowdsourced clustering with deep generative models," in *Advances in Neural Information Processing Systems*, 2018, pp. 3216–3226.

[18] S. Xiong, J. Azimi, and X. Fern, "Active learning of constraints for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 43–54, 2014.

[19] T. Craenendonck, S. Dumancic, and H. Blockeel, "Cobra: A fast and simple method for active clustering with pairwise constraints," in *International Joint Conferences on Artificial Intelligence*, 2017.

[20] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in *Proceedings of 19th International Conference on Machine Learning*, 2002.

[21] H. Liu, Z. Tao, and Y. Fu, "Partition level constrained clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2469–2483, 2017.

[22] D. D. Lee and H. H. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 556–562.

[23] H. Wu and Z. Liu, "Non-negative matrix factorization with constraints," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2010.

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

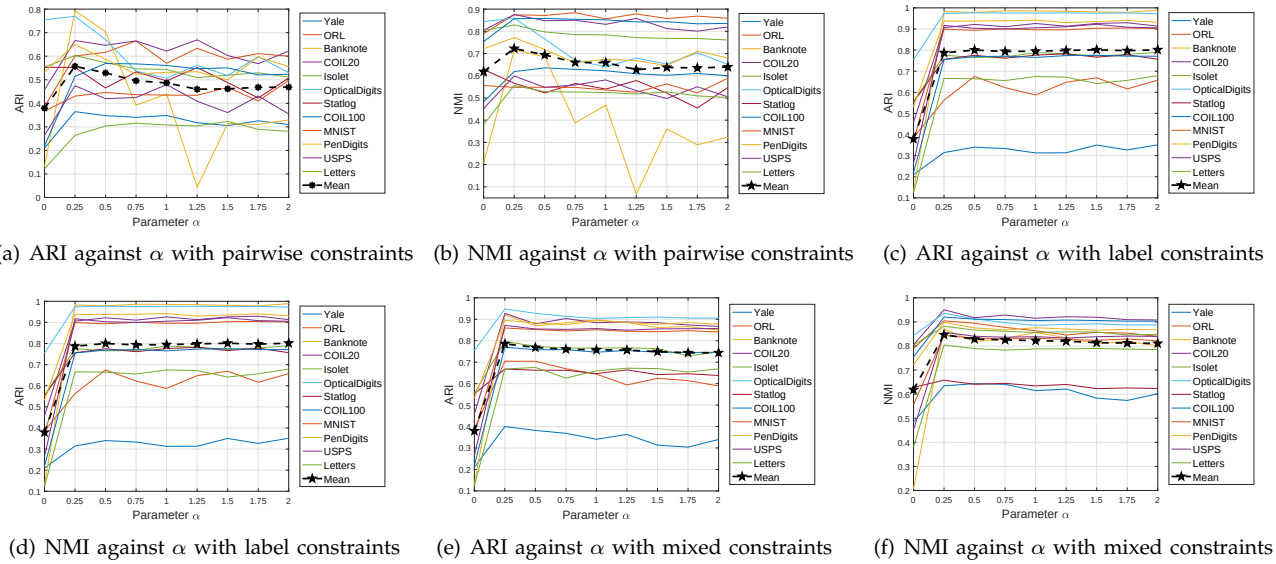
[25] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 849–856.

[26] Z. D., T. Bousquet, O.and Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, 2004, pp. 321–328.

[27] O. Zoidi, A. Tefas, N. Nikolaidis, and I. Pitas, "Positive and negative label propagations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 342–355, 2018.

[28] M. Bilenko, S. Basu, and R. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of International Conference on Machine Learning*, 2004, pp. 81–88.

[29] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Advances in Neural Information Processing Systems*, 2003, pp. 505–512.



(a) ARI against α with pairwise constraints (b) NMI against α with pairwise constraints (c) ARI against α with label constraints

(d) NMI against α with label constraints (e) ARI against α with mixed constraints (f) NMI against α with mixed constraints

Fig. 6. Effect of the parameter α on the performance of the proposed algorithm.

[30] L. Wu, S. Hoi, R. Jin, J. Zhu, and N. Yu, "Learning bregman distance functions for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 478–491, 2012.

[31] J. Davis, B. Kulis, S. Sra, and I. Dhillon, "Information-theoretic metric learning," in *Advances in Neural Information Processing Systems*, 2007.

[32] M. Law, R. Urtasun, and R. Zemel, "Deep spectral clustering learning," in *Proceedings of International Conference on Machine Learning*, 2017.

[33] Z. Lu and H. Ip, "Combining context, consistency, and diversity cues for interactive image categorization," *IEEE Transactions on Multimedia*, vol. 12, no. 3, pp. 194–203, 2010.

[34] I. Ahn and C. Kim, "Face and hair region labeling using semi-supervised spectral clustering-based multiple segmentations," *IEEE Transactions on Multimedia*, vol. 18, no. 7, pp. 1414–1421, 2016.

[35] R. Huang and W. Lam, "An active learning framework for semi-supervised document clustering with language modeling," *Data and Knowledge Engineering*, vol. 68, no. 1, pp. 49–67, 2009.

[36] Z. Yu, Z. Kuang, J. Liu, H. Chen, J. Zhang, J. You, H. Wong, and G. Han, "Adaptive ensembling of semi-supervised clustering solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1577–1590, 2017.

[37] L. Yang, X. Cao, D. Jin, X. Wang, and D. Meng, "A unified semi-supervised community detection framework using latent space graph regularization," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2585–2598, 2015.

[38] B. Scholkopf, A. Smola, E. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[39] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of International Conference on Machine Learning*, 2004.

[40] I. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: a multilevel approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.

[41] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, San Francisco, 2005.

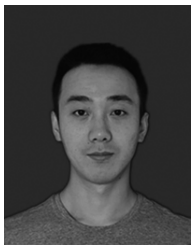
[42] K. Bache and M. Lichman, "Uci machine learning repository," http://archive.ics.uci.edu/ml.

[43] D. Cai, "Codes and datasets for feature learning," http://www.cad.zju.edu.cn/home/dengcai/Data/data.html.

[44] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[45] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Conditional Entropy and Mutual Information*. New York: Cambridge University Press, 2007.

[46] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *International Conference on Machine Learning*, 2006.



Liang Bai received his Ph.D degree in Computer Science from Shanxi University in 2012. He is currently a Professor with the School of Computer and Information Technology, Shanxi University. His research interest is in the areas of cluster analysis. He has published several journal papers in his research fields, including IEEE TPAMI, IEEE TKDE, IEEE TFS, DMKD.

Jiye Liang received the Ph.D degree from Xi'an Jiaotong University. He is a professor in Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, the School of Computer and Information Technology, Shanxi University. His research interests include artificial intelligence, granular computing, data mining, and machine learning. He has published more than 120 papers in his research fields, including IEEE TPAMI, IEEE TKDE, IEEE TFS, DMKD and AI.

Fuyuan Cao received the M.S. and Ph.D degrees in computer science in 2004 and 2009, respectively, from Shanxi University, Taiyuan, China, where he is currently a Professor with the School of Computer and Information Technology. His research interests include data mining and machine learning. He has published several journal papers in his research fields, including IEEE TPAMI, IEEE TNNLS, IEEE TFS and INS.