Joint Representation Learning and Clustering: A Framework for Grouping Partial Multiview Data

Wenzhang Zhuge, Hong Tao, Tingjin Luo, Ling-Li Zeng, Chenping Hou, *Member, IEEE*, and Dongyun Yi

Abstract—Partial multi-view clustering has attracted various attentions from diverse fields. Most existing methods adopt separate steps to obtain unified representations and extract clustering indicators. This separate manner prevents two learning processes to negotiate to achieve optimal performance. In this paper, we propose the Joint Representation Learning and Clustering (JRLC) framework to address this issue. The JRLC framework employs representation matrices to extract view-specific clustering information directly from the presence of partial similarity matrices, and rotates them to learn a common probability label matrix simultaneously, which connects representation learning and clustering seamlessly to achieve better clustering performance. Under the guidance of JRLC framework, several new incomplete multi-view clustering methods can be developed by extending existing single-view graph-based representation learning methods. For illustration, within the framework, we propose two specific methods, JRLC with spectral embedding (JRLC-SE) and JRLC via integrating nonnegative embedding and spectral embedding (JRLC-NS). Two iterative algorithms with guaranteed convergence are designed to solve the resultant optimization problems of JRLC-SE and JRLC-NS. Experimental results on various datasets and news topic clustering application demonstrate the effectiveness of the proposed algorithms.

Index Terms—Representation learning, clustering, partial multi-view data, graph

1 Introduction

With the continuous increase of multi-view data, multi-view learning has become into a hot research direction in last decades [1], [2], [3], [4], [5], [6]. As an important task, multi-view clustering [7], [8], [9], [10], [11] has been applied to many scientific domains such as natural language processing, computer vision and health informatics. Traditional multi-view clustering assumes that each example of data appears in all views. However, in real-world applications, it is often the case that every view suffers from some data missing, which results in partial multi-view data. For example, in cross-language document grouping, documents have been translated into different languages representing multiple views. However, not all documents are translated into each language. Another example is web image retrieval. Not all web images are associated with text descriptions and the image itself may be inaccessible due to deletion or invalid url. Moreover, in disease diagnosis, there are usually different tests representing multiple views, but it is often the case that some individuals would not like to take all tests. Such incompleteness makes it unaccessible to obtain the clustering results of all examples by applying traditional single-view or multi-view clustering methods on these data directly. Therefore, how to effectively cluster partial multi-view data becomes a practical and important problem.

In recent years, incomplete multi-view clustering has received growing attention, and the existing incomplete multi-view clustering methods are mainly developed in three paradigms. The first paradigm is matrix factorization-based methods. As a pioneering work of matrix factorization-based methods, the approach proposed in [12] learns the representations of both view-specific examples and complete examples simultaneously based on nonnegative matrix factorization, and thus in the learned latent subspace, all examples are homogeneously represented. Such strategy has also been adopted by works in [13], [14], [15]. One limitation of this strategy is that it requires each data appears in all views or only one view. For incomplete multi-view data with more than two views, one very common case is that there exist examples presenting on more than one view but not all views. To handle incomplete multi-view data with arbitrary views, some weighted matrix factorization-based methods have been proposed. The approaches proposed in [16], [17] introduce a diagonal weight matrix for each view, which distinguishes its present samples from missing samples. Furthermore, the works in [18], [19] introduce a weight matrix for each view which distinguishes its certain elements from missing elements. The second paradigm is kernel-based methods. The work in [20] focuses on the two-view data and proposes to construct a full kernel on an incomplete view with the help of another complete view. The approach proposed in [21] predicts the missing rows and columns of kernel matrices by modeling both within-view and between-view relationships

• *Wenzhang Zhuge, Hong Tao, Tingjin Luo, and Chenping Hou are with the College of Liberal Arts and Science, National University of Defense Technology, Changsha 410073, China. E-mail: zgwznudt@yeah.net, {taohong.nudt, hcpnudt}@hotmail.com, tingjinluo@gmail.com.*
• *Ling-Li Zeng is with the College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073, China. E-mail: lingl.zeng@gmail.com.*
• *Dongyun Yi is with the School of Mathematics and Computing Science, Hunan First Normal University, Changsha 410205, China and also with the College of Liberal Arts and Science, National University of Defense Technology, Changsha 410073, China. E-mail: dongyun.yi@gmail.com.*

Manuscript received 1 Mar. 2020; revised 16 Sept. 2020; accepted 27 Sept. 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Chenping Hou and Tingjin Luo.)
Recommended for acceptance by Y. Xia.
Digital Object Identifier no. 10.1109/TKDE.2020.3028422

among kernel values. Based on multiple kernel k-means and mutual kernel completion, the work in [22] jointly performs kernel imputation and common representation learning. The third paradigm is graph-based methods. After filling in the missing entries of graph matrices with the average of the columns, the work in [23] adopts a simple co-training strategy to recover view-specific representations of missing samples and learn common representations. Based on self-representation principle, the approach proposed in [24] integrates the partial graph construction and common representation learning. After filling in the missing entries of each similarity matrix with the average of corresponding certain elements, the work in [25] learns the views weights to combine a common graph matrix by analyzing the relation between perturbation risk bounds and the fusion result. Besides, deep methods such as [26], [27], [28] have achieved improved performance in matrix completion, which show the potential to tackle incomplete multi-view clustering problem.

Although the above-mentioned incomplete multi-view clustering methods have achieved convincing results in some applications, their performance can be further improved due to the following reasons. Matrix factorization-based methods are essentially linear, and thus cannot well reveal the non-linear relation between the data and their representations, which limits their learning ability. For example, there are some 2-dimensional data which form a helical segment in order, and we want to obtain their 1-dimensional representations which maintain the order in the helical segment. As a linear method, matrix factorization cannot deal with such task well. Kernel-based and graph-based methods can explore the non-linear relationships among data, however, most of them involve completion processes, and thus introduce uncertain information, which may lead to a performance degradation especially when missing rate is large. Moreover, the above-mentioned methods share a drawback that they disconnect the processes of representation learning and clustering, and this separate manner prevents two learning processes from negotiating with each other to achieve optimal solution.

In this paper, we propose a new graph-based incomplete multi-view clustering framework, namely Joint Representation Learning and Clustering (JRLC), to address the aforementioned issues. Based on partial similarity matrices, JRLC learns view-specific representation matrices and a common probability label matrix simultaneously. Specifically, JRLC employs representation matrices to take part in the reconstruction of certain elements of the partial similarity matrices, which enables them to capture the view-specific clustering information. Besides, a probability co-regularization term is designed in JRLC to extract explicit and common clustering results for all data points from these representation matrices, which in turn makes the clustering results guide the representation learning on each view. By this way, JRLC connects the representation learning and clustering processes seamlessly, with the aim to achieve better clustering performance. Moreover, we analyze several existing single-view graph-based representation learning methods and explain how JRLC extends them to design new incomplete multi-view clustering methods. To validate the effectiveness of JRLC, we introduce two specific methods, i.e., JRLC with spectral embedding (JRLC-SE) and JRLC via integrating nonnegative embedding and spectral embedding (JRLC-NS). Efficient algorithms with proved convergence are developed to solve the optimization problems of JRLC-SE and JRLC-NS. The performance of the proposed algorithms are verified by systematical experimental results on eight multi-view datasets and in the news topic clustering application. As indicated, the proposed algorithms significantly outperform the compared state-of-the-art incomplete multi-view clustering methods.

This work extends our original conference paper [29] in a substantial way. Compared with the conference paper, its significant improvement can be summarized in the following aspects: 1) We propose JRLC framework for incomplete multi-view clustering, which includes the method proposed in [29] as a special case JRLC-SE. Under the guidance of the framework, several new incomplete multi-view clustering methods can be designed based on existing single-view graph-based methods. 2) By inheriting the merits of both nonnegative embedding and spectral embedding, we introduce another specific method JRLC-NS within the JRLC framework, which achieves comparable or better clustering performance than JRLC-SE in most cases. 3) We propose a general algorithm to solve JRLC framework. Based on the general algorithm, two iterative algorithms are developed to solve the resultant optimization problems of JRLC-SE and JRLC-NS, and their convergence behaviors are theoretically analyzed. 4) We conduct comprehensive experiments and news topic clustering application to demonstrate the effectiveness of the proposed algorithms.

The rest of the paper is organized as follows. Section 2 introduces the problem setting and briefly reviews three related works. The formulation, generalization and optimization of JRLC framework are introduced in Section 3. Two methods JRLC-SE and JRLC-NS are introduced in Section 4. Experimental results are displayed in Section 5, followed by the application to news topic clustering in Section 6. Finally, we conclude this paper in Section 7.

2 Background

Throughout the paper, matrices and vectors are written as boldface uppercase letters and boldface lowercase letters, respectively. For a matrix $\mathbf{M}$, its ith row and (i,j)th element are denoted by $\mathbf{m}_i$ and m_{ij}, respectively. The transpose, the trace and Frobenius norm of matrix $\mathbf{M}$ are denoted by $\mathbf{M}^T$, $tr(\mathbf{M})$ and $||\mathbf{M}||_F$, respectively. $\mathbf{M}^{(v)}$ denotes the vth view representations of $\mathbf{M}$. $\mathcal{C}_M$ and $\mathcal{C}_M^{(v)}$ denote the constraints of $\mathbf{M}$ and $\mathbf{M}^{(v)}$, respectively. The 2-norm of a vector $\mathbf{m}_i$ is denoted by $||\mathbf{m}_i||$. $\mathbf{I}_C$ denotes a $C \times C$-size identity matrix. $\mathbf{1}_d$ denotes a d-dimensional vector and its elements are all 1. We list the notations in Table 1.

2.1 Problem Setting

Given a dataset $\mathbf{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_n] \in \mathbb{R}^{n\times d}$ with n instances sampled from V views, where $\mathbf{x}_i \in \mathbb{R}^{1\times d}$ is the ith instance. Each instance has V representations, i.e., $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(V)}]$, where $\mathbf{x}_i^{(v)} \in \mathbb{R}^{1\times d^{(v)}}$ is the ith sample of the vth view and $d = \sum_{v=1}^V d^{(v)}$. $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}; \ldots; \mathbf{x}_n^{(v)}]$ collects the samples of the vth view and $\mathbf{X} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(V)}]$.

In the incomplete multi-view setting, each $\mathbf{x}_i^{(v)}$ can be missing. Incomplete multi-view clustering aims to cluster

TABLE 1
Summary of Notations

C	Number of clusters
d	Dimension of original data
$d^{(v)}$	Dimension of the vth view
n	Data size
$n^{(v)}$	The vth view present data size
V	Number of views
$\boldsymbol{\beta} = [\beta_1, \ldots, \beta_V]^T \in [0,1]^V$	The view weight vector
$\mathbf{F} = [\mathbf{f}_1; \ldots; \mathbf{f}_n] \in \mathbb{R}^{n \times C}$	The common representations
$\mathbf{F}^{(v)} = [\mathbf{f}_1^{(v)}; \ldots; \mathbf{f}_n^{(v)}] \in \mathbb{R}^{n \times C}$	The vth view representations
$\mathbf{H}^{(v)} = [\mathbf{h}_1^{(v)}; \ldots; \mathbf{h}_n^{(v)}] \in \mathbb{R}^{n \times C}$	The vth view auxiliary matrix
$\mathbf{L}^{(v)} \in \mathbb{R}^{n \times n}$	The vth Laplacian matrix
$\mathbf{L} \in \mathbb{R}^{n \times n}$	The unified Laplacian matrix
$\mathbf{K}^{(v)} \in \mathbb{R}^{n \times n}$	The vth view kernel matrix
$\mathbf{K} \in \mathbb{R}^{n \times n}$	The unified kernel matrix
$\mathbf{O}^{(v)} \in \{0,1\}^{n \times n}$	The vth view diagonal matrix
$\mathbf{S}^{(v)} \in \mathbb{R}^{n \times n}$	The vth view graph matrix
$\mathbf{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_n] \in \mathbb{R}^{n \times d}$	Data matrix
$\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}; \ldots; \mathbf{x}_n^{(v)}] \in \mathbb{R}^{n \times d^{(v)}}$	The vth view data matrix
$\mathbf{Y} = [\mathbf{y}_1; \ldots; \mathbf{y}_n] \in [0,1]^{n \times C}$	Cluster indicator matrix
$\mathbf{R}^{(v)} \in \mathbb{R}^{C \times C}$	The vth view rotation matrix

the n instances into C clusters by integrating all incomplete views. For each view, a diagonal indicator matrix $\mathbf{O}^{(v)} \in \{0,1\}^{n \times n}$ is defined as

$$o_{ii}^{(v)} = \begin{cases} 1, & \text{if } \mathbf{x}_i^{(v)} \text{ appears in the } v\text{-th view} \\ 0, & \text{otherwise} \end{cases}. \tag{1}$$

2.2 Related Works

Incomplete Multi-View Learning via Matrix Factorization. Most existing incomplete multi-view clustering methods [12], [13], [17], [18], [19] are based on matrix factorization. There are mainly two separate steps of these methods: First, they factorize each $\mathbf{X}^{(v)}$ into a common latent feature matrix $\mathbf{F} \in \mathbb{R}^{n \times C}$ by solving the following problem

$$\begin{aligned} &\min_{\mathbf{F}, \mathbf{U}^{(v)}} \sum_{v=1}^{V} [||\boldsymbol{\Theta}^{(v)} \odot (\mathbf{X}^{(v)} - \mathbf{F}\mathbf{U}^{(v)})||_F^2 + \Psi(\mathbf{F}, \mathbf{U}^{(v)})] \\ &s.t.\ \mathbf{F} \in \mathcal{C}_F, \mathbf{U}^{(v)} \in \mathcal{C}_U^{(v)}, (\forall v), \end{aligned} \tag{2}$$

where $\boldsymbol{\Theta}^{(v)} \in \{0,1\}^{n \times d^{(v)}}$ identifies the certain elements of $\mathbf{X}^{(v)}$, $\odot$ denotes element-wise product between two matrices, $\mathbf{U}^{(v)} \in \mathbb{R}^{d^{(v)} \times C}$ is the projection matrix of vth view, $\Psi(\mathbf{F}, \mathbf{U}^{(v)})$ is the regularization term, and $\mathcal{C}_F$ and $\mathcal{C}_U^{(v)}$ are the constraints of $\mathbf{F}$ and $\mathbf{U}^{(v)}$, respectively. These matrix factorization-based methods distinguish from each other by employing different regularization terms and constraints. Second, they apply a post-processing algorithm such as K-means on $\mathbf{F}$ to obtain the clustering indicators.

Incomplete Multiple Kernel K-Means Algorithm With Mutual Kernel Completion (IMKK-MKC). IMKK-MKC is an absent multiple kernel k-means algorithm [22], which integrates imputation and representation learning into a single optimization procedure. Based on the incomplete multiple kernels $\{\mathbf{K}^{(v)}\}_{v=1}^{V}$ where $\mathbf{K}^{(v)} \in \mathbb{R}^{n \times n}$, IMKK-MKC imputes the missing entries of $\{\mathbf{K}^{(v)}\}_{v=1}^{V}$ and learns a common representation matrix $\mathbf{F} \in \mathbb{R}^{n \times C}$ together with a view weight vector $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_V]^T \in \mathbb{R}^V$ simultaneously. The optimization problem can be written as

$$\begin{aligned} &\min_{\boldsymbol{\Gamma}} tr[\mathbf{K}(\mathbf{I} - \mathbf{F}\mathbf{F}^T)] + \frac{\lambda}{2} \sum_{v=1}^{V} ||\mathbf{K}^{(v)} - \sum_{j \neq v} \beta_j \mathbf{K}^{(j)}||_F^2 \\ &s.t.\ \mathbf{F}\mathbf{F}^T = \mathbf{I}_C, \boldsymbol{\beta} \geqslant 0, \boldsymbol{\beta}^T \mathbf{1}_V = 1, \mathbf{K} = \sum_{v=1}^{V} \beta_v^2 \mathbf{K}^{(v)}, \\ &\quad \mathbf{K}^{(v)}(\mathbf{p}^{(v)}, \mathbf{p}^{(v)}) = \mathbf{K}_\Omega^{(v)}, (\forall v), \end{aligned} \tag{3}$$

where $\boldsymbol{\Gamma} = \{\mathbf{F}, \boldsymbol{\beta}, \{\mathbf{K}^{(v)}\}_{v=1}^{V}\}$ collects all uncertain variables, $\mathbf{p}^{(v)}$ is the present sample indices of the vth view, $\mathbf{K}_\Omega^{(v)}$ denotes the kernel sub-matrix computed with vth view present samples, $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the common kernel, and $\lambda > 0$ is a balanced parameter. $\mathbf{K}^{(v)}(\mathbf{p}^{(v)}, \mathbf{p}^{(v)}) = \mathbf{K}_\Omega^{(v)}$ ensures that $\mathbf{K}^{(v)}$ maintains the known entries.

Perturbation-Oriented Incomplete Multi-View Clustering (PIC). PIC is a graph-based incomplete multi-view clustering method [25]. After constructing partial graph matrices $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$ on each view, PIC imputes the missing entries of each $\mathbf{S}^{(v)} \in \mathbb{R}^{n \times n}$ by mean of corresponding certain entries of other graph matrices. Then based on the Laplacian matrices $\{\mathbf{L}^{(v)}\}_{v=1}^{V}$ of completed graph matrices, PIC aims to learn a view weight vector $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_V]^T \in \mathbb{R}^V$ to obtain a consensus Laplacian matrix $\mathbf{L}^* \in \mathbb{R}^{n \times n}$. By analyzing the relation between perturbation risk bounds and the fusion result, the optimization problem can be written as

$$\begin{aligned} &\min_{\boldsymbol{\beta}, \mathbf{L}^*} \sum_{v=1}^{V} ||\mathbf{L}^* \mathbf{F}^{(v)} - \mathbf{F}^{(v)} \boldsymbol{\Sigma}^{(v)}||_F^2 + \lambda \boldsymbol{\beta}^T \mathbf{L}_W \boldsymbol{\beta} \\ &s.t.\ \mathbf{L}^* = \sum_{v=1}^{V} \beta_v \mathbf{L}^{(v)}, \boldsymbol{\beta} \geqslant 0, \boldsymbol{\beta}^T \mathbf{1}_V = 1, \end{aligned} \tag{4}$$

where $\lambda > 0$ is a parameter. $\boldsymbol{\Sigma}^{(v)} \in \mathbb{R}^{C \times C}$ is a diagonal matrix formed by the C largest eigenvalues of $\mathbf{L}^{(v)}$, and the corresponding C eigenvectors are collected by $\mathbf{F}^{(v)} \in \mathbb{R}^{n \times C}$. $\mathbf{L}_W$ is the Laplacian matrix of $\mathbf{W} \in \mathbb{R}^{V \times V}$ and elements of $\mathbf{W}$ measure the similarity between paired of views based on their largest canonical angle. Lastly, PIC applies spectral clustering on $\mathbf{L}^*$ to obtain the clustering results.

3 Proposed Framework

In this section, we first introduce the formulation of JRLC framework. Then we explain the generalization of JRLC. Lastly, we propose a general algorithm for optimization.

3.1 Formulation

To disclose the non-linear structure and utilize the complementary information of multiple views, we construct an undirected weighted graph $\mathbf{S}^{(v)} \in \mathbb{R}^{n \times n}$ on each view according to pairwise similarity of $\{\mathbf{x}_i^{(v)}\}_{i=1}^{n}$. Since some samples can be missing, $s_{ij}^{(v)}$ is calculated by

$$s_{ij}^{(v)} = \begin{cases} f(\mathbf{x}_i^{(v)}, \mathbf{x}_j^{(v)}), & \text{if } o_{ii}^{(v)} o_{jj}^{(v)} = 1 \\ \boldsymbol{\Theta}, & \text{otherwise} \end{cases}, \tag{5}$$

where $f(\mathbf{x}_i^{(v)}, \mathbf{x}_j^{(v)})$ is a similarity calculation method such as [30], [31], and Θ denotes the information of $s_{ij}^{(v)}$ is missing.

According to Eq. (5), $s_{ij}^{(v)}$ can be estimated only if both $\mathbf{x}_i^{(v)}$ and $\mathbf{x}_j^{(v)}$ are present. In our proposed framework, uncertain $\mathbf{\Theta}$ has no effect and can be set with any value.

Based on partial similarity matrices $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$ of multiple views, to jointly perform representation learning and clustering, we learn view-specific representation matrices $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$ and a common probability cluster indicator matrix $\mathbf{Y} = [\mathbf{y}_1; \ldots; \mathbf{y}_n] \in [0,1]^{n\times C}$ simultaneously, where $\mathbf{F}^{(v)} = [\mathbf{f}_1^{(v)}; \ldots; \mathbf{f}_n^{(v)}] \in \mathbb{R}^{n\times C}$ is the vth view representation matrix. To extract the view-specific clustering information, $\mathbf{F}^{(v)}$ is used to reconstruct the known entries of $\mathbf{S}^{(v)}$ with the help of an auxiliary matrix $\mathbf{H}^{(v)} \in \mathbb{R}^{n\times C}$. And to obtain consensus clustering results $\mathbf{Y}$ from $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$, a rotation matrix $\mathbf{R}^{(v)} \in \mathbb{R}^{C\times C}$ is introduced for each $\mathbf{F}^{(v)}$. As a result, the objective of JRLC framework can be concluded as

$$\min_{\in \mathcal{C}} \sum_{v=1}^{V} \mathcal{L}(\mathbf{S}^{(v)}|\mathbf{F}^{(v)}, \mathbf{H}^{(v)}) + \mathcal{R}(\mathbf{Y}, \{\mathbf{F}^{(v)}, \mathbf{R}^{(v)}\}_{v=1}^{V}), \tag{6}$$

where $= \{\{\mathbf{F}^{(v)}, \mathbf{H}^{(v)}, \mathbf{R}^{(v)}\}_{v=1}^{V}, \mathbf{Y}\}$ collects all uncertain variables; $\mathcal{L}(\mathbf{S}^{(v)}|\mathbf{F}^{(v)}, \mathbf{H}^{(v)})$ is reconstruction loss term of the vth view graph matrix $\mathbf{S}^{(v)}$; $\mathcal{R}(\mathbf{Y}, \{\mathbf{F}^{(v)}, \mathbf{R}^{(v)}\}_{v=1}^{V})$ is a co-regularization term. The first term enables representation learning to capture view-specific information, while the second term gives common and explicit clustering results and prompts the clustering results to guide view-specific representation learning.

To obtain the clustering information of the vth view, we utilize $\lambda \mathbf{F}^{(v)}(\mathbf{H}^{(v)})^T$ to reconstruct $\mathbf{S}^{(v)}$, where $\lambda > 0$ is a scaling factor. And to avoid introducing uncertain information, only $n^{(v)} \times n^{(v)}$ certain elements of $\mathbf{S}^{(v)}$ are approximated with the help of $\mathbf{O}^{(v)}$, where $n^{(v)} = \sum_{i=1}^{n} o_{ii}^{(v)}$ is the number of the vth view present samples. Therefore, the objective of $\mathcal{L}(\mathbf{S}^{(v)}|\mathbf{F}^{(v)}, \mathbf{H}^{(v)})$ can be formulated as

$$\min_{\mathbf{F}^{(v)} \in \mathcal{C}_F^{(v)}, \mathbf{H}^{(v)} \in \mathcal{C}_H^{(v)}} ||\mathbf{O}^{(v)}(\mathbf{S}^{(v)} - \lambda \mathbf{F}^{(v)}(\mathbf{H}^{(v)})^T)\mathbf{O}^{(v)}||_F^2, \tag{7}$$

where $\mathcal{C}_F^{(v)}$ and $\mathcal{C}_H^{(v)}$ are the constraints of $\mathbf{F}^{(v)}$ and $\mathbf{H}^{(v)}$, respectively. By utilizing different combinations of $\mathcal{C}_F^{(v)}$ and $\mathcal{C}_H^{(v)}$, the reconstruction of $\mathbf{S}^{(v)}$ can be implemented in a variety of ways. Based on Eq. (7), $\mathbf{f}_i^{(v)}$ and $\mathbf{h}_i^{(v)}$ take part in the reconstruction only if $o_{ii}^{(v)} = 1$, and thus, the constraints should focus on the corresponding rows.

To incorporate clustering and representation learning based on the consensus principle, we learn a common probability label matrix $\mathbf{Y}$ together with representation matrices $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$. To establish reasonable interactions between $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$ and $\mathbf{Y}$, rotation matrices $\{\mathbf{R}^{(v)}\}_{v=1}^{V}$ are employed to help to extract the clustering results from $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$, and C coding vectors $\{\mathbf{t}_{(c)}\}_{c=1}^{C}$ are introduced to identify the C classes. For the cth coding vector $\mathbf{t}_{(c)} \in \{0,1\}^{1\times C}$, only its cth element is equal to 1 and the other ones are 0. The probability co-regularization term $\mathcal{R}(\mathbf{Y}, \{\mathbf{F}^{(v)}, \mathbf{R}^{(v)}\}_{v=1}^{V})$ is formulated as

$$\begin{aligned} \min_{\mathbf{Y}, \{\mathbf{F}^{(v)}, \mathbf{R}^{(v)}\}_V} & \sum_{i=1}^{n} \sum_{c=1}^{C} (y_{ic})^{\gamma} \sum_{v=1}^{V} o_{ii}^{(v)} ||\mathbf{t}_{(c)} - \mathbf{f}_i^{(v)} \mathbf{R}^{(v)}||^2 \\ & s.t.\ \mathbf{Y} \geqslant 0, \mathbf{Y}\mathbf{1}_C = \mathbf{1}_n, (\mathbf{R}^{(v)})^T \mathbf{R}^{(v)} = \mathbf{I}_C, (\forall v), \end{aligned} \tag{8}$$

where $\gamma \geqslant 1$ is an adaptive parameter. From Eq. (8), we can observe that $\mathbf{f}_i^{(v)}$ affects $\mathbf{y}_i$ and $\mathbf{R}^{(v)}$ only if $o_{ii}^{(v)} = 1$. Eq. (8) generates a probability label matrix $\mathbf{Y}$ according to the rotation loss of rows of $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$ to $\{\mathbf{t}_{(c)}\}_{c=1}^{C}$. When $\gamma = 1$, Eq. (8) can be regarded as a variant of classical procrustes average technique [32] which rotates rows of $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$ to form a unified binary label matrix. When $\gamma > 1$, Eq. (8) has a weighting mechanism and each sample is weighted automatically according to clustering certainty $\sum_{c=1}^{C} (y_{ic})^{\gamma}$. The mechanism enables clearly clustered samples to play more important roles in the learning stage.

By combining Eqs. (7) and (8), we propose JRLC framework as the following form

$$\begin{aligned} \min_{\mathbf{T}} \sum_{v=1}^{V} \{ & \sum_{i=1}^{n} \sum_{c=1}^{C} (y_{ic})^{\gamma} o_{ii}^{(v)} ||\mathbf{t}_{(c)} - \mathbf{f}_i^{(v)} \mathbf{R}^{(v)}||^2 \\ & + ||\mathbf{O}^{(v)}(\mathbf{S}^{(v)} - \lambda \mathbf{F}^{(v)}(\mathbf{H}^{(v)})^T)\mathbf{O}^{(v)}||_F^2 \} \\ s.t.\ & \mathbf{F}^{(v)} \in \mathcal{C}_F^{(v)}, \mathbf{H}^{(v)} \in \mathcal{C}_H^{(v)}, (\mathbf{R}^{(v)})^T \mathbf{R}^{(v)} = \mathbf{I}_C, (\forall v), \\ & \mathbf{Y} \geqslant 0, \mathbf{Y}\mathbf{1}_C = \mathbf{1}_n. \end{aligned} \tag{9}$$

The proposed JRLC requires the representation learning to meet demands of both view-specific structure information mining and clustering, with the aim of utilizing both diversity and consensus information of multiple views.

Since the partial similarity matrices $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$ are the inputs of the proposed JRLC framework, their quality will further influence the performance of JRLC. In general, the graph construction way is determined empirically based on types of datasets. Certainly, how to choose a suitable graph is still an open problem.

3.2 Generalization of JRLC Framework

In this subsection, we introduce how JRLC extends existing single-view graph-based representation learning methods to generate new incomplete multi-view clustering methods.

Based on a similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, how to learn a representation matrix $\mathbf{F} \in \mathbb{R}^{n\times C}$ which contains the clustering information has been studied by a number of previous works. If $\mathbf{S}$ is a normalized graph matrix, Normalized Spectral Clustering (NSC) [33] problem can be written as

$$\min_{\mathbf{F}} ||\mathbf{S} - \mathbf{F}\mathbf{F}^T||_F^2,\ s.t.\ \mathbf{F}^T\mathbf{F} = \mathbf{I}_C. \tag{10}$$

To inherit merit from nonnegative constraint, the optimization problem of symmetric nonnegative matrix factorization (SymNMF) in [34] is

$$\min_{\mathbf{F}} ||\mathbf{S} - \mathbf{F}\mathbf{F}^T||_F^2,\ s.t.\ \mathbf{F} \geqslant 0. \tag{11}$$

Furthermore, left-stochastic decomposition (LSD) in [35] requires $\mathbf{F}$ to be a probability matrix, and the problem is

$$\min_{\mathbf{F}} ||\mathbf{S} - \lambda \mathbf{F}\mathbf{F}^T||_F^2,\ s.t.\ \mathbf{F} \geqslant 0, \mathbf{F}\mathbf{1}_C = \mathbf{1}_n, \tag{12}$$

where $\lambda > 0$ is a scaling parameter. To obtain merits from (10) and (11), $\mathbf{F}$ satisfies both orthogonal and nonnegative constraints in [36], and the optimization problem of orthogonal nonnegative matrix factorization (ONMF) is

$$\min_{\mathbf{F}} ||\mathbf{S} - \mathbf{F}\mathbf{F}^T||_F^2,\ s.t.\ \mathbf{F} \geqslant 0, \mathbf{F}^T\mathbf{F} = \mathbf{I}_C. \tag{13}$$

TABLE 2
Summary of Some Previous Single-View Representation Learning Methods and the Corresponding Extended Versions Within JRLC Framework

Method	$\{\mathcal{C}_F^{(v)}, \mathcal{C}_H^{(v)}\}_{v=1}^V$
NSC [33]	$(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{F}^{(v)} = \mathbf{I}_C$, $\mathbf{H}^{(v)} = \mathbf{F}^{(v)}$
SymNMF [34]	$\mathbf{F}^{(v)} \geqslant 0$, $\mathbf{H}^{(v)} = \mathbf{F}^{(v)}$
LSD [35]	$\mathbf{F}^{(v)} \geqslant 0$, $\mathbf{F}^{(v)}\mathbf{1}_C = \mathbf{1}_n$, $\mathbf{H}^{(v)} = \mathbf{F}^{(v)}$
ONMF [36]	$(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{F}^{(v)} = \mathbf{I}_C$, $\mathbf{F}^{(v)} \geqslant 0$, $\mathbf{H}^{(v)} = \mathbf{F}^{(v)}$
ONE [37]	$\mathbf{F}^{(v)} \geqslant 0$, $(\mathbf{H}^{(v)})^T\mathbf{O}^{(v)}\mathbf{H}^{(v)} = \mathbf{I}_C$

In [37], [38], orthogonal and nonnegative embedding (ONE) proposes another way to combine (10) and (11), and the optimization problem is

$$\min_{\mathbf{F},\mathbf{H}} ||\mathbf{S} - \mathbf{F}\mathbf{H}^T||_F^2, \;\; s.t. \; \mathbf{F} \geqslant 0, \mathbf{H}^T\mathbf{H} = \mathbf{I}_C, \tag{14}$$

where $\mathbf{H} \in \mathbb{R}^{n \times C}$ is an introduced variable matrix.

By analyzing the aforementioned single-view methods, we propose to summarize them into the following objective

$$\min_{\mathbf{F} \in \mathcal{C}_F, \mathbf{H} \in \mathcal{C}_H} ||\mathbf{S} - \lambda \mathbf{F}\mathbf{H}^T||_F^2, \tag{15}$$

where $\mathcal{C}_F$ and $\mathcal{C}_H$ are the constraints of $\mathbf{F}$ and $\mathbf{H}$.

By adjusting Eq. (15) to reconstruct the certain elements of partial similarity matrices $\{\mathbf{S}^{(v)}\}_{v=1}^V$, we obtain Eq. (7). Therefore, these single-view methods can be extended by JRLC to generate new incomplete multi-view clustering methods. Since methods within JRLC framework distinguish from each other by adopting different $\{\mathcal{C}_F^{(v)}, \mathcal{C}_H^{(v)}\}_{v=1}^V$, we summarize them in Table 2.

The differences among these single-view representation learning methods are inherited by corresponding methods within JRLC framework. These single-view methods share the common objective (15) and have different constraints. NSC uses orthogonal constraint, which tries to reconstruct the similarity matrix by a block-diagonal matrix. SymNMF uses nonnegative constraint, which offers interpretability that entries in the representation matrix directly correspond to relationship between data points and clusters. Based on SymNMF, LSD further requires the representation matrix to be a cluster probability matrix, which makes it reflect the final clustering result in a more accurate way. By introducing the orthogonal and nonnegative constraints simultaneously, ONMF can be regarded as a combination of NSC and SymNMF, and thus the reconstruct graph matrix has a more clear structure. As a relaxed version of ONMF, ONE inherits its good property for representation learning and has low complexity, and the introduced variable matrix makes the reconstruction of the similarity matrix have more flexibility. In multiple graph learning [38], the increased flexibility may improve the clustering performance. Besides, by combining different constraints, some new representation learning methods can be generated.

3.3 Optimization

The problem (9) is not convex over all four groups of variables $\{\mathbf{F}^{(v)}\}_{v=1}^V, \{\mathbf{H}^{(v)}\}_{v=1}^V, \{\mathbf{R}^{(v)}\}_{v=1}^V$ and $\mathbf{Y}$ simultaneously. The problems of specific methods within JRLC framework can be solved by an alternative and iterative minimization strategy which updates one group of variables while fixes others. The updating rules of $\{\mathbf{R}^{(v)}\}_{v=1}^V$ and $\mathbf{Y}$ are standard for all specific methods, and the updating rules w.r.t $\{\mathbf{F}^{(v)}\}_{v=1}^V$ and $\{\mathbf{H}^{(v)}\}_{v=1}^V$ vary according to $\{\mathcal{C}_F^{(v)}\}_{v=1}^V$ and $\{\mathcal{C}_H^{(v)}\}_{v=1}^V$, respectively.

Update $\{\mathbf{R}^{(v)}\}_{v=1}^V$: With $\mathbf{Y}$ and $\{\mathbf{F}^{(v)}, \mathbf{H}^{(v)}\}_{v=1}^V$ fixed, the relations of multiple views are decoupled, and each $\mathbf{R}^{(v)}$ can be updated by solving the following problem

$$\min_{(\mathbf{R}^{(v)})^T\mathbf{R}^{(v)} = \mathbf{I}_C} \sum_{i=1}^n o_{ii}^{(v)} \sum_{c=1}^C (y_{ic})^\gamma ||\mathbf{t}_{(c)} - \mathbf{f}_i^{(v)}\mathbf{R}^{(v)}||^2. \tag{16}$$

By removing constant terms, the minimum problem (16) is equivalent to the following problem

$$\max_{(\mathbf{R}^{(v)})^T\mathbf{R}^{(v)} = \mathbf{I}_C} tr[(\mathbf{R}^{(v)})^T(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{G}], \tag{17}$$

where $\mathbf{G} \in \mathbb{R}^{n \times C}$ and its ith row $\mathbf{g}_i = \sum_{c=1}^C (y_{ic})^\gamma \mathbf{t}_{(c)}$. Since $n^{(v)}$ examples appears in the vth view, the corresponding $n^{(v)}$ rows of $\mathbf{F}^{(v)}$ and $\mathbf{G}$ are collected by $\mathbf{F}^{\Omega(v)} \in \mathbb{R}^{n^{(v)} \times C}$ and $\mathbf{G}^{\Omega(v)} \in \mathbb{R}_+^{n^{(v)} \times C}$, respectively. It can be checked that $(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{G} = (\mathbf{F}^{\Omega(v)})^T\mathbf{G}^{\Omega(v)}$. To update $\mathbf{R}^{(v)}$, we introduce the following proposition.

Proposition 1. *Suppose the SVD of matrix* $(\mathbf{F}^{\Omega(v)})^T\mathbf{G}^{\Omega(v)}$ *is* $(\mathbf{F}^{\Omega(v)})^T\mathbf{G}^{\Omega(v)} = \mathbf{U}_R^{(v)}\mathbf{\Sigma}_R^{(v)}(\mathbf{V}_R^{(v)})^T$, *then the optimal* $\mathbf{R}^{(v)}$ *to the problem (17) is*

$$\mathbf{R}^{(v)} = \mathbf{U}_R^{(v)}(\mathbf{V}_R^{(v)})^T. \tag{18}$$

The detailed proofs of all propositions of this paper can be found in the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2020.3028422.

Update $\mathbf{Y}$*:* With $\{\mathbf{F}^{(v)}, \mathbf{H}^{(v)}, \mathbf{R}^{(v)}\}_{v=1}^V$ fixed, $\mathbf{Y}$ can be updated by solving the following n problems simultaneously and independently

$$\min_{\mathbf{y}_i \geqslant \mathbf{0}, \mathbf{y}_i\mathbf{1}_C = 1} \sum_{c=1}^C (y_{ic})^\gamma \sum_{v=1}^V o_{ii}^{(v)} ||\mathbf{t}_{(c)} - \mathbf{f}_i^{(v)}\mathbf{R}^{(v)}||^2. \tag{19}$$

Denote $q_{ic} = \sum_{v=1}^V o_{ii}^{(v)} ||\mathbf{t}_{(c)} - \mathbf{f}_i^{(v)}\mathbf{R}^{(v)}||^2$, which is the (i, c)th element of matrix $\mathbf{Q} \in \mathbb{R}^{n \times C}$. When $\gamma = 1$, the optimal solution of (19) is

$$y_{ij} = < j = \arg\min_{c \in [1, C]} q_{ic} >, \tag{20}$$

where function $< \cdot >$ is equal to 1 if the argument is true or 0 otherwise. When $\gamma > 1$, the Lagrangian function of the problem (19) is $\mathcal{L}_\mu = \sum_{c=1}^C (y_{ic})^\gamma q_{ic} - \mu(\sum_{c=1}^C y_{ic} - 1)$, where μ is the Lagrange multiplier. Setting the derivative of $\mathcal{L}_\mu$ w.r.t y_{ic} to zero and combining the constraint $\sum_{c=1}^C y_{ic} = 1$, we arrive at the closed-form solution of the problem (19)

$$y_{ic} = \frac{(q_{ic})^{\frac{1}{1-\gamma}}}{\sum_{c=1}^C (q_{ic})^{\frac{1}{1-\gamma}}}. \tag{21}$$

Update $\{\mathbf{F}^{(v)}, \mathbf{H}^{(v)}\}_{v=1}^{V}$: With $\{\mathbf{R}^{(v)}\}_{v=1}^{V}$ and $\mathbf{Y}$ fixed, after removing constant terms, the problem (9) can be decoupled into the following V problems

$$\begin{aligned} \min\ & \mathcal{J}(\mathbf{F}^{(v)}, \mathbf{H}^{(v)}) \\ =& tr[(\mathbf{F}^{(v)})^T \big(\mathbf{O}^{(v)}\mathbf{D}\mathbf{O}^{(v)}\mathbf{F}^{(v)} - 2\mathbf{O}^{(v)}\mathbf{G}(\mathbf{R}^{(v)})^T\big)] \\ &+ \lambda^2 tr[(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{F}^{(v)}(\mathbf{H}^{(v)})^T\mathbf{O}^{(v)}\mathbf{H}^{(v)}] \\ &- 2\lambda tr[(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}(\mathbf{S}^{(v)})^T\mathbf{O}^{(v)}\mathbf{H}^{(v)}] \\ s.t.\ & \mathbf{F}^{(v)} \in \mathcal{C}_F^{(v)}, \mathbf{H}^{(v)} \in \mathcal{C}_H^{(v)}, \end{aligned} \tag{22}$$

where $\mathbf{D} \in \mathbb{R}_+^{n\times n}$ is a diagonal matrix with (i,i)th element $d_{ii} = \sum_{c=1}^{C}(y_{ic})^\gamma$. By adding different constraints $\mathcal{C}_F^{(v)}$ and $\mathcal{C}_H^{(v)}$, methods within JRLC framework apply different ways to solve the problem (22).

Since the proposed (9) is solved in an alternative way, we initialize $\mathbf{R}^{(v)} = \mathbf{I}_C$ and $\mathbf{Y}$ such that $y_{ic} = 1/C$. $\mathbf{F}^{(v)}$ and $\mathbf{H}^{(v)}$ are initialized according to the their explicit constraints. At last, we resort to a decision function to assign the single class label for each $\mathbf{y}_i$

$$y_{ij} = < j = \arg\max_{c\in[1,C]} y_{ic} > . \tag{23}$$

In summary, the general procedure of JRLC framework is listed in Algorithm 1.

Algorithm 1. Optimization of JRLC Framework

Input: Partial similarity matrices $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$, indicator matrices $\{\mathbf{O}^{(v)}\}_{v=1}^{V}$, cluster number C, parameters λ and γ.
Initialization: $\mathbf{Y}$ with $y_{ic} = 1/C$, $\mathbf{R}^{(v)} = \mathbf{I}_C$, $\mathbf{F}^{(v)}$, $\mathbf{H}^{(v)}$.
while *not converged* **do**
1: Update $\{\mathbf{R}^{(v)}\}_{v=1}^{V}$ with Eq. (18).
2: Update $\{\mathbf{y}_i\}_{i=1}^{n}$ with Eqs. (20) or (21).
3: Update $\{\mathbf{F}^{(v)}, \mathbf{H}^{(v)}\}_{v=1}^{V}$ by solving (22).
end while
Output: The discrete indicator matrix $\mathbf{Y}$ with Eq. (23).

---

4 Method and Algorithm

To illustrate the ways of solving methods within JRLC framework, we introduce two specific methods with corresponding algorithms in this section.

4.1 JRLC With Spectral Embedding

The first method based on spectral embedding is named as JRLC-SE, which is the extended version of NSC. We choose JRLC-SE because NSC is the most classical method among single-view representation learning methods introduced in Section 3.2, and the corresponding constraints are

$$\text{JRLC-SE}: (\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{F}^{(v)} = \mathbf{I}_C, \mathbf{H}^{(v)} = \mathbf{F}^{(v)}, (\forall v). \tag{24}$$

Considering the constraints and removing the constant terms, the objective of reconstruction loss term $\mathcal{L}(\mathbf{S}^{(v)}|\mathbf{F}^{(v)}, \mathbf{H}^{(v)})$ of JRLC-SE can be replaced by

$$\min_{(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{F}^{(v)}=\mathbf{I}_C} -2\lambda tr[(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{S}^{(v)}\mathbf{O}^{(v)}\mathbf{F}^{(v)}]. \tag{25}$$

If $\mathbf{S}^{(v)}$ satisfies $\mathbf{O}^{(v)}\mathbf{S}^{(v)}\mathbf{O}^{(v)}\mathbf{1}_n = \mathbf{O}^{(v)}(\mathbf{S}^{(v)})^T\mathbf{O}^{(v)}\mathbf{1}_n = \mathbf{O}^{(v)}\mathbf{1}_n$, we can use $2\lambda tr[(\mathbf{F}^{(v)})^T\mathbf{L}^{(v)}\mathbf{F}^{(v)}]$ to replace the objective of (25), where $\mathbf{L}^{(v)}$ is the Laplacian of $\mathbf{O}^{(v)}\mathbf{S}^{(v)}\mathbf{O}^{(v)}$. Thus, $\mathcal{L}(\mathbf{S}^{(v)}|\mathbf{F}^{(v)}, \mathbf{H}^{(v)})$ of JRLC-SE can be replaced by

$$\min_{(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{F}^{(v)}=\mathbf{I}_C} \lambda \sum_{i,j=1}^{n} o_{ii}^{(v)} s_{ij}^{(v)} o_{jj}^{(v)} ||\mathbf{f}_i^{(v)} - \mathbf{f}_j^{(v)}||^2. \tag{26}$$

Therefore, JRLC-SE can be regarded as SRLC method proposed in our conference paper [29] under certain conditions.

The algorithms for specific methods within JRLC framework distinguish from each other by solving problem (22) with different constraints. For JRLC-SE, since $\mathbf{H}^{(v)} = \mathbf{F}^{(v)}$, we replace $\mathbf{H}^{(v)}$ with $\mathbf{F}^{(v)}$.

Update $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$: By analyzing Eqs. (22) and (24), it can be checked that only $\mathbf{F}^{\Omega(v)}$ needs to be optimized. Let $\mathbf{S}^{\Omega(v)} \in \mathbb{R}^{n^{(v)}\times n^{(v)}}$ collect the $n^{(v)} \times n^{(v)}$ certain elements of $\mathbf{S}^{(v)}$. And $\mathbf{D}^{\Omega(v)} \in \mathbb{R}_+^{n^{(v)}\times n^{(v)}}$ collect the corresponding $n^{(v)} \times n^{(v)}$ elements of $\mathbf{D}$. By removing the constant terms, the problem (22) with constraints (24) is equivalent to

$$\begin{aligned} \min_{(\mathbf{F}^{\Omega(v)})^T\mathbf{F}^{\Omega(v)}=\mathbf{I}_C} & tr[(\mathbf{F}^{\Omega(v)})^T(\mathbf{D}^{\Omega(v)} - 2\lambda\mathbf{S}^{\Omega(v)})\mathbf{F}^{\Omega(v)}] \\ & -2tr[(\mathbf{F}^{\Omega(v)})^T\mathbf{G}^{\Omega(v)}(\mathbf{R}^{(v)})^T]. \end{aligned} \tag{27}$$

The minimization problem (27) is equivalent to the following maximization problem

$$\max_{(\mathbf{F}^{\Omega(v)})^T\mathbf{F}^{\Omega(v)}=\mathbf{I}_C} tr[(\mathbf{F}^{\Omega(v)})^T(\mathbf{A}^{(v)}\mathbf{F}^{\Omega(v)} + \mathbf{B}^{(v)})], \tag{28}$$

where $\mathbf{A}^{(v)} = \alpha^{(v)}\mathbf{I}_{n^{(v)}} - \mathbf{D}^{\Omega(v)} + 2\lambda\mathbf{S}^{\Omega(v)}$ and $\mathbf{B}^{(v)} = 2\mathbf{G}^{\Omega(v)}(\mathbf{R}^{(v)})^T$. $\alpha^{(v)}$ is an arbitrary constant which ensures that $\mathbf{A}^{(v)}$ is a positive definite matrix. Motivated by [39], the problem (27) can be solved by the following iterative and alternative strategy

1) Update $\mathbf{C}^{(v)} = 2\mathbf{A}^{(v)}\mathbf{F}^{\Omega(v)} + \mathbf{B}^{(v)} \in \mathbb{R}^{n^{(v)}\times C}$;
2) Calculate $\mathbf{F}^{\Omega(v)}$ by solving the following problem

$$\max_{(\mathbf{F}^{\Omega(v)})^T\mathbf{F}^{\Omega(v)}=\mathbf{I}_C} tr[(\mathbf{F}^{\Omega(v)})^T\mathbf{C}^{(v)}]. \tag{29}$$

According to Proposition 1, supposing that the compact SVD of $\mathbf{C}^{(v)} = \mathbf{U}_F^{(v)}\mathbf{\Sigma}_F^{(v)}(\mathbf{V}_F^{(v)})^T$, then the optimal $\mathbf{F}^{\Omega(v)}$ of problem (12) is $\mathbf{F}^{\Omega(v)} = \mathbf{U}_F^{(v)}(\mathbf{V}_F^{(v)})^T$.

To analyze the convergence behavior of above two steps, we introduce the following proposition.

Proposition 2. *The above two alternative and iterative steps will monotonically increase the objective of the problem (28) in each iteration until it converges to a stationary point of (28).*

The procedure of JRLC-SE is listed in Algorithm 2.

Convergence Behavior. For the convergence behavior of Algorithm 2, we have the following proposition.

Proposition 3. *The iterative updating rules in Algorithm 2 will monotonically decrease the objective of the optimization problem of JRLC-SE until convergence, which makes the solution be a stationary point of the problem of JRLC-SE when* $\gamma > 1$.

Computational Complexity. In the following, we analyze the computational complexity of Algorithm 2. In each iteration, the computational complexity to update $\mathbf{F}^{(v)}$ is $O(\tau(n^{(v)}kC + n^{(v)}C^2 + C^3))$, where τ is the iteration times of inner loop to solve the problem (27) and k is the number of neighbors of partial similarity matrices; the computational complexity to update $\mathbf{R}^{(v)}$ is $O(n^{(v)}C^2 + C^3)$; the computational complexity to update $\mathbf{Y}$ is $O(\sum_{v=1}^{V} n^{(v)}C^2)$. In general, $C \ll n^{(v)}$. The overall computational complexity is $O(T\tau\sum_{v=1}^{V} n^{(v)}(k+C)C)$, where T is the number of iterations of Algorithm 2.

Algorithm 2. Algorithm to Solve JRLC-SE

Input: Partial similarity matrices $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$, indicator matrices $\{\mathbf{O}^{(v)}\}_{v=1}^{V}$, cluster number C, parameters λ and γ.
Initialization: $\mathbf{Y}$ with $y_{ic} = 1/C$, $\mathbf{R}^{(v)} = \mathbf{I}_C$, $\mathbf{F}^{(v)}$ such that $(\mathbf{F}^{(v)})^T\mathbf{O}^{(v)}\mathbf{F}^{(v)} = \mathbf{I}_C$.
while *not converged* **do**
1: Update $\{\mathbf{F}^{\Omega(v)}\}_{v=1}^{V}$ of $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$ by solving (27).
2: Update $\{\mathbf{R}^{(v)}\}_{v=1}^{V}$ with Eq. (18).
3: Update $\{\mathbf{y}_i\}_{i=1}^{n}$ with Eqs. (20) or (21).
end while
Output: The discrete indicator matrix $\mathbf{Y}$ with Eq. (23).

4.2 JRLC via Integrating Nonnegative Embedding and Spectral Embedding

The second method is named as JRLC-NS, which inherit the merits of both nonnegative embedding and spectral embedding. JRLC-NS is the extended version of ONE. We choose JRLC-NE because ONE is the most advanced methods among these methods introduced in Section 3.2. The identified constraints of JRLC-NS are

$$\text{JRLC-NS}: \mathbf{F}^{(v)} \geqslant 0, (\mathbf{H}^{(v)})^T\mathbf{O}^{(v)}\mathbf{H}^{(v)} = \mathbf{I}_C, (\forall v). \tag{30}$$

With the help of scaling factor λ, $\lambda\mathbf{f}_i^{(v)}\mathbf{R}^{(v)}$ can be more comparable with clustering indicator vectors $\mathbf{t}_{(c)}$.

Considering the constraints of JRLC-NS, we solve the problem (22) with constraints (30) by updating $\mathbf{F}^{(v)}$ and $\mathbf{H}^{(v)}$ in an alternative way.

Update $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$*:* With $\mathbf{H}^{(v)}$ fixed, it is easy to check that only $\mathbf{F}^{\Omega(v)}$ of $\mathbf{F}^{(v)}$ needs to be optimized. The problem (22) with constraints (30) is equivalent to

$$\begin{aligned}\min_{\mathbf{F}^{\Omega(v)}\geqslant 0} & \ tr[(\mathbf{F}^{\Omega(v)})^T(\mathbf{D}^{\Omega(v)} + \lambda^2\mathbf{I}_{n^{(v)}})\mathbf{F}^{\Omega(v)}] \\ & -2tr[(\mathbf{F}^{\Omega(v)})^T(\mathbf{G}^{\Omega(v)}(\mathbf{R}^{(v)})^T + \lambda\mathbf{S}^{\Omega(v)}\mathbf{H}^{\Omega(v)})],\end{aligned} \tag{31}$$

where $\mathbf{H}^{\Omega(v)} \in \mathbb{R}^{n^{(v)}\times C}$ collects $n^{(v)}$ rows of $\mathbf{H}^{(v)}$ corresponding to the vth view present samples. Denote $\mathbf{M}^{(v)} = \mathbf{D}^{\Omega(v)} + \lambda^2\mathbf{I}_{n^{(v)}} \in \mathbb{R}_+^{n^{(v)}\times n^{(v)}}$ and $\mathbf{E}^{(v)} = \mathbf{G}^{\Omega(v)}(\mathbf{R}^{(v)})^T + \lambda\mathbf{S}^{\Omega(v)}\mathbf{H}^{\Omega(v)} \in \mathbb{R}^{n^{(v)}\times C}$. Since $\mathbf{M}^{(v)}$ is a diagonal matrix, the nonnegative quadratic programming problem (31) can be further decoupled into the following $n^{(v)} \times C$ subproblems $(i = 1,\ldots,n^{(v)}; c = 1,\ldots,C)$

$$\min_{f_{ic}^{\Omega(v)}\geqslant 0} m_{ii}^{(v)}(f_{ic}^{\Omega(v)})^2 - 2e_{ic}^{(v)}f_{ic}^{\Omega(v)}. \tag{32}$$

Note that $m_{ii}^{(v)}\geqslant 0$, and considering the nonnegative constraint, the optimal solution of the problem (32) is

$$f_{ic}^{\Omega(v)} = \max\left(0, \frac{e_{ic}^{(v)}}{m_{ii}^{(v)}}\right). \tag{33}$$

Update $\{\mathbf{H}^{(v)}\}_{v=1}^{V}$*:* With $\mathbf{F}^{(v)}$ fixed, by considering the effect of removing $\mathbf{O}^{(v)}$ and removing constant terms, the problem (22) with constraints (30) is equivalent to the following problem

$$\max_{(\mathbf{H}^{\Omega(v)})^T\mathbf{H}^{\Omega(v)}=\mathbf{I}_C} tr[(\mathbf{H}^{\Omega(v)})^T(\mathbf{S}^{\Omega(v)})^T\mathbf{F}^{\Omega(v)}]. \tag{34}$$

The problem (34) is similar to the problem (29). According to Proposition 1, supposing the compact SVD of $(\mathbf{S}^{\Omega(v)})^T\mathbf{F}^{\Omega(v)} = \mathbf{U}_H^{(v)}\mathbf{\Sigma}_H^{(v)}(\mathbf{V}_H^{(v)})^T$, then the solution $\mathbf{H}^{\Omega(v)}$ of (34) is

$$\mathbf{H}^{\Omega(v)} = \mathbf{U}_H^{(v)}(\mathbf{V}_H^{(v)})^T. \tag{35}$$

The procedure of JRLC-NS is listed in Algorithm 3.

Algorithm 3. Algorithm to Solve JRLC-NS

Input: Partial similarity matrices $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$, indicator matrices $\{\mathbf{O}^{(v)}\}_{v=1}^{V}$, cluster number C, parameters λ and γ.
Initialization: $\mathbf{Y}$ with $y_{ic} = 1/C$, $\mathbf{R}^{(v)} = \mathbf{I}_C$, $\mathbf{H}^{(v)}$ such that $(\mathbf{H}^{(v)})^T\mathbf{O}^{(v)}\mathbf{H}^{(v)} = \mathbf{I}_C$.
while *not converged* **do**
1: Update $\{\mathbf{F}^{\Omega(v)}\}_{v=1}^{V}$ of $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$ with Eq. (33).
2: Update $\{\mathbf{H}^{\Omega(v)}\}_{v=1}^{V}$ of $\{\mathbf{H}^{(v)}\}_{v=1}^{V}$ with Eq. (35).
3: Update $\{\mathbf{R}^{(v)}\}_{v=1}^{V}$ with Eq. (18).
4: Update $\{\mathbf{y}_i\}_{i=1}^{n}$ with Eqs. (20) or (21).
end while
Output: The discrete indicator matrix $\mathbf{Y}$ with Eq. (23).

Convergency Behavior. For the convergence behavior of Algorithm 3, we have the following proposition.

Proposition 4. *The iterative updating rules in Algorithm 3 will monotonically decrease the objective of the optimization problem of JRLC-NS until convergence, which makes the solution be a stationary point of the problem of JRLC-NS when* $\gamma > 1$.

Computational Complexity. We analyze the computational complexity of Algorithm 3. In each iteration, the computational complexity to update $\mathbf{F}^{(v)}$ is $O(n^{(v)}kC + n^{(v)}C^2)$; the computational complexity to update $\mathbf{H}^{(v)}$ is $O(n^{(v)}kC + n^{(v)}C^2 + C^3)$; the computational complexity to update $\mathbf{R}^{(v)}$ and $\mathbf{Y}$ are $O(n^{(v)}C^2 + C^3)$ and $O(\sum_{v=1}^{V} n^{(v)}C^2)$, respectively. Since $C \ll n^{(v)}$, the overall computational complexity is $O(T\sum_{v=1}^{V} n^{(v)}(k+C)C)$, where T is number of iterations of Algorithm 3. Compared with Algorithm 2, there is no inner loop in Algorithm 3, which further improve its efficiency.

5 Experiments

In this section, we conduct experiments to evaluate the performance of the proposed algorithms. First, we evaluate the effectiveness of JRLC framework by comparing JRLC-SE and JRLC-NS with some baselines. Second, we present experimental results about convergence and runtime. Third, we validate the effectiveness of the integration of representation learning and clustering. Finally, the impacts of hyperparameters are studied.

5.1 Dataset Description

The experiments are conducted on 8 datasets, i.e., MSRC-v1,[1] Caltech7,[2] Yale,[3] ORL,[4] Dights,[5] Ionosphere,[6] Forest,[7] WebKB.[8] The detailed descriptions of these datasets are listed as follows.

1) MSRC-v1: MSRC-v1 consists of 240 images and is divided into 8 categories. Following [40], 7 widely used classes are selected, i.e., *tree*, *building*, *airplane*, *cow*, *face*, *car*, *bicycle*, and each class has 30 images. Six features are extracted, i.e., 1302 CENTRIST, 256 Local Binary Pattern (LBP), 48 Color Moment (CMT), 100 Histogram of Oriented Gradient (HOG), 200 SIFT and 512 GIST.
2) Caltech7: Caltech101 includes 8677 objective images belonging to 101 classes. Following [41], we select 7 categories, including *Dolla-Bill*, *Faces*, *Garfield*, *Motorbikes*, *Snoopy*, *Stop-Sign* and *Windsor-Chair*. The selected subset with 441 images is named as Caltech7. For each image, the same six kinds of features with MSRC-v1 are extracted.
3) Digits: Digits is composed of 2,000 data points for 0 to 9 ten digit classes, and each class has 200 data points. Six public features are available, i.e., 76 Fourier coefficients of the character shapes (FOU), 216 profile correlations (FAC), 64 Karhunen-love coefficients (KAR), 240 pixel averages in 2×3 windows (PIX), 47 Zernike moment (ZER) and 6 morphological (MOR) features.
4) Yale: Yale contains 165 face images belonging to 15 persons, and each person has 11 images. For each image, we extract 512 GIST, 256 LBP and 168 Pyramid Histogram of Oriented Gradients (PHOG).
5) ORL: ORL is composed of 400 face images belonging to 40 persons, and each person has 10 images. For each image, we extract the same three kinds of features with Yale.
6) Ionosphere: Ionosphere consists of a phased array of 16 high-frequency antennas and result in observations with 34 features. It includes 351 instances in total which are classified into 225 ’Good’ instances and 126 ’Bad’ instances. Following [42], the second view is generated by reducing the dimensionality from 34 to 25 with PCA.
7) Forest: Forest is composed of multi-temporal remote sensing data of a forested area [43]. It includes 523 instances belonging to 4 forest types, i.e., ’Sugi’ forest, ’Hinoki’ forest, ’Mixed deciduous’ forest and ’Other’ non-forest land. Each instance has 9 features about ASTER image bands and 18 features about predicted spectral values.
8) WebKB: WebKB consists of 1051 web documents [44] classified into 2 classes: 230 Course pages and 821 Non-Course pages. Each page has two representations: Fulltext with 2949 features describes the textual content on the web page, while Inlinks with 334 features records the anchor text on the hyperlinks pointing to the pages.

1. https://www.microsoft.com/en-us/research/project/
2. http://www.vision.caltech.edu/Image_Datasets/Caltech101/
3. http://vision.ucsd.edu/content/yale-face-database.
4. http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html.
5. https://archive.ics.uci.edu/ml/datasets/Multiple+Features
6. http://archive.ics.uci.edu/ml/datasets/Ionosphere.
7. https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping
8. http://www.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb/.

5.2 Experimental Setup

5.2.1 Dataset Processing

Since all these datasets are originally complete, to simulate the incomplete multi-view setting, some view samples of each data point are randomly removed. Concretely, for each $\mathbf{x}_i^{(v)}$, there is a probability to remove it. The probability can also be regarded as the incomplete example ratio (IER) of the dataset. In the experiments, we tune IER form 10 to 50 percent with a step 10 percent. And for each data point $\mathbf{x}_i$, it is ensured that there is at least one $\mathbf{x}_i^{(v)}$ remaining.

5.2.2 Baselines and Experimental Environment

In the experiments, we compare the proposed JRLC-SE and JRLC-NS with several state-of-the-art methods: Multiple Incomplete views Clustering (MIC) [16], Multi-view Learning with Incomplete Views (MVL-IV) [18], Incomplete Multi-modality Grouping (IMG) [13], Doubly Aligned Incomplete Multi-view Clustering (DAIMC) [17], Incomplete Multiple Kernel K-means Algorithm with Mutual Kernel Completion [22] (IMKK-MKC) and Perturbation-oriented Incomplete multi-view Clustering (PIC) [25]. Since the original IMG can only deal with two incomplete views, we extend it based on Eq. (3), and the extended version can be applied on data with any number of incomplete views. Besides, we compare our proposed methods with Matrix Completion by Deep Matrix Factorization (DMF) [28]. By apply DMF on the concatenated feature matrix of all views, we can obtain the completed concatenated feature matrix and a common representation matrix, and they corresponds to two baselines called DMF-F and DMF-R, respectively. Since all these baselines need post-processing to extract the clustering indicators, K-means is apply on the common representation matrix to obtain the clustering the results. We conduct experiments by MATLAB R2017a on a work station with Intel(R) Xecon(R) CPU E3-1245 v3(3.4 GHz), 32.0 GB RAM memory, and Windows 10 operating system.

5.2.3 Parameter Determination

In the experiments, all hyper-parameters are determined by grid-search, and the clustering results of using the best tuned parameters are recorded. For baselines, we download the source codes from the authors’ websites and the searching ranges of the parameters are determined according to the corresponding papers. For DMF, the number of nodes in input layer, hidden layer, and output layer are set as $[d, 10C, C]$. For the proposed JRLC-SE and JRLC-NS, the partial similarity matrices are constructed by [30], the adaptive parameter γ is tuned from 1.1 to 2.5 with a step 0.2, and the scaling factor λ is tuned from in the range of 10 of -2 power to 2 power with a step 0.5. The neighbor number k of partial similarity matrix is fixed as 5 in this section. For all compared algorithms which adopt iterative optimization strategy, the stop criteria is

$$\frac{J(t-1) - J(t)}{J(t-1)} < 10^{-5}, \tag{36}$$

TABLE 3
ACC (%) Comparisons on 8 Datasets With Different IERs

Data set	IER	MIC	MVL-IV	IMG	DAIMC	DMF-F	DMF-R	IMKK-MKC	PIC	JRLC-SE	JRLC-NS
MSRC-v1	10%	71.7(3.9)•	85.1(2.9)•	76.0(4.6)•	78.0(2.8)•	73.8(2.5)•	76.5(2.6)•	86.5(4.3)•	88.2(6.1)•	92.0(1.7)•	**95.2(1.3)**
	20%	53.6(5.2)•	83.3(3.3)•	74.9(3.9)•	76.1(2.6)•	73.8(2.5)•	76.5(2.6)•	84.6(2.4)•	89.9(6.6)⊙	91.6(3.1)•	**94.1(1.1)**
	30%	37.2(3.4)•	74.7(4.1)•	72.2(5.1)•	73.0(4.5)•	70.1(4.2)•	69.6(3.4)•	83.9(3.5)•	90.5(4.5)⊙	89.3(2.7)•	**92.1(2.5)**
	40%	30.9(2.9)•	77.0(4.6)•	69.3(3.4)•	70.3(4.9)•	66.8(3.5)•	64.8(4.3)•	78.7(3.1)•	83.4(4.3)•	88.7(2.6)⊙	**89.4(2.3)**
	50%	24.6(1.4)•	74.1(5.3)•	62.4(5.4)•	63.1(5.9)•	62.5(4.5)•	59.2(3.0)•	77.1(5.5)•	81.3(4.6)•	84.0(1.5)•	**85.7(1.9)**
Caltech7	10%	61.8(3.0)•	69.7(2.1)•	69.3(0.9)•	74.5(1.7)•	73.4(1.2)•	**77.2(2.5)**⊙	66.8(2.1)•	76.4(1.6)⊙	75.0(1.6)⊙	76.7(2.9)
	20%	45.6(5.0)•	68.1(2.7)•	69.2(1.3)•	74.2(1.4)⊙	71.2(0.9)•	**76.8(1.6)**⊙	67.1(1.3)•	74.6(3.1)⊙	74.1(1.6)⊙	76.0(2.2)
	30%	32.7(1.9)•	66.1(3.3)•	68.7(1.8)•	73.9(2.0)⊙	71.5(1.4)•	**75.9(2.2)**⊙	64.1(2.6)•	72.2(4.0)⊙	72.6(2.0)•	74.8(2.3)
	40%	26.7(2.2)•	65.2(2.9)•	67.7(1.8)•	73.2(0.6)⊙	70.3(1.8)⊙	**74.1(1.1)**∘	60.0(4.8)•	68.9(6.6)⊙	71.7(2.6)⊙	71.7(2.9)
	50%	24.1(0.4)•	61.9(5.0)•	64.7(2.9)•	69.6(2.8)⊙	66.6(2.7)⊙	**69.7(1.7)**⊙	53.6(3.6)•	61.3(5.9)•	68.8(3.0)⊙	**69.7(2.2)**
Digits	10%	71.7(2.5)•	77.3(4.2)•	81.5(3.9)•	86.2(1.6)•	86.7(2.9)•	81.8(4.6)•	62.6(2.6)•	86.5(0.8)•	**95.9(0.8)**⊙	95.8(0.9)
	20%	63.5(1.3)•	76.3(3.4)•	79.7(4.6)•	87.9(0.8)•	87.0(2.3)•	78.6(5.3)•	59.3(3.8)•	85.7(1.5)•	**95.0(1.1)**⊙	94.9(0.8)
	30%	55.6(2.3)•	76.2(3.7)•	79.2(4.3)•	83.2(3.1)•	87.3(2.3)•	76.3(4.7)•	53.8(4.0)•	86.2(1.0)•	94.0(1.1)⊙	**94.4(1.3)**
	40%	47.6(2.2)•	75.0(4.5)•	74.0(5.2)•	80.9(5.0)•	86.6(2.1)•	75.7(2.8)•	50.0(5.0)•	84.7(2.2)•	**93.0(1.5)**⊙	92.9(1.0)
	50%	39.5(2.0)•	73.8(3.6)•	68.4(5.5)•	75.4(3.6)•	86.6(2.1)•	75.7(2.8)•	47.0(2.8)•	84.5(2.3)•	90.4(1.1)⊙	**90.8(0.8)**
Yale	10%	53.9(3.7)•	55.5(3.6)•	66.7(3.5)•	66.7(1.7)•	63.5(3.8)•	66.8(2.6)•	69.7(2.9)•	69.0(2.1)•	70.7(1.2)•	**72.7(2.1)**
	20%	47.3(3.2)•	53.5(4.5)•	62.4(3.4)•	62.2(2.9)•	59.0(2.4)•	63.2(2.6)•	66.1(3.1)•	66.5(1.8)•	69.6(1.5)•	**70.8(1.8)**
	30%	43.3(2.2)•	46.4(3.3)•	59.1(2.2)•	57.5(3.7)•	55.0(1.7)•	58.7(3.3)•	63.0(3.4)•	61.7(3.5)•	65.3(2.2)•	**67.6(1.3)**
	40%	36.1(2.1)•	45.9(4.5)•	53.5(2.2)•	52.4(3.4)•	50.2(3.1)•	54.7(3.1)•	56.7(3.9)•	56.6(3.6)•	58.4(3.0)•	**60.9(2.5)**
	50%	33.2(2.5)•	43.5(3.5)•	49.9(4.0)•	48.2(3.4)•	50.2(3.1)•	54.7(3.1)•	54.0(3.4)•	54.1(2.9)•	54.4(3.0)•	**58.4(3.7)**
ORL	10%	60.2(2.1)•	62.5(3.7)•	61.7(2.4)•	73.2(1.5)•	62.4(2.1)•	70.9(1.6)•	77.9(2.8)•	74.4(2.5)•	78.6(1.6)⊙	**80.4(1.9)**
	20%	52.0(2.3)•	58.7(2.7)•	56.8(1.8)•	68.9(2.7)•	58.8(2.5)•	69.2(3.6)•	70.2(1.6)•	73.5(2.8)•	74.9(1.4)•	**77.1(1.8)**
	30%	47.5(1.4)•	55.0(4.3)•	53.7(2.1)•	62.8(2.7)•	56.6(2.1)•	63.6(2.3)•	66.6(3.4)•	69.2(1.9)•	71.5(2.6)•	**73.4(2.2)**
	40%	40.8(2.7)•	50.5(2.9)•	50.1(1.9)•	58.3(3.0)•	51.8(2.1)•	57.5(2.6)•	59.1(2.3)•	65.7(3.8)⊙	64.8(2.8)•	**67.1(2.6)**
	50%	34.9(2.2)•	45.1(2.2)•	44.8(1.6)•	49.7(1.5)•	48.5(1.9)•	54.6(3.3)•	53.2(3.2)•	61.5(2.7)⊙	56.4(1.6)•	**61.8(2.4)**
Ionosphere	10%	66.1(1.2)•	61.4(2.9)•	70.1(0.8)•	66.6(1.5)•	71.4(0.7)•	70.6(0.7)•	71.6(0.5)•	52.0(0.8)•	75.7(2.7)⊙	**77.6(2.3)**
	20%	67.2(2.7)•	61.5(2.2)•	71.4(4.5)•	67.4(2.9)•	71.7(0.6)•	71.4(2.6)•	72.4(0.7)•	51.7(1.2)•	**77.1(3.6)**⊙	75.9(2.1)
	30%	65.4(1.3)•	62.1(2.8)•	69.5(2.4)•	65.4(4.1)•	71.6(1.2)•	69.9(1.1)•	73.0(0.7)•	52.4(0.7)•	**77.1(3.7)**⊙	74.8(1.6)
	40%	66.6(3.1)•	61.9(3.7)•	71.2(3.3)•	64.9(4.2)•	71.2(1.2)•	69.6(1.3)•	74.0(1.5)⊙	51.9(1.2)•	**77.0(4.4)**⊙	75.8(3.7)
	50%	66.1(2.6)•	61.4(5.3)•	74.8(2.3)⊙	61.9(2.8)•	71.8(1.0)•	70.0(2.0)•	73.6(1.3)⊙	51.8(1.0)•	**75.6(3.8)**⊙	74.4(2.2)
Forest	10%	68.7(5.1)•	68.2(8.9)•	77.6(0.8)•	79.6(1.4)•	78.5(0.4)•	78.4(0.6)•	78.4(1.2)•	83.3(0.9)•	**86.3(1.0)**⊙	85.5(1.2)
	20%	67.8(4.7)•	68.7(7.4)•	76.7(0.7)•	78.3(1.4)•	78.6(0.7)•	78.6(0.9)•	76.6(1.3)•	83.2(1.1)•	**85.5(0.8)**⊙	85.0(0.9)
	30%	66.1(10)•	68.7(2.5)•	77.0(1.1)•	77.6(2.3)•	77.9(0.8)•	78.5(1.0)•	74.5(1.6)•	83.4(0.6)•	**85.6(1.2)**⊙	85.0(1.0)
	40%	59.0(6.7)•	62.3(8.8)•	77.0(0.9)•	74.4(1.4)•	78.1(0.3)•	78.5(0.8)•	71.1(1.7)•	83.4(1.2)•	84.3(1.4)⊙	**84.8(1.4)**
	50%	51.3(1.5)•	60.9(11.7)•	76.5(1.1)•	72.5(1.7)•	78.2(0.6)•	78.4(0.7)•	71.1(3.6)•	82.9(1.3)⊙	**84.4(1.9)**∘	83.2(1.7)
WebKB	10%	79.1(1.4)•	81.3(9.5)⊙	74.4(1.0)•	78.1(0.6)•	70.4(2.8)•	83.7(4.5)•	72.7(2.7)•	78.3(1.3)•	87.5(1.3)⊙	**87.6(1.0)**
	20%	79.0(1.1)•	75.7(8.6)•	75.9(1.6)•	78.6(1.1)•	72.2(1.8)•	84.9(4.4)⊙	76.9(3.2)•	78.7(2.3)•	**87.8(1.5)**⊙	86.4(1.8)
	30%	78.4(0.2)•	78.4(7.7)•	76.2(2.1)•	80.3(1.0)•	75.0(2.5)•	85.3(3.9)⊙	79.4(2.7)•	78.0(0.2)•	**87.0(1.4)**∘	85.3(1.2)
	40%	78.3(0.3)•	75.7(5.8)•	75.3(1.7)•	80.9(1.4)•	76.2(1.4)•	86.2(3.3)⊙	78.2(3.3)•	78.0(0.2)•	85.9(1.9)⊙	**86.9(1.8)**
	50%	78.2(0.1)•	76.2(5.6)•	75.6(2.2)•	80.8(1.8)•	76.5(2.4)•	**86.3(2.6)**⊙	77.1(2.8)•	78.0(0.2)•	85.7(1.4)⊙	86.2(1.9)
win/tie/lose		40/0/0	39/1/0	39/1/0	36/4/0	38/2/0	31/8/1	37/3/0	31/9/0	14/24/2	-

STD (%) is in the parentheses. The first highest score is in bold. Symbols '•/⊙/∘' denote that JRLC-NS is better/tied/worse than the corresponding method by the paired t-test with confidence level 0.05, respectively. The win/tie/loss counts are reported in the last row.

where $J(t)$ is the objective value in the tth iteration. Since the problem (28) is solved by an iterative strategy, its stop criteria is (36) and its max iteration number is set as 20.

5.2.4 Evaluation Metric

The clustering performance is evaluated in terms of accuracy (ACC) and the normalized mutual information (NMI). For a fair comparison, on each multi-view dataset, every time we create incomplete datasets with different IERs of missing samples and repeat 10 independent times. The average result with standard deviation (STD) is reported.

5.3 Clustering Results Comparison

Tables 3 and 4 show the clustering comparisons of all ten compared methods on eight datasets in incomplete multi-view setting w.r.t. ACC and NMI, respectively. According to results, we have the following observations.

As IER increases, in terms of both ACC and NMI, the performance of all compared methods becomes worse in most cases, which is consistent with intuition.

With the increment of IER, MIC suffers from more performance degeneration than other matrix factorization-based methods MVL-IV, IMG and DAIMC on datasets MSRC-v1, Caltech7, Dights and Forest. This might be because that MIC simply fills the missing samples of each view with the global feature average, which may lead to a deviation especially when IER is large.

Comparing the results of MVL-IV, IMG and DAIMC, each of them achieves good performance on certain datasets but performs worse on other datasets. Since MVL-IV, IMG and DAIMC can all be categorized into Eq. (2), the possible

TABLE 4
NMI (%) Comparisons on 8 Datasets With Different IERs

Data set	IER	MIC	MVL-IV	IMG	DAIMC	DMF-F	DMF-R	IMKK-MKC	PIC	JRLC-SE	JRLC-NS
	10%	61.6(3.2)•	75.1(2.2)•	65.0(3.4)•	66.6(2.4)•	63.0(1.8)•	65.9(2.7)•	79.4(3.7)•	86.1(4.9)•	86.1(2.5)•	**90.4(2.0)**
	20%	43.6(4.2)•	72.8(3.8)•	63.5(3.4)•	64.5(1.7)•	61.8(2.3)•	61.4(2.9)•	77.4(2.8)•	86.1(5.9)⊙	85.6(4.3)•	**88.6(1.5)**
MSRC-v1	30%	27.6(2.9)•	65.6(4.1)•	60.4(3.9)•	62.2(3.7)•	58.3(2.6)•	59.7(3.9)•	76.2(2.9)•	85.7(4.0)⊙	82.8(2.9)•	**85.8(3.3)**
	40%	17.2(2.9)•	65.4(4.5)•	56.8(3.4)•	57.3(3.6)•	56.2(3.2)•	52.8(3.5)•	70.0(2.4)•	77.2(3.3)•	79.4(4.1)⊙	**80.8(2.9)**
	50%	11.2(1.5)•	61.6(4.3)•	50.1(4.5)•	50.5(4.2)•	51.1(4.8)•	48.0(2.9)•	66.3(5.6)•	74.2(3.0)⊙	72.5(1.6)•	**74.4(2.7)**
	10%	52.4(4.4)•	59.4(1.7)•	62.2(1.4)•	68.3(1.8)•	69.8(0.7)•	70.3(1.8)•	55.0(1.5)•	73.9(1.9)⊙	72.9(3.3)•	**75.9(3.2)**
	20%	31.7(4.4)•	58.1(3.5)•	61.9(2.6)•	66.3(1.1)•	67.3(1.9)•	69.0(1.9)•	56.1(2.5)•	70.9(2.4)•	70.4(2.9)•	**74.2(3.3)**
Caltech7	30%	16.2(2.3)•	55.5(2.3)•	60.8(2.2)•	62.8(2.8)•	63.0(1.6)•	65.5(1.7)•	52.0(3.6)•	66.1(3.7)•	68.0(2.7)⊙	**70.9(2.9)**
	40%	9.81(2.1)•	52.2(3.0)•	56.9(2.9)•	59.3(1.9)•	61.4(1.8)•	62.5(1.5)•	45.8(3.7)•	63.0(4.7)•	**67.1(4.0)**⊙	65.7(4.1)
	50%	6.37(1.0)•	47.1(4.9)•	54.6(2.7)•	53.6(2.8)•	54.4(2.0)•	56.3(1.7)⊙	40.8(2.8)•	54.0(6.0)•	60.3(4.0)⊙	**60.7(3.1)**
	10%	64.7(1.9)•	71.4(2.7)•	73.8(1.9)•	76.5(1.5)•	79.2(1.7)•	76.1(2.0)•	57.6(0.9)•	86.6(0.9)•	**91.7(1.0)**⊙	**91.7(0.9)**
	20%	56.8(1.5)•	70.2(1.6)•	72.0(2.1)•	78.0(1.1)•	78.8(1.8)•	74.6(2.2)•	54.3(2.0)•	86.1(1.2)•	90.1(1.2)⊙	**90.2(1.1)**
Digits	30%	50.9(1.6)•	70.1(1.8)•	71.7(1.4)•	73.3(1.9)•	78.3(1.6)•	72.1(2.4)•	49.0(2.0)•	85.8(1.2)•	88.1(1.5)⊙	**89.1(1.2)**
	40%	44.2(1.5)•	67.7(3.5)•	67.3(2.4)•	70.4(3.1)•	76.4(1.5)•	70.2(2.9)•	45.1(2.3)•	84.7(0.9)•	86.3(1.4)⊙	**86.6(1.1)**
	50%	35.3(2.8)•	65.1(2.5)•	61.5(2.6)•	64.6(1.8)•	73.0(1.4)•	65.4(2.2)•	45.4(2.4)•	**84.0(1.0)**⊙	82.6(1.2)⊙	83.5(0.8)
	10%	58.9(3.2)•	59.7(3.0)•	70.3(2.2)•	70.6(1.8)•	70.2(2.7)•	70.9(2.9)•	72.4(2.7)⊙	72.0(1.4)•	72.2(1.4)•	**74.0(2.0)**
	20%	53.4(3.4)•	58.1(3.3)•	66.9(3.3)•	66.4(3.2)•	65.6(2.6)•	66.8(2.5)•	68.7(2.4)•	69.7(2.0)•	70.3(1.4)•	**71.9(1.2)**
Yale	30%	49.0(1.8)•	53.2(3.5)•	62.7(1.5)•	61.8(2.4)•	61.6(1.5)•	63.4(2.2)•	64.7(2.1)•	65.0(2.9)•	65.4(2.0)•	**68.3(1.8)**
	40%	43.1(2.8)•	50.4(5.1)•	58.5(2.1)•	56.9(2.5)•	55.5(1.7)•	59.0(2.3)•	59.0(2.9)•	60.1(2.8)•	59.5(3.3)•	**62.0(2.7)**
	50%	40.4(2.4)•	48.9(3.3)•	54.3(3.0)•	53.9(3.3)•	53.9(3.0)•	55.6(2.6)•	56.7(3.2)⊙	57.5(3.3)•	56.2(3.0)•	**59.5(2.9)**
	10%	76.1(1.1)•	79.7(1.6)•	78.8(1.6)•	87.0(1.5)•	79.8(0.9)•	85.5(1.0)•	88.6(1.3)⊙	87.6(0.8)•	87.9(0.8)•	**89.1(0.8)**
	20%	68.3(1.7)•	76.3(1.6)•	75.2(1.1)•	82.7(1.1)•	77.1(1.2)•	83.5(2.5)•	83.5(1.3)•	**86.0(1.5)**⊙	84.9(0.6)•	**86.0(0.9)**
ORL	30%	63.5(1.3)•	72.6(2.9)•	72.5(1.5)•	78.4(1.5)•	74.5(1.0)•	79.1(1.2)•	80.2(2.0)•	**83.5(1.2)**⊙	81.8(1.6)•	83.1(1.4)
	40%	58.5(2.0)•	68.7(1.7)•	68.8(1.5)•	74.7(1.9)•	70.9(1.9)•	74.6(1.4)•	74.8(1.6)•	**79.7(2.4)**○	76.8(1.3)•	78.3(1.5)
	50%	53.7(1.7)•	63.7(1.7)•	64.4(1.3)•	68.8(0.6)•	68.5(1.4)•	72.1(1.5)•	69.9(1.6)•	**76.0(1.6)**○	71.6(1.0)•	74.7(1.9)
	10%	9.1(0.9)•	2.6(1.6)•	11.1(1.1)•	7.6(1.6)•	13.6(1.1)•	12.2(0.8)•	14.3(0.8)•	11.5(1.3)•	24.3(5.3)⊙	**27.4(4.1)**
	20%	8.7(1.9)•	2.6(1.4)•	12.3(6.4)•	8.9(1.6)•	14.1(0.9)•	12.9(2.9)•	15.7(1.2)•	10.9(1.7)•	**26.5(8.4)**⊙	25.3(3.5)
Ionosphere	30%	6.7(1.2)•	3.2(1.9)•	9.7(2.0)•	7.4(3.7)•	13.8(1.7)•	11.4(1.7)•	16.8(1.7)•	12.6(1.8)•	**26.9(6.5)**⊙	22.9(3.7)
	40%	6.1(2.9)•	3.3(2.0)•	10.8(5.3)•	7.1(3.5)•	13.1(1.5)•	10.5(1.9)•	18.6(3.1)•	10.8(2.1)•	**26.6(7.1)**⊙	22.6(5.1)
	50%	5.0(2.3)•	3.7(3.9)•	16.0(4.3)•	3.6(2.0)•	13.9(1.6)•	12.0(3.5)•	18.1(2.6)•	11.2(1.6)•	**24.9(6.2)**⊙	21.7(3.3)
	10%	45.3(2.5)•	42.0(7.6)•	53.0(0.8)•	55.3(2.5)•	53.8(0.6)•	53.9(0.8)•	53.1(1.4)•	62.0(1.1)⊙	**64.7(1.8)**○	63.0(2.2)
	20%	44.0(3.7)•	41.9(7.2)•	52.0(0.5)•	52.8(2.2)•	54.2(0.9)•	54.3(1.3)•	50.5(1.8)•	61.4(1.6)⊙	**63.7(1.2)**⊙	62.5(1.5)
Forest	30%	44.6(5.8)•	39.5(5.1)•	52.4(0.8)•	52.0(2.4)•	53.6(0.7)•	53.7(1.6)•	47.7(2.2)•	61.4(0.7)⊙	**63.8(2.2)**○	62.4(1.5)
	40%	36.2(5.1)•	31.9(8.1)•	52.1(1.0)•	46.8(2.6)•	53.4(0.6)•	53.5(1.0)•	43.9(1.7)•	61.2(1.7)⊙	62.3(2.0)⊙	**62.6(2.3)**
	50%	30.0(1.4)•	32.8(11.3)•	51.7(1.2)•	45.1(3.1)•	53.8(0.5)•	53.3(1.2)•	44.2(4.8)•	60.0(2.1)⊙	**61.7(2.8)**○	60.0(2.3)
	10%	11.3(3.8)•	24.7(17.5)•	5.8(4.1)•	6.18(3.6)•	0.8(0.2)•	36.2(7.5)⊙	10.2(4.0)•	3.5(5.9)•	39.6(4.7)⊙	**41.5(3.0)**
	20%	10.6(3.4)•	11.4(12.5)•	5.4(1.4)•	5.6(4.2)•	2.0(0.7)•	35.3(9.3)⊙	17.7(6.7)•	3.9(8.1)•	**39.8(4.6)**⊙	37.7(5.7)
WebKB	30%	7.7(2.0)•	16.6(8.9)•	6.2(2.5)•	11.5(4.6)•	4.1(2.2)•	**36.4(8.5)**⊙	22.5(6.1)•	1.3(1.0)•	35.3(4.6)⊙	32.4(3.9)
	40%	6.0(2.0)•	9.3(6.3)•	5.7(1.7)•	14.0(5.4)•	5.3(1.8)•	**35.5(9.9)**⊙	21.4(7.5)•	1.4(0.8)•	31.5(6.2)⊙	33.6(6.5)
	50%	3.4(2.0)•	7.6(7.1)•	5.4(2.7)•	12.5(5.3)•	5.3(2.5)•	**33.4(6.4)**⊙	20.7(7.7)•	1.6(1.1)•	29.1(6.0)⊙	32.5(3.9)
win/tie/lose		40/0/0	40/0/0	40/0/0	40/0/0	40/0/0	34/6/0	37/3/0	26/12/2	16/21/3	-

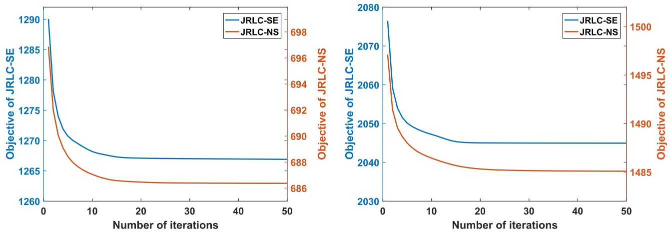
(See the title of Table 1 for more information).

reason is that these methods adopt different regularization terms or constraints, which makes them good at grouping certain kind of data and poor at clustering the others.

DMF-F and DMF-R achieve comparable or even better results than matrix factorization-based incomplete multi-view clustering methods, and on some datasets such as Caltech7 and WebKB, the advantages of DMF-R are significant. That can be owing to that as a deep method, DMF-R can better disclose the non-linear structure of data than traditional matrix factorization-based methods.

IMKK-MKC and PIC achieve significantly better results than matrix factorization-based methods on some datasets. This might be because that kernel-based method IMKK-MKC and graph-based method PIC can disclose the nonlinear structure of data. However, IMKK-MKC and PIC achieve the worst performance on some datasets, respectively. The possible reason is that the completion of kernels or graph similarity matrices introduces uncertain information, which may degenerate the performance of their subsequent clustering process.

JRLC-SE and JRLC-NS achieve better or comparable performance than other methods do over all datasets in most cases as IER varies from 10 to 50 percent. This may be because that JRLC-SE and JRLC-NS integrate representation learning and clustering, which explore the underlying structure of partial similarity matrices directly for clustering without introducing uncertain information. Compared with JRLC-SE, the performance of JRLC-NS achieves more considerable improvements in several datasets. It can be owing to that JRLC-NS inherits advantages of both nonnegative embedding and spectral embedding, which makes its reconstructed graph matrices have more clear structures than them of JRLC-SE. Besides, by learning different $\mathbf{F}^{(v)}$ and $\mathbf{H}^{(v)}$, JRLC-NS enables $\{\mathbf{F}^{(v)}\}_{v=1}^{V}$ to pay more attention to



(a) MSRC-v1 (b) Caltech7

Fig. 1. Sensitivity analysis on parameters λ and γ.

seek the consistent clustering, thereby further improving clustering performance.

5.4 Convergence Analysis and Time Comparison

In order to verify the convergence behaviors of the proposed Algorithm 2 for JRLC-SE and Algorithm 3 for JRLC-NS, we present their convergence behavior curves on datasets MSRC-v1 and Caltech7 with IER=50%. The convergence behavior curves are displayed in Fig. 1.

As we can see from Fig. 1, both Algorithms 2 and 3 monotonically decrease their corresponding objective values as the iteration round increases and converge to a fixed value. Additionally, as the iteration round increases, the objective values of both JRLC-SE and JRLC-NS decrease fast, indicating Algorithms 2 and 3 have fast convergence property.

To demonstrate the efficiency of the proposed algorithms to deal with incomplete multi-view data, we report runtime comparisons on two datasets Digits and WebKB. Digits has the largest data size while WebKB is with the largest dimensionality. For every time, we create incomplete data with IER=50%, and implement each method on it with predetermined parameters. The average results of 5 independent times with STD are reported in Table 5.

From the results of Table 5, we have the following observations: 1) JRLC-NS and JRLC-SE spend less time than other methods, because their optimization only have linear complexity w.r.t. present data size and are irrelevant w.r.t. dimensionality. Compared with JRLC-SE, JRLC-NS has no inner loop, which further reduces the runtime. 2) PIC and IMKK-MKC use less time than matrix factorization-based methods on WebKB because they only have cubic complexity w.r.t. data size. 3) MVL-IV takes less time than other matrix factorization-based methods because it has linear complexity w.r.t. both data size and dimensionality. 4) DMF-F and DMF-R cost more time than MVL-IV because DMF needs more number of iterations. 5) IMG spends the most time on Digits, because it constructs an adaptive graph matrix based on common representations in each iteration. 6) DAIMC costs the most time on WebKB, because it solves the continuous Sylvester equation in each iteration, which has cubic complexity w.r.t. dimensionality of each view.

TABLE 5
Computational Time (seconds) on 2 Datasets With IER=50%

	Digits	WebKB
MIC	365.1702(7.3071)	39.9132(11.529)
MVL-IV	6.3518(0.8086)	13.1667(3.8341)
IMG	741.2341(4.6199)	146.3718(13.247)
DAIMC	113.9044(6.6345)	196.8161(47.287)
DMF-F	52.1686(0.2251)	75.7302(0.3392)
DMF-R	51.2226(0.1988)	75.1241(0.2350)
IMKK-MKC	103.5268(3.6927)	3.7295(1.2187)
PIC	185.7268(1.8307)	3.0080(0.3263)
JRLC-SE	4.6706(0.2129)	0.6453(0.0645)
JRLC-NS	3.1948(0.1146)	0.3548(0.0388)

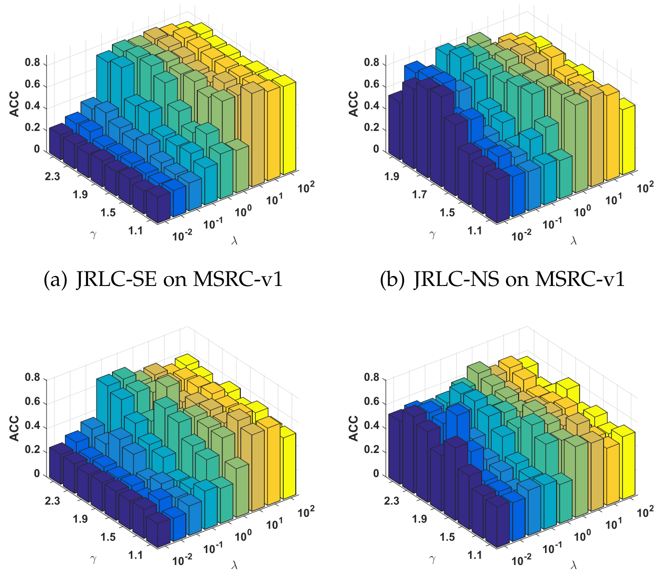
STD (seconds) is in the parentheses.

TABLE 6
ACC (%) and NMI (%) Comparisons on 8 Datasets With IER=25%

Dataset	Merit	SE+C	JRLC-SE	NS+C	JRLC-NS
MSRC-v1	ACC	87.7(3.1)•	91.3(2.2)	85.7(2.1)•	93.0(1.2)
	NMI	79.8(2.6)•	84.8(2.3)	77.9(2.6)•	87.2(2.2)
Caltech7	ACC	67.1(2.2)•	73.7(1.5)	66.0(2.9)•	74.9(2.1)
	NMI	60.3(2.8)•	70.4(3.6)	59.3(3.9)•	72.4(2.8)
Digits	ACC	93.9(0.5)•	95.5(0.5)	92.3(0.8)•	94.9(0.8)
	NMI	87.6(0.8)•	90.1(0.8)	85.5(1.1)•	89.5(1.0)
Yale	ACC	66.5(2.8)•	67.8(2.4)	66.8(2.9)•	69.4(2.0)
	NMI	67.0(2.0)•	68.1(2.0)	66.5(1.8)•	69.8(1.8)
ORL	ACC	71.4(2.1)⊙	72.2(2.1)	71.9(1.3)•	75.2(1.3)
	NMI	82.6(1.0)⊙	83.1(1.1)	82.7(0.7)•	84.2(1.2)
Ionosphere	ACC	69.4(4.1)•	78.0(3.4)	70.9(6.1)⊙	75.2(1.7)
	NMI	14.5(4.5)•	28.5(5.0)	16.6(4.9)•	24.5(4.5)
Forest	ACC	75.9(6.7)•	85.6(0.8)	78.9(5.4)•	84.8(0.7)
	NMI	52.6(6.2)•	63.9(1.6)	55.2(6.1)•	62.5(1.0)
WebKB	ACC	76.7(0.6)•	87.2(1.1)	78.3(2.3)•	87.0(1.9)
	NMI	0.6(0.4)•	37.3(2.5)	5.5(6.7)•	36.0(5.7)
win/tie/lose		14/2/0	-	15/1/0	-

5.5 Ablation Test

To demonstrate the effectiveness of integrating both representation learning and clustering processes, we compare JRLC-SE and JRLC-NS with SE+C and NS+C, respectively.



(a) JRLC-SE on MSRC-v1 (b) JRLC-NS on MSRC-v1

(c) JRLC-SE on Caltech7 (d) JRLC-NS on Caltech7

Fig. 2. Sensitivity analysis on parameters λ and γ.

TABLE 7
Details of Ten Incomplete Multi-View Datasets (present data size (dimensionality))

View	3Sourses	B-G	B-R	G-R	BBC2	BBC3	BBC4	BBCSport2	BBCSport3	BBCSport4
1	352(3560)	352(3560)	352(3560)	302(3631)	2125 (6838)	1828 (5470)	1543 (4659)	644 (3183)	519 (2582)	400 (1991)
2	302(3631)	302(3631)	294(3068)	294(3068)	2112 (6790)	1832 (5549)	1524 (4633)	637 (3203)	531 (2544)	410 (2063)
3	294(3068)	-	-	-	-	1845 (5483)	1574 (4665)	-	513 (2465)	437 (2113)
4	-	-	-	-	-	-	1549 (4684)	-	-	432 (2158)
Dats size	416	404	407	384		2225			737	
Classes	6	6	6	6		5			5	

SE+C and NS+C first learn view-specific representations by solving the problem (7) with corresponding constraints, and then extract the clustering results based on these fixed representations by solving the problem (8). The experiments are conducted on afore-mentioned eight datasets with IER=25%. All parameters are determined by grid search, and the search ranges are introduced in Section 5.2.3. On each dataset, we create incomplete data for 10 independent times, and average results of ACC and NMI with best tuned parameters are reported in Table 6.

As we can see from Table 6, JRLC-SE and JRLC-NS achieves better results in terms of both ACC and NMI on all datasets than the corresponding SE+C and NS+C, and the improvements are significant on most cases. Compared with results of baselines in Tables 3 and 4, on some datasets, the results of SE+C and NS+C are worse than them with larger IERs, while JRLC-SE and JRLC-NS outperform them, which further indicates indicates that connecting representation learning and clustering can achieve better performance.

5.6 Parameter Study

We study the influence of hyper-parameters γ and λ on the performance of JRLC-SE and JRLC-NS. $\gamma \geqslant 1$ controls the smoothness extent of the distribution of the common probability label matrix and balances the view-specific term and the co-regularization term. λ controls the scaling of the reconstructed similarity matrices. γ is tuned in the range of $\{1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5\}$ while λ is varied from $\{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0}, 10^{0.5}, 10^{1}, 10^{1.5}, 10^{2}\}$. The experiments are conducted on MSRC-v1 and Caltech7. On each dataset, IER is fixed as 50 percent. Since NMI has similar tendency with ACC, Fig. 2 shows ACC results with varying parameters γ and λ on 2 datasets.

From Fig. 2, we observe that: 1) The performance of JRLC-SE is more affected by λ. With suitable λ, JRLC-SE can achieves acceptable results by tuning γ. Compared with JRLC-SE, JRLC-NS achieves acceptable performance in a wider range. 2) On the two datasets, JRLC-SE have different optimal parameters. And JRLC-NS has the same situation. Therefore, for both JRLC-SE and JRLC-NS, how to identify the optimal parameters is data-dependent. Two datasets have different optimal parameters because their data characteristics are different.

6 Application to News Clustering

News topic clustering aims to identify a set of clusters that accurately reflects the topics present in the news collection. Compared with other traditional clustering tasks, news topic clustering is more complex due to the following two reasons: 1) There are usually different sources to report the same news, which results in multi-view data; 2) Different from those tasks with quantitative features, more time and effort are required for pre-processing data. In real applications, both of these factors can cause incomplete multi-view clustering problem.

3Sources[9] consists 416 news stories collected from three online news sources: BBC, Reuters, and The Guardian. The 416 news are classified into 6 classes, i.e., 104 business stories, 60 entertainment stories, 54 health stories, 49 politics stories, 89 sport stories and 60 tech stories. Since each story may not be reported by all three sources, which results in incomplete views of 3Sources. By selecting news stories belonging two sources, three incomplete datasets can be generated, i.e., BBC-Guardian (B-G), BBC-Reuters (B-R) and Guardian-Reuters (G-R). BBC and BBCSport are two news datasets collected by [45]. BBC is composed of 2225 news documents and is divided to 5 classes, i.e., 510 business documents, 386 entertainment documents, 417 politics documents, 511 sport documents, and 401 tech documents. BBCSport consists of 737 news documents and is divided into 5 classes, i.e., 101 athletics documents, 124 cricket documents, 265 football documents, 147 rugby documents, and 100 tennis documents. In [46], a pre-processing methodology has been proposed. First, it splits each raw document into segments by merging consecutive paragraphs, and this process makes sure that each segment has at least 200 words. Then each segment is assigned to at most one view. Since segments of each document may be assigned to some but not all of views, this methodology results in six incomplete multi-view datasets,[10] i.e., BBC2, BBC3, BBC4, BBCSport2, BBCSport3 and BBCSport4. The brief summaries of the ten incomplete datasets are listed in Table 7.

In this section, we cluster these ten datasets. Similarly, we compare JRLC-SE and JRLC-NS with eight baselines. In this section, the neighbor number k is fixed as 15, and other parameters are determined by the same way as Section 5.2.3. And we repeat 10 independent times, and report the mean ACC and NMI results with STD in Table 8.

From Table 8, we have the following observations: 1) Compared with MVL-IV and IMG, MIC and DAIMC achieve worse results on datasets with large IERs. The possible reason is that the nonnegative constraint limits the flexibility of the representation learning. 2) DMF-F and DMF-R outperform traditional matrix factorization-based methods in most cases. DMF-F achieves better results in some cases, indicating that matrix factorization cannot always well reflect the

9. http://mlg.ucd.ie/datasets/3sources.html
10. http://mlg.ucd.ie/datasets/segment.html

TABLE 8
ACC (%) and NMI (%) Comparisons on 10 Datasets

Dataset	Merit	MIC	MLIV	IMG	DAIMC	DMF-F	DMF-R	IMKK-MKC	PIC	JRLC-SE	JRLC-NS
3Sourses	ACC	61.8(7.5)•	75.8(1.6)•	78.0(0.9)•	71.2(3.0)•	75.7(1.9)•	84.6(2.2)•	83.8(1.0)•	89.3(1.1)⊙	88.9(0.0)•	**89.5(0.5)**
	NMI	57.8(4.6)•	64.1(1.0)•	66.0(1.2)•	60.0(2.1)•	67.8(3.2)•	68.9(2.4)•	71.1(1.2)•	74.9(1.5)⊙	74.5(0.0)•	**75.4(0.7)**
B-G	ACC	63.1(4.6)•	67.4(3.6)•	71.4(3.7)•	69.6(3.1)•	75.1(4.5)•	80.0(2.9)•	75.5(1.7)•	78.7(4.2)•	88.1(0.0)⊙	**88.2(0.8)**
	NMI	55.3(5.0)•	56.2(4.3)•	63.0(1.9)•	59.9(2.9)•	65.4(3.6)•	64.4(3.2)•	62.9(1.4)•	68.6(1.3)•	73.1(0.0)⊙	**73.5(1.4)**
B-R	ACC	58.4(2.6)•	68.0(4.2)•	75.0(2.0)•	73.8(2.3)•	76.6(5.8)•	79.6(2.4)•	77.8(0.2)•	87.6(1.3)•	89.2(0.0)⊙	**89.4(0.3)**
	NMI	54.1(2.6)•	59.3(3.7)•	63.1(1.9)•	60.9(1.4)•	67.7(3.1)•	65.4(2.5)•	65.6(0.3)•	73.7(1.6)•	**75.7(0.0)**⊙	75.6(0.5)
G-R	ACC	61.8(5.7)•	68.9(3.0)•	74.5(1.5)•	66.1(4.1)•	75.1(5.5)•	79.7(5.6)•	74.5(1.5)•	86.2(0.4)•	85.4(0.0)•	**87.7(0.6)**
	NMI	54.7(3.8)•	57.7(2.5)•	62.6(1.1)•	57.0(2.2)•	67.6(3.6)•	64.7(5.2)•	63.3(1.0)•	71.3(0.3)•	70.5(0.0)•	**73.3(1.2)**
BBC2	ACC	80.5(6.2)•	72.2(14)•	86.4(0.2)•	83.7(1.6)•	93.3(0.6)•	89.1(1.2)•	92.1(0.0)•	76.9(5.9)•	**93.8(0.0)**⊙	93.7(0.3)
	NMI	62.8(6.2)•	55.3(14)•	69.5(0.1)•	66.7(1.8)•	**81.5(1.0)**⊙	73.1(1.9)•	77.9(0.1)•	71.3(0.6)•	81.4(0.0)⊙	**81.5(0.7)**
BBC3	ACC	77.8(6.7)•	86.8(4.1)•	86.8(0.3)•	85.0(2.3)•	92.8(1.5)⊙	89.3(2.5)•	91.6(0.1)•	76.6(5.2)•	93.1(0.0)•	**93.5(0.3)**
	NMI	60.1(6.9)•	69.8(5.2)•	69.7(0.5)•	68.1(2.4)•	80.6(3.0)⊙	73.8(3.9)•	76.8(0.2)•	70.8(0.0)•	79.9(0.0)•	**80.9(0.8)**
BBC4	ACC	68.4(6.0)•	83.3(6.8)•	87.5(0.2)•	78.8(3.1)•	85.4(6.9)•	88.8(2.7)•	92.1(0.0)•	77.7(6.6)•	92.4(0.0)⊙	**92.6(0.4)**
	NMI	51.2(3.5)•	65.6(8.3)•	70.6(0.3)•	61.1(4.3)•	74.3(4.6)•	73.3(4.2)•	78.0(0.1)•	67.8(1.7)•	78.7(0.0)⊙	**79.2(0.9)**
BBCSport2	ACC	72.7(8.1)•	79.7(9.2)•	88.7(1.4)•	89.0(1.5)•	93.3(0.7)⊙	88.0(4.7)•	88.9(0.2)•	76.0(5.1)•	**93.5(0.0)**⊙	93.1(1.2)
	NMI	57.9(6.9)•	63.2(6.8)•	72.6(2.1)•	72.8(2.5)•	**82.3(1.4)**⊙	73.2(4.2)•	72.5(0.4)•	75.7(1.5)•	81.8(0.0)⊙	81.2(1.9)
BBCSport3	ACC	68.0(4.5)•	80.0(7.2)•	88.6(0.6)•	59.8(4.7)•	85.5(7.0)•	86.6(2.6)•	88.8(0.0)•	76.0(5.5)•	**93.1(0.0)**⊙	92.9(0.3)
	NMI	54.4(5.4)•	62.6(7.0)•	72.3(1.2)•	42.3(5.8)•	76.6(4.1)•	69.8(3.5)•	72.5(0.1)•	73.7(0.0)•	**81.8(0.0)**○	80.7(0.7)
BBCSport4	ACC	62.8(8.2)•	80.1(5.9)•	88.1(1.3)•	59.0(2.3)•	87.6(5.3)•	85.8(5.1)•	88.7(0.4)•	77.7(7.1)•	**92.1(0.0)**⊙	91.8(0.6)
	NMI	49.2(5.1)•	62.3(4.4)•	71.8(2.2)•	39.7(2.4)•	75.8(3.8)•	70.0(3.9)•	72.4(0.4)•	75.7(2.6)•	**79.3(0.0)**⊙	78.6(1.4)
win/tie/lose		20/0/0	20/0/0	20/0/0	20/0/0	15/5/0	20/0/0	20/0/0	18/2/0	6/13/1	-

(See the title of Table 1 for more information).

structure of data. 3) IMMK-MKC outperforms matrix factorization methods on all datasets, while PIC achieves better performance than IMG only on 3sourse. It is probably because that simple average filling strategy may introduce more bad information. 4) The proposed JRLC-SE and JRLC-NS outperform the state-of-the-art methods on all ten datasets, and the improvements become more significant on datasets with large IERs. On BBC2, BBC3 and BBC4, the increased IER makes the performances of JRLC-SE and JRLC-NS decline slightly. Similar phenomena can be observed on BBCSport2, BBCSport3 and BBCSport4. The possible reason is that the view data matrices have very sparse features, which makes the quality of partial graph construction robust to incompleteness of views.

7 Conclusion

In this paper, we propose JRLC framework, which makes view-specific representation learning and clustering integrated seamlessly to achieve better performance. Under guidance of this framework, several new graph-based incomplete multi-view clustering methods can be developed based on existing single-view representation learning methods. As shown in this paper, within the framework, we propose two specific methods JRLC-SE and JRLC-NS. The optimization algorithms effectively and efficiently solve the resultant problems of JRLC-SE and JRLC-NS, and it demonstrates well improved clustering performance via extensive experiments on several datasets and news topic clustering application. In the future, we plan to design new methods within this framework to achieve better clustering performance. Besides, we are planning to analyze the convergence rate of the proposed algorithms in a future study. Moreover, we want to further improve the proposed framework by taking the correlations of partial similarity matrices into account. Also, extending this framework for incomplete multi-view semi-supervised classification problem is interesting and worth exploring in future.

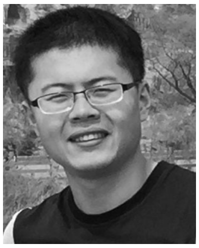
Acknowledgments

This work was supported by the NSF of China under Grant No. 61922087, Grant No. 61906201 and Grant No. 62006238, the NSF for Distinguished Young Scholars of Hunan Province under Grant No. 2019JJ20020, and NSF for Young Scholars of Hunan Province Grant No. 2020JJ5669.

References

[1] Z. Jing, X. Xie, X. Xin, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, 2017.
[2] H. Tao, C. Hou, F. Nie, J. Zhu, and D. Yi, "Scalable multi-view semi-supervised classification via adaptive regression," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4283–4296, Sep. 2017.
[3] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.
[4] M. Karasuyama and H. Mamitsuka, "Multiple graph label propagation by sparse integration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1999–2012, Dec. 2013.
[5] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, no. 7/8, pp. 2031–2038, 2013.

[6] C. Hou, C. Zhang, Y. Wu, and F. Nie, “Multiple view semi-supervised dimensionality reduction,” *Pattern Recognit.*, vol. 43, no. 3, pp. 720–730, 2010.
[7] T. Joachims, N. Cristianini, and J. Shawetaylor, “Composite kernels for hypertext categorisation,” in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 250–257.
[8] A. Kumar, P. Rai, and H. Daume, “Co-regularized multi-view spectral clustering,” in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.
[9] A. Djelouah, J. Franco, E. Boyer, F. L. Clerc, and P. Perez, “Multi-view object segmentation in space and time,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2640–2647.
[10] C. Jin, W. Mao, R. Zhang, Y. Zhang, and X. Xue, “Cross-modal image clustering via canonical correlation analysis,” in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 151–159.
[11] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, “From ensemble clustering to multi-view clustering,” in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2843–2849.
[12] S. Li, Y. Jiang, and Z. Zhou, “Partial multi-view clustering,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1968–1974.
[13] H. Zhao, H. Liu, and Y. Fu, “Incomplete multi-modal visual data grouping,” in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2392–2398.
[14] N. Xu, Y. Guo, X. Zheng, Q. Wang, and X. Luo, “Partial multi-view subspace clustering,” in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 1794–1801.
[15] Q. Yin, S. Wu, and L. Wang, “Incomplete multi-view clustering via subspace learning,” in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 383–392.
[16] W. Shao, L. He, and P. S. Yu, “Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization,” in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2015, pp. 318–334.
[17] M. Hu and S. Chen, “Doubly aligned incomplete multi-view clustering,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2262–2268.
[18] C. Xu, D. Tao, and C. Xu, “Multi-view learning with incomplete views,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5812–5825, Dec. 2015.
[19] H. Tao, C. Hou, D. Yi, and J. Zhu, “Unsupervised maximum margin incomplete multi-view clustering,” in *Proc. Int. CCF Conf. Artif. Intell.*, 2018, pp. 13–25.
[20] A. Trivedi, P. Rai, H. Daumé III, and S. L. DuVall, “Multiview clustering with incomplete views,” in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2010, vol. 224.
[21] S. Bhadra, S. Kaski, and J. Rousu, “Multi-view kernel completion,” *Mach. Learn.*, vol. 106, no. 5, pp. 713–739, 2017.
[22] X. Liu *et al.*, “Multiple kernel k-means with incomplete kernels,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, May 2020.
[23] H. Gao, Y. Peng, and S. Jian, “Incomplete multi-view clustering,” in *Proc. Int. Conf. Intell. Inf. Process.*, 2016, pp. 245–255.
[24] J. Wen, Y. Xu, and H. Liu, “Incomplete multiview spectral clustering with adaptive graph learning,” *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1418–1429, Apr. 2020.
[25] H. Wang, L. Zong, B. Liu, Y. Yang, and W. Zhou, “Spectral perturbation meets incomplete multi-view data,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3677–3683.
[26] F. Monti, M. Bronstein, and X. Bresson, “Geometric matrix completion with recurrent multi-graph neural networks,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3697–3707.
[27] J. Fan and T. W. S. Chow, “Deep learning based matrix completion,” *Neurocomputing*, vol. 266, pp. 540–549, 2017.
[28] J. Fan and J. Cheng, “Matrix completion by deep matrix factorization,” *Neural Netw.*, vol. 98, pp. 34–41, 2018.
[29] W. Zhuge, C. Hou, X. Liu, H. Tao, and D. Yi, “Simultaneous representation learning and clustering for incomplete multi-view data,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4482–4488.
[30] L. Zelnik-Manor, “Self-tuning spectral clustering,” in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, 2004, vol. 17, pp. 1601–1608.
[31] F. Nie, X. Wang, M. I. Jordan, and H. Huang, “The constrained Laplacian rank algorithm for graph-based clustering,” in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.
[32] T. Hastie, J. Friedman, and R. Tibshirani, “The elements of statistical learning,” *Technometrics*, vol. 45, no. 3, pp. 267–268, 2010.
[33] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.*, 2002, vol. 2, pp. 849–856.
[34] D. Kuang, C. Ding, and H. Park, “Symmetric nonnegative matrix factorization for graph clustering,” in *Proc. SIAM Int. Conf. Data Mining*, 2012, pp. 106–117.
[35] R. Arora, M. R. Gupta, A. Kapila, and M. Fazel, “Clustering by left-stochastic matrix factorization,” in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 761–768.
[36] C. Ding, T. Li, and M. I. Jordan, “Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding,” in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 183–192.
[37] J. Han, K. Xiong, and F. Nie, “Orthogonal and nonnegative graph reconstruction for large scale clustering,” in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1809–1815.
[38] Z. Hu, F. Nie, R. Wang, and X. Li, “Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding,” *Inf. Fusion*, vol. 55, pp. 251–259, 2020.
[39] F. Nie, R. Zhang, and X. Li, “A generalized power iteration method for solving quadratic problem on the stiefel manifold,” *Sci. China Inf. Sci.*, vol. 60, no. 11, pp. 146–155, 2017.
[40] Y. J. Lee and K. Grauman, “Foreground focus: Unsupervised learning from partially matching images,” *Int. J. Comput. Vis.*, vol. 85, no. 2, pp. 143–166, 2009.
[41] D. Dueck and B. J. Frey, “Non-metric affinity propagation for unsupervised image categorization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[42] C. Hou, L.-L. Zeng, and D. Hu, “Safe classification with augmented features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2176–2192, Sep. 2019.
[43] B. Johnson, R. Tateishi, and Z. Xie, “Using geographically weighted variables for image classification,” *Remote Sens. Lett.*, vol. 3, no. 6, pp. 491–499, 2012.
[44] V. Sindhwani, P. Niyogi, and M. Belkin, “Beyond the point cloud: From transductive to semi-supervised learning,” in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 824–831.
[45] D. Greene and P. Cunningham, “Practical solutions to the problem of diagonal dominance in kernel document clustering,” in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 377–384.
[46] D. Greene and P. Cunningham, “A matrix factorization approach for integrating multiple data views,” in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2009, pp. 423–438.

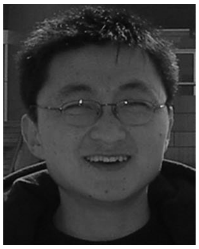


Wenzhang Zhuge received the BS degree from Shandong University, Jinan, China, in 2015 and the MS degree from the National University of Defense Technology, Changsha, China, in 2017. He is currently working toward the PhD degree at the National University of Defense Technology, Changsha, China. His research interests include machine learning, system science, and data mining.

Hong Tao received the PhD degree from the National University of Defense Technology, Changsha, China, in 2019. She is currently a lecturer with the College of Liberal Arts and Science, National University of Defense Technology, China. Her research interests include machine learning, system science, and data mining.

Tingjin Luo received the BS, master's, and PhD degrees from the National University of Defense Technology, Changsha, China, in 2011, 2013 and 2018, respectively. He is currently a lecturer with the College of Science, National University of Defense Technology, Changsha, China. He was a visiting PhD student with the University of Michigan, Ann Arbor, Michigan, from 2015 to 2017. He has authored more than 15 papers in journals and conferences, such as the *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Image Processing* and KDD etc. He has been a program committee member of several conferences including IJCAI, AAAI etc. His research interests include machine learning, multimedia analysis, optimization, and computer vision.

Ling-Li Zeng received the BSc, MSc, and PhD degrees from the National University of Defense Technology, China, in 2007, 2009, and 2014, respectively. From November 2012 to November 2013, he was a visiting PhD student with the Harvard Medical School and Massachusetts General Hospital, Boston, Massachusetts. He is currently an associate professor with the College of Mechatronics and Automation, National University of Defense Technology, China. He has authored several papers in journals, such as the *Proceedings of the National Academy of Sciences of the United States of America*, *Brain*, *Human Brain Mapping*, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, etc. His research interests include cognitive neuroscience, image processing, and pattern recognition in neuroimaging.



Chenping Hou (Member, IEEE) received the PhD degrees from the National University of Defense Technology, Changsha, China, in 2009. He is currently a full professor with the Department of Systems Science, National University of Defense Technology, Changsha, China. He has authored more than 80 peer-reviewed papers in journals and conferences, such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Neural Networks and Learning Systems*, IEEE TSMCB/TCB, *IEEE Transactions on Image Processing*, IJCAI, and AAAI. His current research interests include machine learning, data mining, and computer vision.

Dongyun Yi received the BS degree from Nankai University, Tianjin, China, and the MS and PhD degrees from the National University of Defense Technology, Changsha, China. He was a visiting researcher with the University of Warwick, Coventry, United Kingdom, in 2008. He is a professor with the College of Science, National University of Defense Technology, China. His current research interests include statistics, systems science, and data mining.