Multiview Clustering via Unified and View-Specific Embeddings Learning

Qiyue Yin, Shu Wu, *Member, IEEE*, and Liang Wang, *Senior Member, IEEE*

***Abstract*— Multiview clustering, which aims at using multiple distinct feature sets to boost clustering performance, has a wide range of applications. A subspace-based approach, a type of widely used methods, learns unified embedding from multiple sources of information and gives a relatively good performance. However, these methods usually ignore data similarity rankings; for example, example A may be more similar to B than C, and such similarity triplets may be more effective in revealing the data cluster structure. Motivated by recent embedding methods for modeling knowledge graph in natural-language processing, this paper proposes to mimic different views as different relations in a knowledge graph for unified and view-specific embedding learning. Moreover, in real applications, it happens so often that some views suffer from missing information, leading to incomplete multiview data. Under such a scenario, the performance of conventional multiview clustering degenerates notably, whereas the method we propose here can be naturally extended for incomplete multiview clustering, which enables full use of examples with incomplete feature sets for model promotion. Finally, we demonstrate through extensive experiments that our method performs better than the state-of-the-art clustering methods.**

***Index Terms*— Incomplete multiview data, knowledge graph embedding, multiview learning, subspace learning.**

I. Introduction

IN REAL applications, it happens so often that the data consist of multiple distinct feature representations, which we call multiview data, each view indicating a feature set. For example, an image can be represented by its color and shape descriptors, and a webpage by using images, texts, and hyperlinks. Usually, these multiple sources of information encode complementary information, which motivates the development of multiview learning whose goal is to explore such information for a better performance [1]–[3]. Multiview learning is widely studied for a variety of applications, e.g., image processing, data mining, and multimedia [4]–[7].

When multiview learning has to deal with clustering tasks, multiview clustering provides a natural way to organize data with multiple feature sets [8], [9]. To explore complementary information between different views, many promising methods have been developed, which can be roughly divided into four major categories [1], [10]–[12]. The first category methods are subspace-based [13]–[21], which learn a unified embedding for final clustering. The second category methods are cotraining-based, with typical examples, such as [22]–[24]. The third category methods are late fusion-based [25]–[27], which obtain final clustering by combining the results from each view. The fourth category methods learn optimal similarity matrix to reflect data cluster structure, which serves as an affinity matrix for spectral clustering [28]–[30]. Section II presents more details in this regard.

Among the diverse multiview clustering methods, we focus on the subspace-based ones, which are widely studied and generally give a good performance. Based on the technique used for obtaining low-dimensional embedding, these methods can be divided into four typical classes. The first one is based on the canonical correlation analysis [13], [31], [32], which aims at finding linear projections of different views with maximal mutual correlation. The second class is based on spectral analysis [14], [30], [33]. Kumar *et al.* [14] proposed two methods to regularize spectral embedding in such a way that each view is similar to the other for final clustering. The third one uses matrix factorization to obtain an embedding. Usually, nonnegative matrix factorization (NMF) [15], [34], [35] is used. The last one uses a regression-based objective to obtain such an embedding that is designed to approximate the scaled indicator matrix [16], [17], [36].

For the above-mentioned embedding learning approaches, the former two kinds of subspace-based learning methods use a covariance matrix or an affinity matrix, while the latter two types factorize the feature sets or directly project different views to the desired subspace. To the best of our knowledge, all the above-mentioned subspace-based methods ignore data similarity rankings; for example, example A may be more similar to B than C, which may better reflect the clustering structure. The similarities between any two examples can be utilized for embedding learning, such as

Manuscript received October 24, 2016; revised June 29, 2017 and October 20, 2017; accepted December 12, 2017. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001000, in part by the National Natural Science Foundation of China under Grant 61772528, Grant 61525306, Grant 61633021, Grant 61572504, and Grant 61420106015, in part by the Strategic Priority Research Program of the CAS under Grant XDB02070001, and in part by the Beijing Natural Science Foundation under Grant 4162058. *(Corresponding author: Liang Wang.)*

Q. Yin and S. Wu are with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and also with the University of Chinese Academy of Sciences, Beijing, China (e-mail: qyyin@nlpr.ia.ac.cn; shu.wu@nlpr.ia.ac.cn).

L. Wang is with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, also with the Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and also with the University of Chinese Academy of Sciences, Beijing, China (e-mail: wangliang@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2786743

isomap [37], locally linear embedding [38], and the Laplacian eigenmaps [39], but these approaches require exact similarity values, which are harder to be accurately obtained than that of partial similarity triplets. Moreover, some studies were carried out for leaning embedding of data points, based on such triplets, achieving good results in such areas as crowd sourcing and image classification [40]–[42]. However, such triplets are obtained based on supervised information, which are not accessible in unsupervised clustering tasks. More importantly, previous methods focus mainly on embedding of a single view as they cannot deal with multiple views, possibly because contrasting triplets might appear among different views.

Recently, the embedding methods designed for modeling the knowledge graph in natural-language processing have become popular [43]–[45], which inspires us to learn the embedding of similarity triplets from multiple views. Given the knowledge graph, with each entity showing abstract and directed links representing different types of relations, those methods usually learn unified embedding of all entities and use several matrices or vectors to change that embedding for relation-specific embedding. In the knowledge graph, the entity appears in different *relations*, e.g., California *contains* Los Angeles, and California *is located in* North America. Similarly, in multiview data, example A may appear in similarity triplets of different *views*; for example, example A is more similar to B than other examples *in one view* and example A is more similar to C than other examples *in another view*. By contrasting these two scenarios and borrowing the idea behind the modeling relations of the knowledge graph, we can cluster data with multiple views.

In this paper, motivated by the recent embedding methods adopted for modeling the knowledge graph, we propose here a novel subspace-based multiview clustering method. We learn unified embedding for all examples, based on similarity triplets, calculated from multiple views. To fit the triplets produced from each view, view-specific embeddings are learned through some basic matrices and view-specific vectors, imposed on the unified embedding. By doing so, we establish connection between the unified embedding and view-specific embeddings and meanwhile explore multiview characteristics. In real applications, we may confront the problem of missing information in a multiview data set, as some examples may have incomplete feature sets. We show that the method we propose here can be extended even to such incomplete multiview clustering, which is important but not studied much by previous works. We demonstrate through extensive experiments that our method outperforms the state-of-the-art multiview clustering methods.

The main contributions of this paper are as follows.

1) Motivated by recent embedding methods designed for modeling knowledge graph, we propose, probably for the first time, to learn embeddings from similarity triplets, calculated from multiple views, for the task of multiview clustering.
2) We learn unified embedding and several view-specific embeddings for multiview data and design their relations appropriately, based on multiview characteristics, i.e., multiple views, describing the same content, share only partial characteristics of data.
3) The proposed method is more practical in real applications and can be extended even to incomplete multiview clustering, which involves dealing with data of incomplete feature sets, a scenario not studied much by previous researchers.
4) Extensive experiments are conducted, using the method we propose, for complete and incomplete multiview clustering, achieving a better performance than that of the state-of-the-art methods.

The remainder of this paper is organized as follows. Section II briefly reviews related works; Section III elaborates the proposed multiview clustering algorithm; Section IV presents the experimental results and analysis, and finally, Section V sums up the conclusions drawn from this paper.

II. Related Work

In this section, we briefly review works relating to multiview clustering, embedding for similarity triplets, and embedding for the knowledge graph.

A. Complete and Incomplete Multiview Clustering

1) Complete Multiview Clustering: Multiview clustering, which exploits complementary characteristics between multiview data sets for better clustering, can be roughly classified into four categories based on when multiple sources of information are utilized [1], [10], [11]. The first category methods are subspace-based [13]–[17], [46], which learn unified embedding, irrespective of the view. Usually, these methods are based on spectral analysis and matrix factorization through some regularization techniques. The second category methods integrate multiple sources of information during the clustering process [22]–[24]. As one of the most popular semisupervised tools, cotraining framework is used to complete the clustering process. The third category methods obtain final clustering by combining individual results from each view through late fusion [25]–[27]. The last category methods learn unified similarity matrix from multiview data, which serves as an affinity matrix for final spectral clustering [28]–[30], [47]–[49]. Some recently proposed methods [28], [30], [50] are extensions of single view subspace segmentation methods.

Among the various multiview clustering methods, subspace-based ones, which utilize various kinds of techniques for low-dimensional embedding learning, are the most widely studied. Usually, they are easy to explain and can reduce the dimensionality of original data. Because of this property, we based our method on subspace learning. The cotraining framework is popular for semisupervised classification, which needs strong assumptions [10], such as sufficiency, compatibility, and conditional independence, for its success. If these are not satisfied, good clustering results may not be guaranteed. Unlike subspace learning, which learns feature from different views, late fusion-based methods obtain the final results in a decision-level fusion. So, such methods mainly rely on clustering results from each view. Unified similarity matrix learning methods are similar to multiple

kernel learning [10]. The recently proposed spectral subspace segmentation-based methods usually give a good performance, but their computation cost is rather high. Besides, with a large number of examples, the data representation becomes more voluminous than the original space.

2) Incomplete Multiview Clustering: Usually, the existing multiview clustering methods assume that all the examples have complete feature sets. However, in real applications, it so happens that some examples lose certain feature sets, which are called incomplete multiview data sets. A naive approach to deal with such data sets is to remove the examples with incomplete feature sets and use only the examples with complete views for model training. However, such preprocessing will lead to loss of information, which is found to be useful for multiview clustering [35].

Recently, a few studies have been carried out, focusing on incomplete multiview clustering, which can be classified into two major categories. The first category methods preprocess incomplete views by filling missing information. Rai *et al.* [51] propose to use the kernel matrix of a complete view to fill the kernel matrices of incomplete views, through the Laplacian regularization, which can deal with scenarios in which at least one view is complete. Shao *et al.* [52] improved on this approach [51] by dealing with scenarios in which no views are complete. Unfortunately, both methods are kernel matrix-based and thus can only be used for kernel-based multiview clustering. The second category methods need no preprocessing of missing information as they directly carry out multiview clustering. Recently, Li *et al.* [35] and Shao *et al.* [53] proposed subspace-based methods, which learn low-dimensional embedding of incomplete multiview data through NMF and obtain a better performance, compared with that of the first category methods. However, NMF cannot be used for examples with negative features. More recently, Xu *et al.* [54] developed a matrix completion-based incomplete multiview learning method, which is effective in restoring the missing variables and obtaining unified embedding. However, this method cannot effectively explore the multiview complementarity and consistency.

In this paper, a novel subspace learning-based method is proposed, which can reveal the structure of data cluster better than the existing subspace-based methods. Besides, it can be a natural extension to incomplete multiview clustering.

B. Embedding for Knowledge Graph

The knowledge graph is a directed graph whose nodes represent the entities and edges of different types of relations [43], [55]. Usually, the goal of modeling such multirelational data is to discover connectivity patterns between entities, so as to predict their relations and then find new relational facts, which play an important role in various areas, such as link prediction and natural language understanding. Recently, some promising types of approaches have been developed to embed the knowledge graph into a continuous vector space, while preserving its structure [43], [44], [56]. Bordes *et al.* [55] designed two relation-specific matrices that can adjust to different types of relations. Socher *et al.* [45] modified this design by taking into consideration the second-order correlations between entity embeddings. Recently, apart from learning embedding for entities, Bordes *et al.* [43] learned embedding for each type of relation in the same space as that of entities. Lin *et al.* [44] modified this model by separately building entity in entity space and relation embedding in relation space.

Generally, all the above-cited methods use some relation-specific matrices or vectors to change the unified embedding of directed entities, so that they can adjust to different relations connecting them. Intuitively, by operating the embedding of an entity, a new embedding is learned for such entity, under a specific relation. Besides, to adjust to different types of relations, they make no constraints on the matrices. What is interesting to us are the nonconstraint matrices used for operating embeddings. In our model, we can mimic such an idea to learn unified- and view-specific embeddings of complete or incomplete multiview data for clustering. For learning embeddings of the knowledge graph, we are given a known directed and unweighted graph. However, in the clustering task, we may construct multiple undirected and weighted graphs indirectly. Thus, for mimicking embedding methods of multirelational data for clustering, the above-mentioned problems still remain to be solved.

C. Embedding for Similarity Triplets

Following rapid developments in crowd sourcing, human similarity judgment has gained popularity, e.g., example A is more similar to B than C. Thus, a variety of machine-learning techniques have been used for embedding of such similarity triplets, which provides a chance to learn embeddings that put similar data close and dissimilar examples far away. Recently, various methods have been proposed to deal with the comparison of triplets, obtained from a single view [40]–[42]. Agarwal *et al.* [40] resorted to nonmetric multidimensional scaling to find a low-rank kernel matrix, so that pairwise distances between embeddings satisfy the observed triplets. Maaten and Weinberger [42] developed a technique called t-distributed stochastic triplet embedding to collapse similar points and repel dissimilar points in the embedding space.

Recently, McFee and Lanckriet [57] used multiple kernel learning to learn unified embedding of similarity triplets. Zhang *et al.* [58] explored correlations between views and proposed to learn view-specific embeddings. However, both these methods are not designed for unsupervised tasks. More importantly, in [57], we use a tensor and several vectors, rather than weights, to adjust the importance of each view for obtaining view-specific embeddings that can adapt to view-specific similarity triplets. And, rather than [58] using the Mahalanobis distance to adapt to view-specific embedding, we follow no constraints, and thus obtain a fewer regularization parameters and a good interpretability. In summary, for unsupervised multiview clustering task, ours is probably the first attempt to fuse embedding methods for similarity triplets and the knowledge graph.

III. Model

Given a data set $\mathbf{X} = \{\mathbf{x}_k\}_{k=1}^n$ with n examples, we use $\mathbf{X}^v = \{\mathbf{x}_k^v\}_{k=1}^n \in \Re^{d_v \times n}, (v = 1, \ldots, l)$ to denote samples in

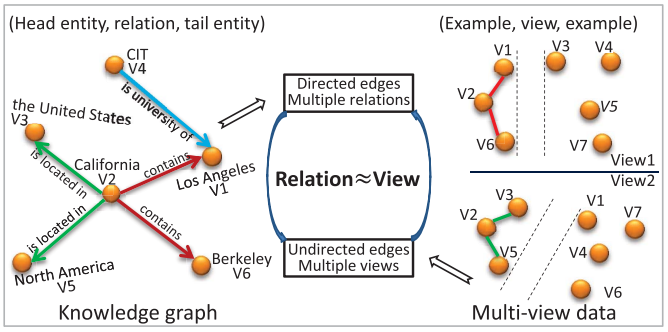


Fig. 1. Comparison of multiview data and the knowledge graph. Given a knowledge graph, the entity California *contains* Los Angeles and Berkeley, in relation *location_contains_location*, can be seen as the multiview data example V2, which is more similar to V1 and V6 than to other examples in *View 1*. By regarding a *relation* as a *view*, we establish the connection between the knowledge graph and multiview data.

the vth view, with d_v as the feature dimensionality. Our task here is to cluster data set $\mathbf{X}$ into the predefined c classes.

Before elaborating our approach, we assume that modeling multiview data characteristics, i.e., consistency and complementarity, can well reflect the data, and it is helpful for multiview clustering as least in most cases. More specifically, we aim at learning unified and view-specific embeddings revealing the triplet structure of multiview data, i.e., example A is more similar to B than C. By doing so, we can learn embeddings that put similar data close and dissimilar data away, providing a better way to reveal the group structure of data than the above-mentioned subspace-based methods. Moreover, the multiview data characteristics can be well explored.

A. Comparison Between Knowledge Graph and Multiview Data Embedding

The knowledge graph, which uses the directed graph to model entities and their relations, provides a natural way for describing multirelational data. Among the various learning methods for modeling the knowledge graph, recent embedding-based methods give promising results in various applications. Before presenting our model, we briefly introduce those representative embedding methods. Given a knowledge graph, as shown in the left part of Fig. 1, those methods usually learn unified embedding for all entities to reveal the structure of the graph. As an entity may have multiple relations with other entities, e.g., California *contains* Los Angeles, and California *is located in* North America, those methods learn relation-specific embeddings to adapt to each relation, e.g., two embeddings for the entity California under the above-mentioned two relations.

Intuitively, the scenario of multiview clustering is very similar to that of knowledge graph embedding, wherein a view is treated as a type of relation, as shown in Fig. 1. More specifically, *in knowledge graph modeling, an entity (such as California), which is connected with some entities (like Los Angeles and Berkeley) in a relation (contains), can also be linked to other entities (like the USA and North America) in another relation (is located in). Similarly, in multiview clustering, an example (such as V2) is more similar to some examples (like V1 and V6) in one view (view 1), while they can also be similar to several examples (like V3 and V5) in another view (view 2)*. For multiview clustering, we contrast these two scenarios and resort to the embedding methods, designed for modeling the knowledge graph.

Based on the above-mentioned comparison, we borrow the idea behind knowledge graph embedding to learn unified and view-specific embeddings for multiview data, from which the multiview characteristics are explored, and the embeddings are accordingly expected to well reflect the data cluster structure.

B. Formulation

As has been mentioned in Section I, similarity triplets can better reveal the data cluster structure; so, we make the learned embeddings a good reveal of such triplets. As no similarity triplets are given to us, we need to construct them, which should be accurate at least in part.

Fortunately, given an example under a specific view, it may be simple to determine a few examples that are similar to this example and possibly to many other examples that are dissimilar to this example. By imposing a similarity metric, say the Euclidean or Cosine distance, on a feature matrix, we can regard the K nearest neighbors of an example as similar data and those far away from the example as dissimilar ones. Then, the similarity triplets set, calculated from the vth view $\mathcal{S}^v$, is obtained by the following equation:

$$\mathcal{S}^v = \{(\mathbf{x}_i^v, \mathbf{x}_j^v, \mathbf{x}_k^v) | i = \{1:n\}; \mathbf{x}_j^v \in \mathcal{S}_+^v(\mathbf{x}_i^v); \mathbf{x}_k^v \in \mathcal{S}_-^v(\mathbf{x}_k^v)\} \quad (1)$$

where the examples of the K nearest neighbors of $\mathbf{x}_i^v$, and those far away from $\mathbf{x}_i^v$, consist of positive set $\mathcal{S}_+^v(\mathbf{x}_i^v)$ and negative set $\mathcal{S}_-^v(\mathbf{x}_i^v)$, respectively. For simplicity, we can select a fixed number of examples to construct $\mathcal{S}_-^v(\mathbf{x}_i^v)$, say, half of the total examples in the data set.

Generally, using the above-mentioned equation, we can construct many similarity triplets, which may provide a chance to learn an embedding that can reveal the structure of data cluster better than the existing subspace-based methods. After obtaining similarity triplets of different views, we aim at learning view-specific embeddings to reveal such data cluster structure, and the formulation is written thus

$$\min_{\mathbf{E}} \sum_i \sum_v \sum_{\left(\mathbf{x}_j^v, \mathbf{x}_k^v\right)} \ell(\mathrm{dis}(\psi^v(\mathbf{e}_i), \psi^v(\mathbf{e}_j)), \mathrm{dis}(\psi^v(\mathbf{e}_i), \psi^v(\mathbf{e}_k)))$$
$$\text{s.t.} \quad ||\mathbf{e}_t|| = 1, \quad \forall t \quad (2)$$

where dis is a distance function, ℓ is a loss function, measuring the embeddings of a triplet $(\mathbf{x}_i^v, \mathbf{x}_j^v, \mathbf{x}_k^v)$, and $\mathbf{x}_j^v \in S_+^v(\mathbf{x}_i^v)$ and $\mathbf{x}_k^v \in \mathcal{S}_-^v(\mathbf{x}_k^v)$. $\mathbf{E} \in \Re^{d \times n}$ is the unified embedding with dimensionality d, and $\mathbf{e}_i$ is the unified embedding for the ith example. We impose the normalization constraint on each column of $\mathbf{E}$, which helps remove the scaling freedom from the model. ψ^v is a function that models the relation between the unified embedding and the embedding in the vth view.

Because different views consisting of common and view-specific characteristics represent the same content, it will be appropriate to generate view-specific embeddings through

several basic matrices, imposed on the unified embedding. This way, the unified embedding is expected to give an integrated representation of the content, and the view-specific embeddings are forced to reflect the view-specific characteristics. Moreover, the connection between the unified and view-specific embeddings is established to explore the multiview characteristics. The formulation is as follows:

$$\psi^v(\mathbf{e}_i) = \mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_i = \begin{bmatrix} \mathbf{s}^v\mathbf{M}^{[1]}\mathbf{e}_i \\ \cdots \\ \mathbf{s}^v\mathbf{M}^{[d]}\mathbf{e}_i \end{bmatrix} \tag{3}$$

where $\mathbf{M}^{[1:d]}$ is a $u \times d \times d$ tensor, $\mathbf{M}^{[k]} \in \Re^{u\times d}$, serving as a basic matrix, is the kth slice of the tensor, and $\mathbf{s}^v$ is a u-dimensional latent vector. With this formulation, we can explicitly model complementarity through several basic matrices, i.e., the tensor $\mathbf{M}$ shared by all the views. Considering that different views represent partial characteristics of the examples, view-specific vector $\mathbf{s}^v$ is imposed on the shared tensor for view-specific operator matrix and then the view-specific embedding obtained by changing the unified embedding $\mathbf{e}_i$.

It should be noted that several view-specific matrices $\mathbf{P}^v \in \Re^{d\times d}$ can be utilized as an alternative of $\mathbf{s}^v$ and $\mathbf{M}$, but the purpose to explicitly explore multiview characteristics cannot be achieved, because we cannot explicitly model the relation between those view-specific matrices.

A good choice for ℓ is a margin loss, which is widely used in various algorithms, such as support vector machine. Besides, the distance function can be a p-norm distance, which is selected as the widely utilized Euclidean distance here. Hence, we have

$$\begin{aligned}&\ell(d(\psi^v(\mathbf{e}_i), \psi^v(\mathbf{e}_j)), d(\psi^v(\mathbf{e}_i), \psi^v(\mathbf{e}_k))) \\ &= \max(||\psi^v(\mathbf{e}_i) - \psi^v(\mathbf{e}_j)||^2 + \gamma - ||\psi^v(\mathbf{e}_i) - \psi^v(\mathbf{e}_k)||^2, 0)\end{aligned} \tag{4}$$

where γ is a parameter controlling the margin size.

Finally, the overall objective is

$$\begin{aligned}\min_{\mathbf{E},\mathbf{s},\mathbf{M}} &\sum_i \sum_v \sum_{\left(\mathbf{x}_i^v,\mathbf{x}_j^v,\mathbf{x}_k^v\right)\in\mathcal{S}^v} \max(||\mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_i - \mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_j||^2 \\ &\qquad + \gamma - ||\mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_i - \mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_k||^2, 0) \\ \text{s.t.}\quad &||\mathbf{e}_t||_2 = 1, \quad \forall t\end{aligned} \tag{5}$$

where all the variables are the same as in (1)–(4).

C. Extension to Incomplete Multiview Clustering

In real applications, it happens so often that some examples lose several feature sets, resulting in incomplete multiview data. Since traditional multiview clustering algorithms often assume complete views, they may be unable to effectively deal with such data. In such a situation, the major challenge for model training is how to make full use of the examples with incomplete views. Fortunately, various approaches have been developed to model samples with complete views; so, examples with incomplete feature sets can be combined to enhance the learning process. In this section, we will show how our model can be extended seamlessly to incomplete multiview clustering.

While dealing with incomplete multiview data, not all examples appear in every views. Instead of modeling a point in each view as in (5), we can model the triplets constructed from each view. This way, all the samples are utilized whether they consist of complete views or not. The objective is written as

$$\begin{aligned}\min_{\mathbf{E},\mathbf{s},\mathbf{M}} &\sum_v \sum_{(\mathbf{x}_i^v,\mathbf{x}_j^v,\mathbf{x}_k^v)\in\mathcal{S}_{in}^v} \max(||\mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_i - \mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_j||^2 \\ &\qquad + \gamma - ||\mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_i - \mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_k||^2, 0) \\ \text{s.t.}\quad &||\mathbf{e}_t||_2 = 1, \quad \forall t\end{aligned} \tag{6}$$

where $(\mathbf{x}_i^v, \mathbf{x}_j^v, \mathbf{x}_k^v)$ is calculated using examples appearing in the vth view. Because of incomplete setting, not all the examples appear in the vth view, i.e., $\mathcal{S}_{in}^v = \{(\mathbf{x}_i^v, \mathbf{x}_j^v, \mathbf{x}_k^v)|i = \{1 : n_v\}; \mathbf{x}_j^v \in \mathcal{S}_+^v(\mathbf{x}_i^v); \mathbf{x}_k^v \in \mathcal{S}_-^v(\mathbf{x}_k^v)\}$, where n_v is the number of examples appearing in the vth view.

Overall, our model can make full use of the incomplete multiview data, regardless of whether the views of the samples are complete or incomplete, for the following reasons: 1) we establish the relation between unified embedding and incomplete view-specific embeddings, which is much more complex than in the complete view setting and 2) tensor $\mathbf{M}$ is learned, based on all samples, whether the example views are complete or incomplete, and view-specific vector $\mathbf{s}^v$ is learned, using all the examples in the vth view.

It should be noted that we learn unified embedding for all the data and establish the relation between unified embedding and view-specific embeddings, through (3); so, given an example that does not appear in the vth view, we can still obtain its embedding in view v by just using (3). This is interesting, because we can fill up the missing embedding information directly.

D. Optimization

Given a multiview data set, we construct similarity triplets, denoted as $\{\mathcal{S}^v, v = 1, \ldots, l\}$. As the sizes of these sets are very large because of their construction process, calculation of gradients, based on all the similarity triplets, may be time-consuming. Furthermore, as different types of variables are to be coupled, it may be difficult to optimize all the variables simultaneously. Therefore, the stochastic gradient descent method (in minibatch mode) is used to optimize the variables $\mathbf{E}$, $\mathbf{s}^v$, and $\mathbf{M}$ iteratively.

As for the constraint imposed on each column of the unified embedding, we just normalize the columns at each updating iteration [44], [55]. The objective in (6) is denoted as L. Then, the gradient of each variable, under a specific triplet $(\mathbf{x}_i^v, \mathbf{x}_j^v, \mathbf{x}_k^v)$ is calculated as follows. If $||\mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_i - \mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_j||^2 + \gamma - ||\mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_i - \mathbf{s}^v\mathbf{M}^{[1:d]}\mathbf{e}_k||^2 <= 0$, there will be no gradients [1]; otherwise, the gradients are as follows:

$$\partial L/\partial \mathbf{e}_i = 2(\mathbf{s}^v\mathbf{M}^{[1:d]})^T\mathbf{s}^v\mathbf{M}^{[1:d]}((\mathbf{e}_i - \mathbf{e}_j) - (\mathbf{e}_i - \mathbf{e}_k)) \tag{7}$$

$$\partial L/\partial \mathbf{e}_j = -2(\mathbf{s}^v\mathbf{M}^{[1:d]})^T\mathbf{s}^v\mathbf{M}^{[1:d]}(\mathbf{e}_i - \mathbf{e}_j) \tag{8}$$

$$\partial L/\partial \mathbf{e}_k = 2(\mathbf{s}^v\mathbf{M}^{[1:d]})^T\mathbf{s}^v\mathbf{M}^{[1:d]}(\mathbf{e}_i - \mathbf{e}_k) \tag{9}$$

[1]When the value is zero, the gradient is not accessible, and some smooth techniques can be utilized [59], However, in the batch-based stochastic gradient descent method, we find that a zero gradient is fine as adopted in [60].

$$\partial L/\partial \mathbf{s}^v = 2\mathbf{s}^v\mathbf{G}\mathbf{G}^T - 2\mathbf{s}^v\mathbf{H}\mathbf{H}^T \tag{10}$$

$$\partial L/\partial \mathbf{M}^{[p]} = 2(\mathbf{s}^v)^T\mathbf{s}^v\mathbf{M}^{[p]}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T - 2(\mathbf{s}^v)^T\mathbf{s}^v\mathbf{M}^{[p]}(\mathbf{e}_i - \mathbf{e}_k)(\mathbf{e}_i - \mathbf{e}_k)^T \tag{11}$$

where $\mathbf{G}$ and $\mathbf{H}$ are two matrices, their pth columns being $\mathbf{M}^{[p]}(\mathbf{e}_i - \mathbf{e}_j)$ and $\mathbf{M}^{[p]}(\mathbf{e}_i - \mathbf{e}_k)$, respectively, and $\mathbf{M}^{[p]}$ is the pth slice of the tensor $\mathbf{M}^{[1:d]}$.

Algorithm 1: Unified and View-Specific Embeddings Learning

Input: Multiview data set $\mathbf{X}$; Parameters K, γ and d
Output: Latent embedding $\mathbf{E}$ for $\mathbf{X}$

```
Calculate similarity triplets using Equation 1;
Initialize E uniformly from (-1,+1), s^v M^[1:d] = I;
while not converge do
    Enforce the constraints ||e_i||_2 = 1, ∀i;
    Randomly sample part of triplets to construct a batch
    (even distributed from all the views);
    Update all variables through gradients, calculated
    from Equations (7) to (11);
return E.
```

Based on the above-mentioned gradients, the optimization is summarized in Algorithm 1. First, we calculate similarity triplets for each view, regardless of whether the view is complete or not. With the initialized variables, we update them, based on the gradients calculated from (7) to (11). It is to be noted that in each iteration, we normalize the unified embedding $\mathbf{E}$ in such a way that it lies on a unit circle or on the surface of a hypersphere as in [44] and [55].

It should be mentioned here that we uniformly sample all the triplets from all the views to construct a batch for each updating iteration. However, in this process, if a view loses too many examples, a few triplets will be selected, failing which this view loses much of its information and hence cannot be considered as good complementary information for parameters learning. Besides, in the above-mentioned scenario, we can reduce the size of K, when constructing triplets of this view to alleviate performance degeneration.

After optimization, we can obtain unified embedding $\mathbf{E}$ for final clustering. For simplicity, the k-means is used to cluster the unified embedding $\mathbf{E}$. Finally, the overall clustering procedure is summarized in Algorithm 2.

Algorithm 2: Multiview Clustering Via Unified And View-Specific Embeddings Learning

Input: Multiview data set $\mathbf{X}$; Parameters K, γ and d.
Output: Groups of the multiview data set.

```
Calculate the unified embedding E of multiview data set
using Algorithm 1
Perform k-means clustering algorithm on E;
return groups of the data sets.
```

E. Algorithm Analysis

For optimization, we use stochastic gradient descent algorithm, which guarantees convergence with some clues, e.g., the carefully selected learning rate. Although normalization is imposed on the embeddings, it does not influence convergence, and the same has been verified in various knowledge graph embedding algorithms [43], [44], [55].

In our proposed model, the number of parameters used is $O(nd + ud^2 + lu)$, where n is the size of data set, d is the dimensionality of the embedding, l is the number of views, and u is the dimensionality of the latent vector, encoding view-specific information. Generally, the size of parameters is about nd, considering that l, d, and u are much smaller than the data size. So, the parameter size is nearly linear with respect to the size of data set. Such a small size of parameters is expected to be scalable to large-scale problems, as proved in [43].

IV. Experiments

This section deals with the experiments conducted extensively to validate the effectiveness of our model.

A. Data Sets

In this section, we report the experimental results on six kinds of public databases and summarize their information in Table I.

1) US Postal Service Data Set[2]*:* This data set consists of features of handwritten numerals, extracted from a collection of Dutch utility maps. There are 2000 samples, uniformly distributed in 10 categories, and each example is encoded in terms of six types of features. For performance validation, we use only 76 Fourier coefficients of the character shapes and 216 profile correlations as two views on the same lines as [14].

2) 3Source Data Set[3]*:* This data set is constructed using three well-known online news sources, i.e., BBC, Reuters, and the Guardian. It includes a total of 416 distinctly new items, divided into six categories. Of them, 169 news are reported by all the three sources and are used, as in [15], with each source serving as one view. Besides, the feature used is word frequency for all the three views.

3) Cora Data Set[4]*:* This is a document data set with a total of 2708 documents of seven classes (Neural_Networks, Rule_Learning, Reinforcement_Learning, Probabilistic_ Methods, Theory, Genetic_Algorithms, and Case_Based). Two feature sets, i.e., citations and content, are used as two views, where the content feature is 0/1 valued word vector, indicating the absence/presence of corresponding words.

4) BBC Data Set[5]*:* This data set is a synthetic multiview text database constructed by using single-view BBC and BBCSport corpora. It includes a total of 2012 examples, divided into five categories. Segment representations of the same text are used here as two views, their dimensionalities being 6838 and 6790. The data are preprocessed using principal component analysis and the dimensionality is carefully selected, based on the eigenvalues of the covariance matrix obtained from the data.

[2]http://archive.ics.uci.edu/ml/data sets/Multiple+Features.
[3]http://mlg.ucd.ie/data sets/3sources.html.
[4]http://lig-membres.imag.fr/grimal/data.html.
[5]http://mlg.ucd.ie/datasets/segment.html.

TABLE I
INFORMATION OF MULTIVIEW DATA SETS USED IN THE EXPERIMENTS

Dataset	USPS	3Source	Cora	BBC	CCV	VOC
# size	2,000	169	2,708	2,012	9,317	9,963
# view	2	3	2	2	2	2
# cluster	10	6	7	5	20	20
# feature size	76+216	3,560+3,631+3,068	2,708+1,433	6,838+6,790	5,000+5,000	399+512

5) CCV Data Set[6]*:* This is a Columbia Consumer Video database containing 9317 YouTube videos, categorized into 20 semantic classes. Two kinds of features, i.e., scale-invariant feature transform and space-time interest points, are used here for performance validation. Some of the videos are multilabeled; so, we select only those with one label, and thus we are finally left with 6743 examples.

6) Pascal Visual Object Classes (VOC) 2007 Data Set[7]*:* This consists of 9963 images, divided into 20 categories. For performance evaluation, we use 512 dimensional GIST features and 399 dimensional tag feature as two views. Besides, images with multiple labels and those whose tag features are all zeros are deleted. Thus, we are left with 5619 examples.

B. Compared Methods and Experimental Settings

Because our approach is based on the subspace learning framework, we compare our method with representative multiview subspace learning, which mainly includes four typical techniques, presented in Section I, i.e., canonical correlation analysis (CCA), PairwiseSC, CenroidSC, partial multi-view clustering (PVC), and MultiCF. These methods are described in the following in detail.

1) *SingleV:* We run spectral clustering [61] on all the views and report the best performance among them.
2) *CCA:* We use the canonical correlation analysis to obtain low-dimensional embedding of multiview data and then apply k-means on the embedding for final clustering.
3) *PairwiseSC:* For final clustering, we first regularize the spectral embeddings of all the views to be similar and then perform k-means on one of the embeddings, as proposed by Kumar *et al.* [14].
4) *CentroidSC:* Kumar *et al.* [14] proposed to regularize the spectral embeddings of all views to be similar toward a unified spectral embedding and performed k-means on the unified embedding for final clustering.
5) *MultiCF:* We approximate the scaled indicator matrix, using the structure sparsity-based unsupervised feature selection method, as proposed by Wang *et al.* [16].
6) *Robust Multi-View Spectral Clustering (RMSC):* We learn the unified transition matrix, based on sparse and low-rank constraints, as proposed by Xia *et al.* [30] and then use spectral analysis for final clustering.
7) *Partial multi-View Clustering (PVC):* We use the NMF-based method, proposed by Li *et al.* [35] in dealing with complete and incomplete multiview clustering.
8) *MultiTE:* Our proposed **multi**view clustering uses the unified **e**mbedding of similarity **t**riplets.
9) *SpeMultiTE:* As one baseline, we implement MultiTE by forcing unified and view-specific embeddings to be the same.
10) *SingleTE:* As another baseline, we implement MultiTE using only one view and report the best performance among all the views.

For PairwiseSC, CentroidSC, RMSC, and PVC methods, we use the codes released by their authors to achieve the best performance. To implement the CCA method, we use the code LSCCA package.[8] For implementing the MultiCF method, we follow the authors' suggestions to obtain clustering results. For our method, the sizes of positive and negative sets in (1) are empirically selected as 10 and half the number of total examples, respectively, based on the Euclidean distance between the data sets. As for the parameters γ and d, we control the margin size of our loss function and the dimensionality of unified embedding by empirically selecting them to be 5 and 30, respectively, in all the experiments. As regard to the dimensionality of $\mathbf{s}^v$, we empirically set it to be the number of views. As k-means algorithm is used in all the experiments, it is run 20 times with random initialization, and the mean value and the standard deviation are reported.

C. Evaluation Metrics

Two widely used measures, namely, the accuracy (ACC) and the normalized mutual information (NMI), are used for performance evaluation

$$\mathrm{NMI} = \frac{\mathrm{I(CAT; CLS)}}{\sqrt{\mathrm{H(CAT)H(CLS)}}} \tag{12}$$

where CAT and CLS are true labels and cluster labels, respectively. I(CAT; CLS) is the mutual information between CAT and CLS, and function $\mathrm{H}(\cdot)$ is the entropy of the variables. $(\mathrm{H(CAT)H(CLS)})^{1/2}$ is used to normalize the mutual information to be in the range of [0, 1]

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \delta(y_i, \mathrm{map}(c_i))}{n} \tag{13}$$

where n denotes the size of the data, and y_i and c_i denote the true class label and the calculated cluster label, respectively, for the ith example. $\mathrm{map}(\cdot)$ is a permutation function that aligns the category label and the cluster label, using the Hungarian algorithm [62]. $\delta(y_i, \mathrm{map}(c_i))$ is an indicator function, which returns 1, if $y_i = \mathrm{map}(c_i)$, otherwise 0.

For the two measures, higher values represent a better performance. For more details about their definitions, the readers can refer to [63].

[6]http://www.ee.columbia.edu/ln/dvmm/CCV/.
[7]http://pascallin.ecs.soton.ac.uk/challenges/VOC/.
[8]http://www.public.asu.edu/jye02/Software/CCA/index.html.

TABLE II
CLUSTERING RESULTS IN TERMS OF NMI ON THE SIX DATABASES. BOTH THE MEAN VALUE AND THE STANDARD DEVIATION ARE REPORTED

NMI(%)	USPS	BBC	Cora	CCV	3Source	VOC
SingleV	59.12(1.95)	62.32(3.16)	18.20(1.00)	19.03(0.40)	53.38(2.12)	46.93(2.99)
CCA	75.54(3.07)	17.14(8.54)	1.30(0.46)	22.71(1.38)	60.39(6.94)	45.54(4.66)
PairwiseSC	71.16(1.45)	73.37(4.30)	27.94(2.00)	19.71(0.38)	62.25(2.76)	51.34(1.25)
CentroidSC	73.40(2.42)	73.39(3.78)	24.47(1.48)	22.09(0.62)	62.25(2.51)	53.08(0.99)
MultiCF	68.12(2.04)	76.47(1.72)	25.45(2.11)	23.24(0.38)	67.91(4.41)	58.43(3.57)
RMSC	71.81(1.16)	74.73(3.59)	14.55(1.37)	21.43(0.74)	62.02(2.28)	55.94(0.94)
PVC	65.38(1.81)	N/A	23.40(1.82)	14.30(0.61)	60.20(0.06)	67.20(1.24)
SingleTE	66.17(0.73)	63.82(0.11)	32.73(0.61)	19.54(0.16)	69.30(0.94)	62.86(1.00)
SpeMultiTE	77.80(0.50)	67.52(1.95)	25.39(0.34)	24.64(0.14)	70.65(0.98)	66.84(1.02)
MultiTE	**82.32(0.53)**	**81.58(0.60)**	**39.75(0.25)**	**25.24(0.26)**	**79.36(1.95)**	**67.85(0.44)**

TABLE III
CLUSTERING RESULTS IN TERMS OF ACC ON THE SIX DATABASES. BOTH THE MEAN VALUE AND THE STANDARD DEVIATION ARE REPORTED

ACC(%)	USPS	BBC	Cora	CCV	3Source	VOC
SingleV	61.81(4.81)	81.96(5.57)	32.75(1.47)	21.27(0.67)	52.93(3.59)	46.35(3.18)
CCA	74.53(4.96)	35.55(8.49)	26.95(1.58)	26.86(2.23)	62.37(6.76)	41.52(6.83)
PairwiseSC	75.85(5.89)	85.85(9.00)	44.57(3.57)	23.49(0.50)	58.37(3.28)	51.66(1.76)
CentroidSC	77.73(5.89)	87.29(7.99)	43.26(2.98)	25.39(0.93)	58.93(3.07)	56.43(2.03)
MultiCF	71.82(4.80)	89.29(1.14)	42.38(2.87)	26.00(0.36)	69.23(3.54)	55.22(4.83)
RMSC	78.77(4.96)	89.11(6.00)	33.74(2.41)	25.77(1.00)	58.17(3.45)	51.65(3.05)
PVC	67.64(4.70)	N/A	42.01(1.34)	18.29(0.72)	68.40(0.06)	65.72(3.14)
SingleTE	63.61(1.77)	84.70(0.05)	46.08(0.67)	21.19(0.31)	78.69(1.40)	56.41(2.51)
SpeMultiTE	73.08(1.42)	84.64(3.65)	45.82(1.02)	23.83(0.24)	80.16(1.46)	68.34(2.14)
MultiTE	**85.96(1.45)**	**93.74(0.99)**	**59.95(0.86)**	**26.88(0.61)**	**82.91(2.64)**	**69.97(1.03)**

D. Complete Multiview Clustering

The NMI and the accuracies of different clustering methods on the six data sets are shown in Tables II and III, respectively. Overall, it can be seen that our method outperforms all the algorithms with which it is compared. In particular, the improvements achieved by the proposed method MultiTE, in terms of ACC and NMI are, respectively, 15% and 12% on the Cora database, and 13% and 12% on the 3Source database.

PairwiseSC and CentroidSC methods obtain embeddings through the spectral analysis of the Laplacian matrices from multiview data, which are based on the similarities calculated between any two examples. On the other hand, our method obtains unified embedding through partial similarity triplets, constructed from the data, which are easier to obtain accurately and may even better reveal the true cluster structure than other methods. As RMSC achieves clustering in a similar manner with PairwiseSC and CentroidSC, the performance of MultiTE is better than that of RMSC.

As regard to PVC, it uses NMF to obtain unified low-dimensional embedding. We can explore and compare the rankings of data similarity between PVC and other methods, which may be more useful to discover the structure of group data. However, as PVC is based on NMF, it may limit its applications to multiview data with negative features.

MultiCF is a regressionlike clustering method, which may directly obtain the normalized cluster indicator matrix. However, it may be difficult to learn the cluster structure directly by projecting original feature spaces to the extent of indicating data labels because of semantic gaps. Thus, the performance of MultiCF is worse than that of our method. Besides, MultiCF needs to solve the projection matrices once per every column, and thus it consumes more time with increasing number of clusters.

As regard to SpeMultiTE, it learns low-dimensional embedding, adjusted to all views, which is unreasonable, because different views may produce contrasting triplets. Besides, as the views are complementary, each view may share only partial characteristics of the data, which is not a good way to learn view-specific embeddings of data. As MultiTE can eliminate the above-cited problems, its performance is considered better than that of SpeMultiTE. Furthermore, it is observed that SpeMultiTE does not outperform SingleTE on any data set, because using the same embedding for all views harms the data cluster structure, sometimes. The comparison with SpeMultiTE and SingleTE further validates the usefulness of considering multiview characteristics, i.e., consistency and complementarity.

E. Incomplete Multiview Clustering

Similar to [35], we consider two settings of incomplete multiview clustering. For a two-view data set, let c, m, and n be the numbers of examples appearing in the first and second

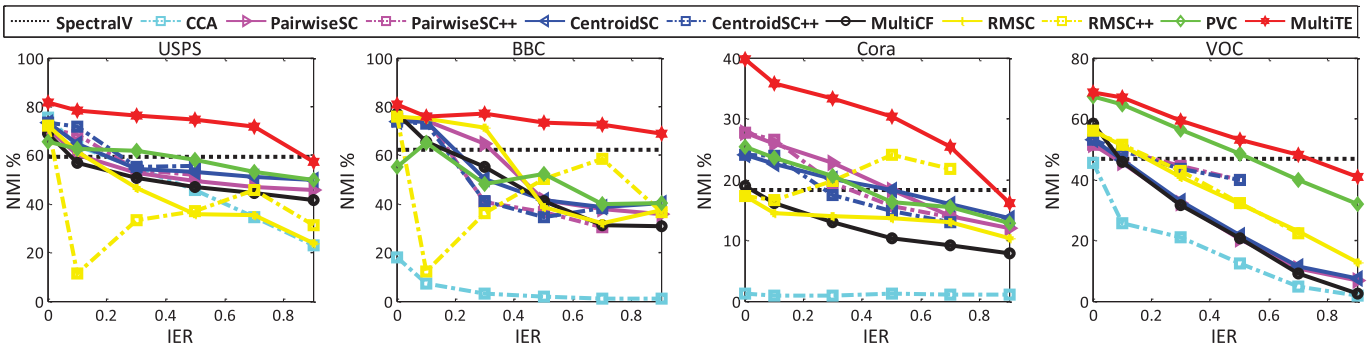


Fig. 2. NMI results on the four databases when both views suffer from the loss of examples.

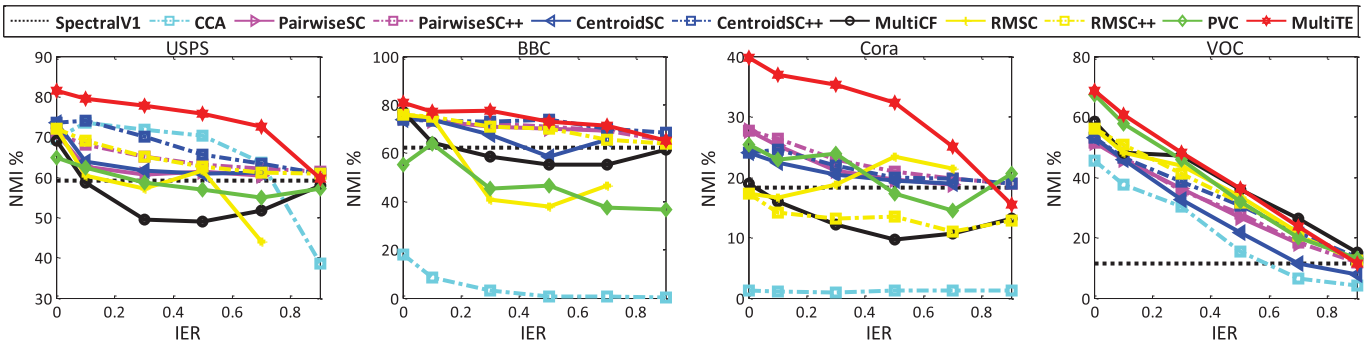


Fig. 3. NMI results on the four databases when the first view suffers from the loss of examples.

views, respectively. Then, the two settings are listed as follows.

1) *First Setting:* $m > 0$ and $n > 0$, which means that both views suffer from information loss.
2) *Second Setting:* Either $m = 0$ or $n = 0$, which means that at least one view is complete.

For the above-mentioned two settings, we randomly select 10%–90% of examples, with 20% as interval, to appear in only one view. This process is repeated 10 times and the average is reported. Furthermore, for the first setting, we even distribute the number of examples to appear in only one view, for simplicity.

Apart from the NMF-based incomplete multiview clustering method [35], there are a few algorithms that can preprocess incomplete views by filling missing information, as described in Section II. We compare our method with those algorithms as well. The method in [51] can deal with the situation wherein at least one view is complete; so, under the second setting, we can use [51] to preprocess kernel-based methods, i.e., PairwiseSC, CentroidSC, and RMSC. Furthermore, the method in [52] can deal with the scenario wherein no view is complete; so, under the first setting, we can use it to preprocess PairwiseSC, CentroidSC, and RMSC. In summary, the methods preprocessed in [51] or [52] are denoted as **PairwiseSC++**, **CentroidSC++**, and **RMSC++**, respectively.

1) Results Under the First Setting: Fig. 2 shows the NMI of all clustering methods on USPS, BBC, Cora, and VOC databases, under the first setting. Due to space limitation, we have to conduct our experiments on only four databases, and it is possible that similar results would be obtained with other data sets. An incomplete example ratio (IER) represents the percentage of examples appearing only in one view. Overall, it can be seen that our method outperforms all the methods with which it is compared, with different IERs.

For spectral-based methods, i.e., PairwiseSC, CentroidSC, and RMSC, we use the method proposed in [52] to fill their incomplete kernel matrices, which results in the methods of PairwiseSC++, CentroidSC++, and RMSC++. From the figure, it can be seen that the modified methods show a few or no improvements and, in some cases, the final clustering (especially for RMSC) may have been even harmed, because under incomplete views, the methods proposed in [43] may not promote kernel matrix completion, or the parameters selection will have to be done more carefully for the preprocessing method [52] and for approaching RMSC.

As regard to PVC, it learns unified low-dimensional embedding for incomplete multiview data, based on NMF, which is designed for incomplete multiview clustering without preprocessing. However, it cannot effectively explore the data relations, and this could be one reason why our method outperforms it.

In dealing with the methods not designed for incomplete multiview clustering, i.e., CCA, PairwiseSC, CentroidSC, MultiCF, and RMSC, we just use zeros to replace the missing features. This may be a little arbitrary, but there is, perhaps, no other method that can simultaneously fill missing information of visual features and the textual features. Besides, it is fair for comparison, as we do not preprocess the data at all.

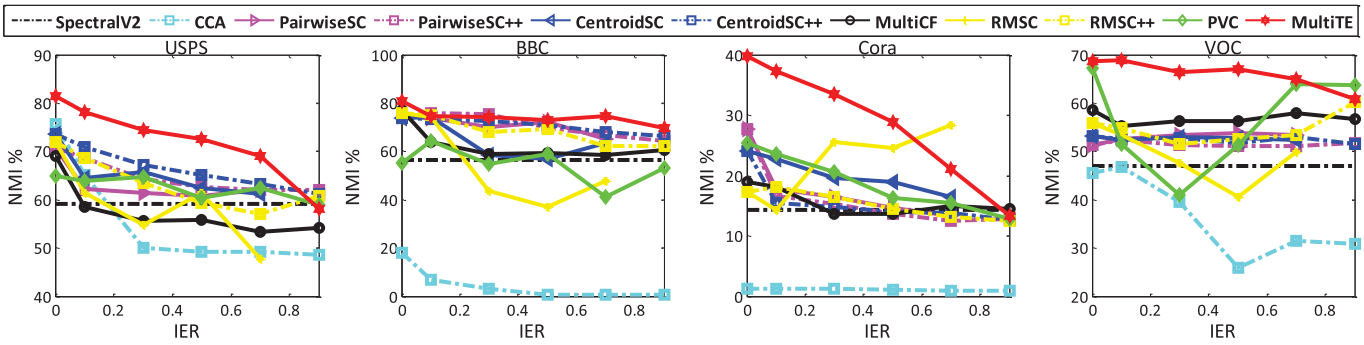


Fig. 4. NMI results on the four databases when the second view suffers from the loss of examples.

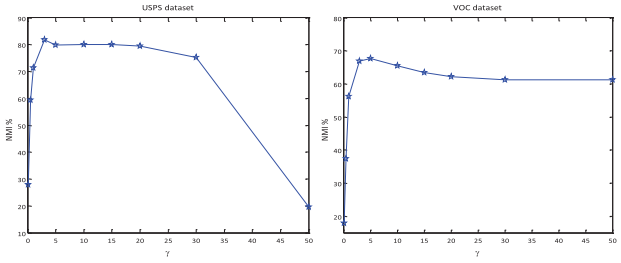


Fig. 5. NMI on the USPS and VOC data sets with the varying γ.

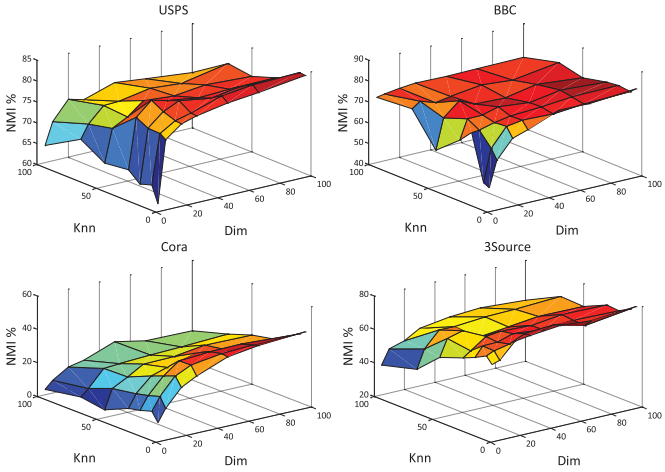


Fig. 6. NMI on the four databases with the varying Dim and K.

From the figure, it can be seen that, with increasing IER, our method degenerates less than the other methods. This implies that the proposed method can well deal with incomplete multiview clustering.

2) Results Under the Second Setting: The NMI of all the clustering methods on USPS, BBC, Cora, and VOC databases, under the second setting, with either the first view or the second view being incomplete, is shown in Figs. 3 and 4. IER is the same as in the first setting. Besides, the clustering performance of spectral clustering, under a complete view, is also reported. Overall, it can be seen that the proposed method outperforms the methods with which it is compared.

It should be noted that we use the method proposed in [51] to fill the kernel matrix of incomplete view, which results in the methods of PairwiseSC++, CentroidSC++, and RMSC++.

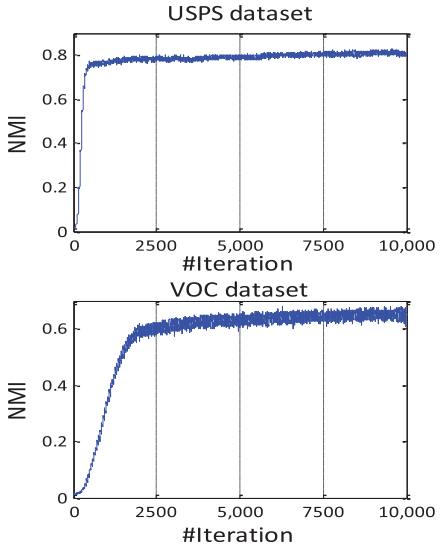


Fig. 7. NMI versus varying iterations.

From the figure, it can be seen that these modified methods obtain a better performance than the original ones, because using a complete view can better guide the learning of kernel matrices of incomplete views.

As regard to the performance of other methods, the results obtained are similar to those obtained in the first setting, excepting that all the methods achieve a better performance with the same IER. This may be because, under the second setting, we have a complete view, which may be more helpful than the scenario of incomplete views.

F. Parameter Study

In (6), parameter γ controls the margin size, which can be empirically set to be 5 in all the experiments. We find that this parameter is not so sensitive for the final results as shown in Fig. 5. Furthermore, there are two implicit parameters in the whole algorithm, i.e., the dimensionality of the unified embedding (denoted as Dim) and the size of K nearest neighbors (denoted as K). In this section, we test how the performance varies with changes in these two parameters. Due to space limitation, we have conducted our experiments on only four databases, and it is possible that similar results would be obtained with other data sets.

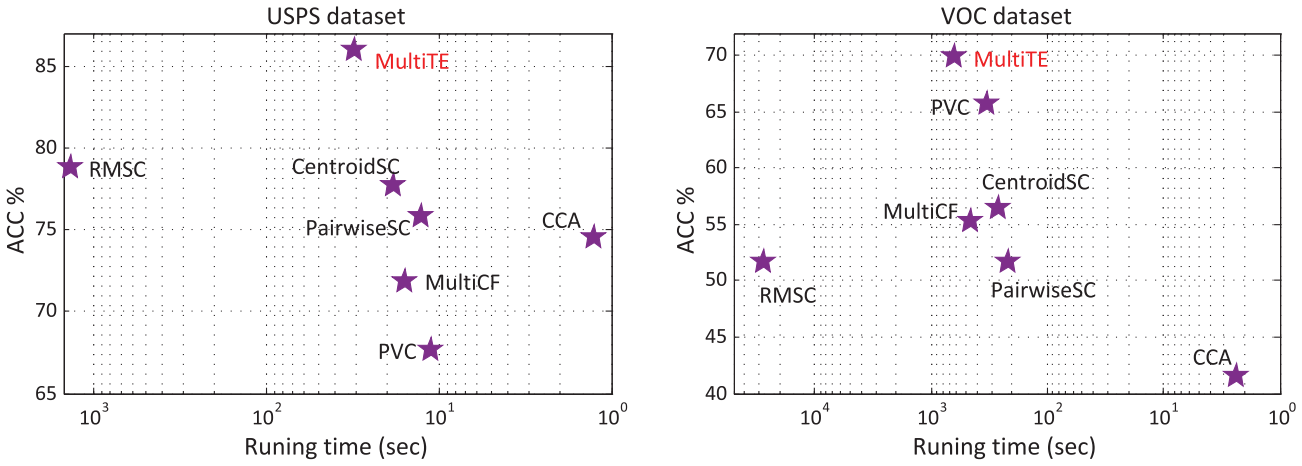


Fig. 8. Running time versus ACC for all the methods on the USPS and VOC data sets.

Fig. 6 shows that, when K is too large, the triplets may be wrongly constructed due to limitation of similarity metrics, but, when it is too small, the triplet sets are not big enough to reveal the data clustering structure. As regard to the dimensionality (Dim), when it is too small, unified embedding cannot embed enough information to reflect data, and when it is bigger than 40, the performance ceases to increase. Overall, we find that [10,20] and [30,100] are the optimal intervals for K and Dim, respectively.

G. Performance Versus Iteration

Fig. 7 illustrates the NMI curves obtained by varying iterations on USPS and VOC data sets. Due to space limitation, we have conducted our experiments on only two databases, and it is possible that similar results would be obtained with other data sets. In all the experiments, the size of a batch is selected to be 50. From the figure, it can be seen that not many iterations are required to obtain acceptable results, compared with the size of the triplet sets produced by multiview data, e.g., $4E + 7$ pairs on the USPS data set. This is reasonable, because many triplets on a view are repetitive. For example, sample A is more similar to B than C, and if the semantics between C and D are similar, the comparison of the triplet (A, B, and D) is not necessary. Furthermore, even though different views may produce contrasting triplets, they share numerous same triplets, because different views describe the same semantic content. Overall, the complexity is not a heavy burden for our algorithm.

H. Running Time

In this section, we would show the running time required for obtaining the embeddings by all the methods on USPS and VOC data sets, using the same machine (Intel CPU 3.1 GHz and 12-GB memory). The publicly available codes of all the methods, compared here, are written in MATLAB, excepting the main parts of PVC, which are written in C++. Our method is also written in MATLAB. As regard to MultiCF, we implement it using MATLAB, following the authors' suggestions. The experimental results are shown in Fig. 8.

From Fig. 8, we can see that our method achieves the best results; the time it takes for obtaining an embedding is the same as that taken by the mainstream methods. For PairwiseSC and CentroidSC methods, the kernel matrix of each view has to be calculated. More importantly, eigenvalue decomposition is performed in every iteration, when the embedding for each view is solved iteratively. For RMSC, kernel matrices will have to be precalculated before solving a problem, constrained by low rank and structure sparsity. Then, the augmented lagrangian multiplier scheme [64] is used for optimization, which brings in more auxiliary variables that render optimization more time-consuming because of the need for more iterations. For our method, as shown in Fig. 7, not many iterations are needed to achieve acceptable results by the stochastic method.

V. Conclusion and Future Work

In this paper, considering different views as different relations in a knowledge graph, we learn from the knowledge graph embedding and propose a novel complete and incomplete multiview clustering method. To model multiview characteristics, a novel measurement is developed to establish the relation between the learned unified and view-specific embeddings. Furthermore, our model can be extended to incomplete multiview clustering, which clusters data with incomplete feature sets. Extensive experiments have validated the effectiveness of the proposed method in relation to the state-of-the-art methods, for both complete and incomplete multiview clustering.

In our model, similarity triplets construction and latent embedding learning are two separate procedures, and hence, the similarity triplets used are constant in all the training procedures. Therefore, the potential possibility of interaction between similarity triplets construction and embedding learning is likely be ignored, but we may consider combining these two steps into a unified framework, in our future research.

References

[1] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, nos. 7–8, pp. 2031–2038, 2013.

[2] Y.-M. Xu, C.-D. Wang, and J.-H. Lai, "Weighted multi-view clustering with feature selection," *Pattern Recognit.*, vol. 53, pp. 25–35, May 2016.

[3] C.-D. Wang, J.-H. La, and P. S. Yu, "Multi-view clustering based on belief propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 1007–1021, Apr. 2016.

[4] R. Zall and M. R. Keyvanpour, "Semi-supervised multi-view ensemble learning based on extracting cross-view correlation," *Adv. Elect. Comput. Eng.*, vol. 16, no. 2, pp. 111–124, 2016.

[5] Z. Zhang, Z. Zhai, and L. Li, "Uniform projection for multi-view learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1675–1689, Aug. 2017.

[6] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 689–696.

[7] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2222–2230.

[8] C. Xu, D. Tao, and C. Xu, "Multi-view self-paced learning for clustering," in *Proc. Int. Conf. Artif. Intell.*, 2015, pp. 3974–3980.

[9] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[10] C. Xu, D. Tao, and C. Xu. (2013). "A survey on multi-view learning," [Online]. Available: https://arxiv.org/abs/1304.5634

[11] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 905–918, Feb. 2014.

[12] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, Nov. 2017.

[13] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 129–136.

[14] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.

[15] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 252–260.

[16] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 352–360.

[17] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 270–278.

[18] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2750–2756.

[19] J. Xu, J. Han, and F. Nie, "Discriminatively embedded K-means for multi-view clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5356–5364.

[20] L. Zong, X. Zhang, L. Zhao, H. Yu, and Q. Zho, "Multi-view clustering via multi-manifold regularized non-negative matrix factorization," *Neural Netw.*, vol. 88, pp. 74–89, Apr. 2017.

[21] J. Xu, J. Han, F. Nie, and X. Li, "Re-weighted discriminatively embedded k-means for multi-view clustering," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 3016–3027, Jun. 2017.

[22] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. Int. Conf. Data Mining*, 2004, pp. 19–26.

[23] A. Kumar and H. Daumé, III, "A co-training approach for multi-view spectral clustering," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 393–400.

[24] X. Zhao, N. Evans, and J.-L. Dugelay, "A subspace co-training framework for multi-view clustering," *Pattern Recognit. Lett.*, vol. 41, pp. 73–82, May 2014.

[25] E. Bruno and S. Marchand-Maillet, "Multiview clustering: A late fusion approach using latent models," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2009, pp. 736–737.

[26] D. Greene and P. Cunningham, "A matrix factorization approach for integrating multiple data views," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Bled, Slovenia, 2009, pp. 423–438.

[27] S. F. Hussain, M. Mushtaq, and Z. Halim, "Multi-view document clustering via ensemble method," *J. Intell. Inf. Syst.*, vol. 43, no. 1, pp. 81–99, 2014.

[28] Q. Yin, S. Wu, R. He, and L. Wang, "Multi-view clustering via pairwise sparse subspace representation," *Neurocomputing*, vol. 156, pp. 12–21, May 2015.

[29] P. Muthukrishnan, D. Radev, and Q. Mei, "Edge weight regularization over multiple graphs for similarity learning," in *Proc. Int. Conf. Data Mining*, 2010, pp. 374–383.

[30] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.

[31] O. Arandjelović, "Discriminative extended canonical correlation analysis for pattern set matching," *Mach. Learn.*, vol. 94, no. 3, pp. 353–370, Mar. 2014.

[32] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," Dept. Statist., Univ. California, Berkeley, CA, USA, Tech. Rep. 688, 2005.

[33] L. Shu and L. J. Latecki, "Integration of single-view graphs with diffusion of tensor product graphs for multi-view spectral clustering," in *Proc. Asian Conf. Mach. Learn.*, 2015, pp. 362–377.

[34] X. He, M.-Y. Kan, P. Xie, and X. Chen, "Comment-based multi-view clustering of Web 2.0 items," in *Proc. Int. Conf. World Wide Web*, 2014, pp. 771–782.

[35] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Partial multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1968–1974.

[36] M. Qian and C. Zhai, "Unsupervised feature selection for multi-view clustering on text-image Web news data," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 1963–1966.

[37] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[38] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[39] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[40] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. J. Kriegman, and S. Belongie, "Generalized non-metric multidimensional scaling," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2007, pp. 11–18.

[41] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai. (2011). "Adaptively learning the crowd kernel." [Online]. Available: https://arxiv.org/abs/1105.1033

[42] L. van der Maaten and K. Weinberger, "Stochastic triplet embedding," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Sep. 2012, pp. 1–6.

[43] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[44] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.

[45] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 926–934.

[46] A. Klami and S. Kaski, "Probabilistic approach to detecting dependencies between data sets," *Neurocomputing*, vol. 72, nos. 1–3, pp. 39–46, 2008.

[47] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 586–594.

[48] 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. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.

[49] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1302–1308.

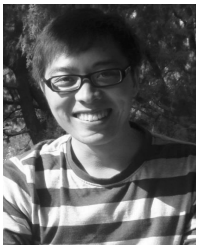
[50] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2833–2843, Jun. 2016.

[51] P. Rai, A. Trivedi, H. Daumé, III, and S. L. DuVall, "Multiview clustering with incomplete views," in *Proc. NIPS Workshop Mach. Learn. Social Comput.*, 2010, pp. 1–7.

[52] W. Shao, X. Shi, and P. S. Yu, "Clustering on multiple incomplete datasets via collective kernel learning," in *Proc. Int. Conf. Data Mining*, 2013, pp. 1181–1186.

[53] W. Shao, L. He, and P. S. Yu, "Multiple incomplete views clustering via weighted nonnegative matrix factorization with $L_{2,1}$ regularization," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 318–334.

[54] 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.
[55] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proc. AAAI Conf. Artif. Intell.*, 2011, pp. 301–306.
[56] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1112–1119.
[57] B. McFee and G. Lanckriet, "Learning multi-modal similarity," *J. Mach. Learn. Res.*, vol. 12, pp. 491–523, Feb. 2011.
[58] L. Zhang, S. Maji, and R. Tomioka. (2015). "Jointly learning multiple measures of similarities from triplet comparisons." [Online]. Available: https://arxiv.org/abs/1503.01521
[59] J. D. M. Rennie and N. Srebro, "Loss functions for preference levels: Regression with discrete ordered labels," in *Proc. IJCAI Multidiscipl. Workshop Adv. Preference Handling*, 2005, pp. 180–186.
[60] Y. Huang, W. Wang, and L. Wang, "Instance-aware image and sentence matching with selective multimodal lstm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2310–2318.
[61] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[62] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1982.
[63] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 568–586, Mar. 2011.
[64] Z. Lin, M. Chen, and Y. Ma. (2010). "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices." [Online]. Available: https://arxiv.org/abs/1009.5055



Qiyue Yin received the B.S. degree in automation control from Harbin Engineering University, Harbin, China, in 2012. He is currently pursuing the Ph.D. degree with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His current research interests include clustering, recommender system, and computer vision.

Shu Wu (M'13) received the B.S. degree from Hunan University, Changsha, China, in 2004, the M.S. degree from Xiamen University, Xiamen, China, in 2007, and the Ph.D. degree from the University of Sherbrooke, Sherbrooke, QC, Canada, in 2012, all in computer science.

He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include data mining and recommendation systems.

Liang Wang (SM'09) received the B.S. and M.S. degrees from Anhui University, Hefei, China, in 1997 and 2000, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2004.

From 2004 to 2010, he was a Research Assistant with Imperial College London, London, U.K., and Monash University, Melbourne, VIC, Australia; a Research Fellow with the University of Melbourne, Melbourne; and a Lecturer with the University of Bath, Bath, U.K. He is currently a Full Professor of the Hundred Talents Program, National Laboratory of Pattern Recognition, CASIA. He has widely published at highly ranked international journals, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and IEEE TRANSACTIONS ON IMAGE PROCESSING, and leading international conferences, such as the Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, and the IEEE International Conference on Data Mining. His current research interests include machine learning, pattern recognition, and computer vision.

Dr. Wang is currently an International Association for Pattern Recognition Fellow. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-Part B.