Pattern Recognition 97 (2020) 107015

Contents lists available at ScienceDirect

Pattern Recognition

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

Auto-weighted multi-view clustering via deep matrix decomposition

Shudong Huang, Zhao Kang, Zenglin Xu*

SMILE Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

ARTICLE INFO

Article history:
Received 16 February 2019
Revised 8 August 2019
Accepted 17 August 2019
Available online 28 August 2019

Keywords:
Multi-view learning
Deep matrix decomposition
Clustering
Optimization algorithm

ABSTRACT

Real data are often collected from multiple channels or comprised of different representations (i.e., views). Multi-view learning provides an elegant way to analyze the multi-view data for low-dimensional representation. In recent years, several multi-view learning methods have been designed and successfully applied in various tasks. However, existing multi-view learning methods usually work in a single layer formulation. Since the mapping between the obtained representation and the original data contains rather complex hierarchical information with implicit lower-level hidden attributes, it is desirable to fully explore the hidden structures hierarchically. In this paper, a novel deep multi-view clustering model is proposed by uncovering the hierarchical semantics of the input data in a layer-wise way. By utilizing a novel collaborative deep matrix decomposition framework, the hidden representations are learned with respect to different attributes. The proposed model is able to collaboratively learn the hierarchical semantics obtained by each layer. The instances from the same class are forced to be closer layer by layer in the low-dimensional space, which is beneficial for the subsequent clustering task. Furthermore, an idea weight is automatically assigned to each view without introducing extra hyperparameter as previous methods do. To solve the optimization problem of our model, an efficient iterative updating algorithm is proposed and its convergence is also guaranteed theoretically. Our empirical study on multi-view clustering task shows encouraging results of our model in comparison to the state-of-the-art algorithms.

1. Introduction

In many real-world applications, datasets have heterogeneous features which are collected from multiple channels or with multiple modalities. For instance, images can be introduced by different visual descriptors like LBP [1], HOG [2] and SIFT [3]; a video can be represented by different types (views) of features such as images and sounds [4]. Different views usually capture different aspects of information, any of which suffices for mining knowledge. Furthermore, the encoded information of different views is consistent and complementary to each other, which is instrumental to produce better performance. Thus it is expected to exploit multiple views in depth to generate more promising results rather than relying on a single view. Therefore, it is critical to develop a new learning paradigm, i.e., multi-view learning, to efficiently analyze these heterogeneous features such that the accuracy and robustness of learning algorithms can be improved. Previous works have also shown that the hidden common structure share by different views can be fully extracted [5,6].

In recent years, a number of multi-view learning models have been proposed under the framework of different theories and methodologies [7]. Moreover, these methods have been widely applied in supervised and unsupervised framework (e.g., multi-view clustering) [8,9]. The key point of utilizing the multiple views to deal with the multi-view learning task is to reasonably fuses these views to produce a more accurate and robust result [10]. Existing multi-view learning methods can be roughly categorized into graph-based approaches and subspace approaches. The former considers the multi-view learning problem as a graph partition problem by transforming the multi-view learning process into a multiple graph fusion situation [10,11]. While the latter tries to uncover the common latent subspace shared by multiple views [12,13]. It is worth noting that graph-based multi-view methods are generally studied by extending the traditional spectral methods, while subspace multi-view methods are designed by utilizing the matrix decomposition (also called matrix factorization (MF)). Since graph-based methods suffer from the problem of time-consuming due to the graph construction as well as the eigen decomposition, more and more researchers focus on applying the MF strategy to solve the multi-view learning problems [14]. There are several MF-based multi-view works based on information theoretic [15], which essentially can be treated as a special kind of subspace methods. Some multi-view learning methods are also in-

* Corresponding author.
E-mail address: zlxu@uestc.edu.cn (Z. Xu).

https://doi.org/10.1016/j.patcog.2019.107015
0031-3203/© 2019 Elsevier Ltd. All rights reserved.

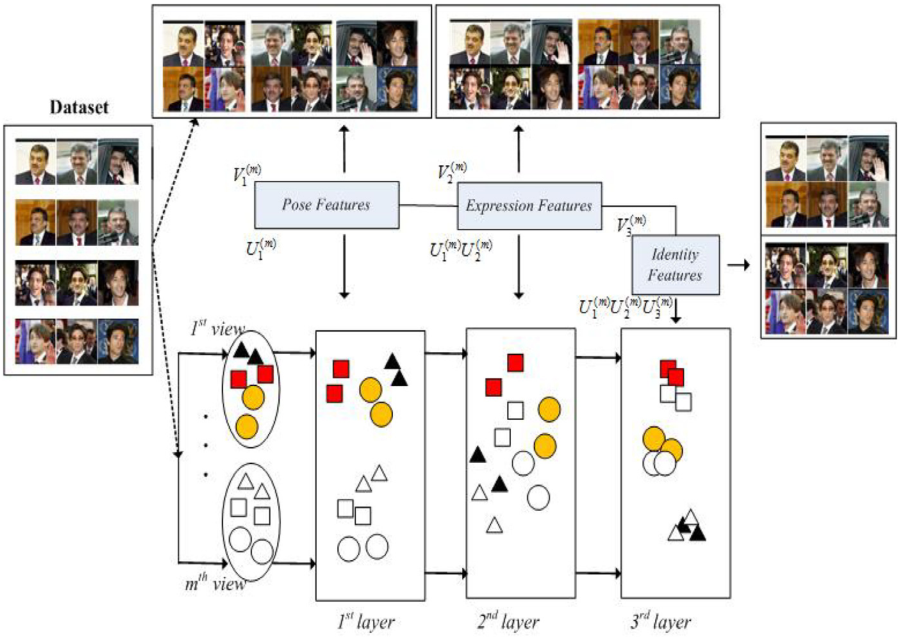


Fig. 1. There are three deep MF structures proposed to demonstrate the framework of our model. Same shape represents the instances belong to the same class. We can see the variability in face data can stem from the attributes such as the pose (eyes left, right or front) or the facial expression (with or without a smile) of the subject. By utilizing the proposed deep structure, the hierarchical information share by the views can be fully exploited layer by layer. And finally, a more discriminative representation is generated.

vestigated based on low-rank learning [14], canonical correlation analysis [16], latent subspace learning [17], and so on. Multi-view learning has been successfully applied in face recognition [18], image classification [19], text mining [20], *etc*.

As mentioned before, MF-based multi-view learning models have been widely studied in recent years and were reported with good performance. However, these models usually work in a single layer formulation, in which the mapping between the obtained representation and the original data contains rather complex hierarchical information with implicit lower-level hidden attributes. In addition, determining on the weights of different views of data is a crucial challenge to a number of multi-view clustering algorithms, since different views can play different roles to clustering. In this paper, a novel deep multi-view learning model is proposed by uncovering the hierarchical semantics of the input data in a layer-wise way, as shown in Fig. 1. As illustrated, by utilizing a novel deep matrix factorization framework, the hidden representations are learned with respect to different attributes. Taking the human face dataset as an example, the variability of data not only stems from the difference in the appearance of the subjects, but also the other attributes such as the facial expression or the pose of the subject. Since such attributes can help identify the faces, the face recognition problem can be better solved using our deep MF framework, as each subsequent layers can capture the corresponding hierarchical structures. As a result, the instances from the same class but from different views are forced to be closer layer by layer in the low-dimensional space, which is beneficial for the subsequent learning task. In addition, in order to automatically determine the weights of different views, we introduce the auto-weighting scheme into the deep multi-view clustering algorithm. Furthermore, to solve the optimization problem of our model, an efficient iterative updating algorithm is proposed with a theoretical guarantee of its convergence.

The rest of the paper is organized as follows. We present a brief review of previous related research in Section 2. We give a detailed description of our model in Section 3. Experimental results are presented in Section 4. The paper ends with a conclusion in Section 5.

2. Preliminaries

Before presenting the details of our method, we first briefly review previous research closely related to this paper.

Matrix decomposition techniques have been widely applied in data mining and pattern recognition. Among them, Nonnegative Matrix Factorization (NMF) has received much attention due to its physiological and psychological interpretation. In this section, we will briefly review NMF [21,22]. Given a nonnegative data matrix $\mathbf{X} = [\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_n] \in \mathbb{R}^{f\times n}$, where n is the number of data points and f is the data dimension. NMF aims to find two nonnegative matrices $\mathbf{U} \in \mathbb{R}^{f\times C}$ and $\mathbf{V} \in \mathbb{R}^{n\times C}$ which minimize the objective function as follows

$$J_{NMF} = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\mathbf{X}_{ij} - (\mathbf{U}\mathbf{V}^T)_{ij}\right)^2 = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \ \text{s.t.} \mathbf{U} \geq 0, \mathbf{V} \geq 0, \quad (1)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\mathbf{X}_{ij}$ denotes the (i, j)-th element of $\mathbf{X}$. It has been proved that the objective function J_{NMF} in Eq. (1) is convex in $\mathbf{U}$ only or $\mathbf{V}$ only. To optimize the objective function, Lee et al. [21] proposed the following iterative updating algorithm

$$\begin{aligned}\mathbf{U}_{ij} &\leftarrow \mathbf{U}_{ij}\frac{(\mathbf{X}\mathbf{V})_{ij}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ij}},\\ \mathbf{V}_{ij} &\leftarrow \mathbf{V}_{ij}\frac{(\mathbf{X}^T\mathbf{U})_{ij}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ij}}.\end{aligned}$$

In the clustering setting of NMF [23,24], $\mathbf{U} \in \mathbb{R}^{f\times C}$ represents the basis matrix, $\mathbf{V} \in \mathbb{R}^{n\times C}$ denotes the representation matrix and C is the number of hidden patterns. Since $C \ll n$ and $C \ll f$, NMF would find a low-dimensional representation $\mathbf{V}$ of $\mathbf{X}$. It is worth noting that previous studies have shown that NMF is equivalent to relaxed k-means clustering, which is a centroid based method and only works for single view data clustering [25].

Real-world data set usually contains multiple modalities (i.e., factors). Take the face data set as an example, it contains modalities including pose, expression, etc. Traditional NMF cannot fully uncovers the hidden structure of those factors. The multi-layer decomposition process of a data matrix can be formulated as

$$\begin{aligned}
&\mathbf{X} \approx \mathbf{U}_1\mathbf{V}_1^T,\\
&\mathbf{X} \approx \mathbf{U}_1\mathbf{U}_2\mathbf{V}_2^T,\\
&\quad\vdots\\
&\mathbf{X} \approx \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T,
\end{aligned} \tag{2}$$

where $\mathbf{U}_i$ is the i-th layer basis matrix and $\mathbf{V}_i$ denotes i-th layer representation matrix. It is clear that the deep NMF model also aims to find a low-dimensional representation of the original data matrix that has a similar interpretation at the top layer, i.e., the last factor $\mathbf{V}_r$ (r is the number of layers).

By further factorizing $\mathbf{V}_i$, the deep model can automatically learn what this latent hierarchy of attributes is. In other words, by introducing an extra layer of abstraction minimizing the dimensionality of obtained representation, we can automatically exploit the corresponding latent attributes as well as the intermediate hidden representations that are implied, allowing for a better higher-level representation $\mathbf{V}_r$. Finally, a better high-level, final-layer representation for the for subsequent learning task according to the attribute with the lowest variability. Compared with the single layer NMF model as described in Eq. (1), deep NMF model has a better ability to uncover the hidden structure since the representations of each layer can identify the different attributes [26].

3. Problem formulation

The MF models mentioned above only work for single-view data clustering. In this paper, we focus on extending the traditional MF model to a novel deep matrix factorization framework for learning multi-view clustering. Generally, a multi-view clustering system consists of four components: feature extraction, algorithm design, view fusion and data clustering. The first component is not our main concern as we compare with other methods on benchmark multi-view datasets. This section introduces the last three components in detail. It is noteworthy that the last two components, i.e., view fusion and data clustering, can be completed simultaneously in our model.

3.1. The objective function

We first introduce the following notations. Denote the multi-view data with M views as $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(M)}\}$. $\{\mathbf{X}^{(m)}\}|_{m=1}^{M} = [\mathrm{x}_1^{(m)}, \mathrm{x}_2^{(m)}, \cdots, \mathrm{x}_n^{(m)}] \in \mathbb{R}^{f^{(m)}\times n}$, where n is the number of data points ($\mathrm{x}_j^{(m)}$ denotes the j-th data point of $\mathbf{X}^{(m)}$), $f^{(m)}$ is the feature dimension of the m-th view. Denote $\mathbf{U}_i^{(m)}$ and $\mathbf{V}_i^{(m)}$ as the basis matrix and the obtained representation matrix of the i-th layer mapping for the m-th view, respectively.

The key point of utilizing the multiple views to deal with the multi-view learning task is to reasonably fuses these views to produce a more accurate and robust result. It is obviously unwise to analyse the multiple views based on the concatenated features of different views with equal weight, as the good views and weak views are treated equally in this way. In order to exploit the complementary aspects of information shared by different views, a reasonable strategy is to combine the views with suitable weights $\alpha^{(m)} (m = 1, \ldots, M)$, and keep the smooth of the weights distribution by introducing an extra parameter λ. Moreover, multi-view learning methods seek to find the solution which uncovers the common latent structure shared by multiple views, thus the representation matrix obtained in the final layer of different views should be unique, i.e., $\mathbf{V}_r^{(1)} = \mathbf{V}_r^{(2)} = \cdots = \mathbf{V}_r^{(m)} = \mathbf{V}_r$. Unlike Eq. (1), we remove the non-negative constraint on $\mathbf{U}_i^{(m)}$, thus the input data matrix $\mathbf{X}$ is allowed to have mixed signs, which extends the applicable range of our model. Meanwhile, to obtain more stable data representation performance with respect to a fixed initialization, the robust multi-view clustering method is desired. In this paper, we further utilize a sparsity-inducing norm to reduce the effects of the data noise, inspired by recent developments in $l_{2,1}$-norm [27]. We propose a novel deep matrix decomposition framework for learning multi-view clustering by solving:

$$\begin{aligned}
&\min_{\mathbf{U}_i^{(m)},\mathbf{V}_i^{(m)},\alpha^{(m)}} \sum_{m=1}^{M} \left(\alpha^{(m)}\right)^{\lambda} \|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\cdots\mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1},\\
&\text{s.t.} \mathbf{V}_i^{(m)} \geq 0, i \in [1, \ldots, r-1], (\mathbf{V}_r)_{.c} = \{0, 1\},\\
&\sum_{c=1}^{C} (\mathbf{V}_r)_{.c} = 1, \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0,
\end{aligned} \tag{3}$$

where $(\mathbf{V}_r)_{.c}$ denotes a row element of $\mathbf{V}_r$, $\alpha^{(m)}$ is the weight for the m-th view, λ is a positive scalar and it could be regularization parameter in another form:

$$\begin{aligned}
&\min_{\mathbf{U}_i^{(m)},\mathbf{V}_i^{(m)},\alpha^{(m)}} \sum_{m=1}^{M} \left(\alpha^{(m)} \sum_{i=1}^{n} \|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\cdots\mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1} + \lambda\|\alpha^{(m)}\|_2^2\right)\\
&\text{s.t.} \mathbf{V}_i^{(m)} \geq 0, i \in [1, \ldots, r-1], (\mathbf{V}_r)_{.c} = \{0, 1\},\\
&\sum_{c=1}^{C} (\mathbf{V}_r)_{.c} = 1, \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0.
\end{aligned} \tag{4}$$

As we can see, each row of $\mathbf{V}_r$ in our model is coded as 1-of-C scheme. The main purpose is to guarantee the uniqueness of our solution. Furthermore, the partition result can be directly obtained without any post process.

Since λ can be searched in a large range and the choice of its value is crucial to final multi-view clustering performance. Thus we expect to remove such a parameter while pursuing good performance. In this paper, we propose an auto-weighted strategy to address such challenging problem.

3.2. Auto-weighted deep MF multi-view clustering

Motivated by recently proposed *Iteratively Re-weighted* technique [28,29], we propose a novel model for multi-view clustering as

$$\begin{aligned}
&\min_{\mathbf{U}_i^{(m)},\mathbf{V}_i^{(m)}} \sum_{m=1}^{M} \sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)}\cdots\mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}\\
&\text{s.t.} \mathbf{V}_i^{(m)} \geq 0, i \in [1, \ldots, r-1], (\mathbf{V}_r)_{.c} = \{0, 1\},\\
&\sum_{c=1}^{C} (\mathbf{V}_r)_{.c} = 1.
\end{aligned} \tag{5}$$

We can see there is no weight hyperparameter explicitly defined therein. The Lagrange function of Eq. (5) can be defined as

$$\min_{\mathbf{U}_i^{(m)},\mathbf{V}_i^{(m)}} \sum_{m=1}^{M} \sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)}\cdots\mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}} + \Gamma(\Psi, \mathbf{V}_i^{(m)}), \tag{6}$$

where Ψ denotes the Lagrange multiplier for constraints on $\mathbf{V}_i^{(m)}$, and $\Gamma(\Psi, \mathbf{V}_i^{(m)})$ is the formalized term derived from constraints. Taking the derivative of Eq. (6) w.r.t $\mathbf{V}_i^{(m)}$ and setting the derivative to zero, we have

$$\sum_{m=1}^{M} \alpha^{(m)} \frac{\partial \|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)}\cdots\mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}{\partial \mathbf{V}_i^{(m)}} + \frac{\partial \Gamma(\Psi, \mathbf{V}_i^{(m)})}{\partial \mathbf{V}_i^{(m)}} = 0, \tag{7}$$

where

$$\alpha^{(m)} = 1 \Big/ \left(2\sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}} \right). \tag{8}$$

Since the weights $\alpha^{(m)}$ are dependent on target variables $\mathbf{V}_i^{(m)}$ and $\mathbf{U}_i^{(m)}$, we cannot solve Eq. (7) directly. But if $\alpha^{(m)}$ are set to be stationary, Eq. (7) can be denoted as the solution to the following problem

$$\begin{aligned} &\min_{\mathbf{U}_i^{(m)},\mathbf{V}_i^{(m)}} \sum_{m=1}^{M} \alpha^{(m)} \|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1} \\ &\text{s.t.} \mathbf{V}_i^{(m)} \geq 0, i \in [1, \ldots, r-1], (\mathbf{V}_r)_{.c} = \{0, 1\}, \\ &\sum_{c=1}^{C} (\mathbf{V}_r)_{.c} = 1. \end{aligned} \tag{9}$$

Based on the assumption that the weights $\alpha^{(m)}$ are stationary, the Lagrange function of Eq. (5) can also be applied to Eq. (9). If we update $\mathbf{U}_i^{(m)}$ and $\mathbf{V}_i^{(m)}$ by solving Eq. (9), the values of $\alpha^{(m)}$ can be further updated correspondingly. This inspires us to optimize Eq. (5) using an iterative way. When the proposed iterative algorithm converges, the converged values of $\mathbf{U}_i^{(m)}$ and $\mathbf{V}_i^{(m)}$ are the local optimal solution to Eq. (5) according to Eqs. (7) and (8). Similarly, $\alpha^{(m)}$ are tuned to the optimal values correspondingly, and they are exactly the learned weights for all views.

It is worth mentioning that our model is different from previous deep structure based multi-view learning methods [30,31]. The major difference is that [30,31] are based on Canonical Correlation Analysis. And these methods can only handle the 2-view case, while our model has no such limitation. Another related work is proposed in [32], which is designed for deep multi-view clustering (DMVC). However, the residual calculation of DMVC is based on traditional Frobenius norm, which is sensitive to data noise in practical problems. In addition, the obtained data representation in DMVC cannot directly assign the partition result, therefore needs a post-processing (e.g., spectral clustering algorithm used in [32]) to assign the class label to each data point. Furthermore, the weight for each view in DMVC is computed by introducing an hyperparameter, thus the final performance is highly dependent on parameter searching.

In the following, we propose an efficient iterative updating algorithm to solve the optimization problem of Eq. (9). We will optimize the objective with respect to one variable while fixing the other variables. This procedure repeats until convergence.

3.3. Optimization

Before the optimization, we conduct the pre-training by decomposing each view $\mathbf{X}^{(m)} \approx \mathbf{U}_1^{(m)}\mathbf{V}_1^{(m)^T}$, where $\mathbf{V}_1^{(m)} \in \mathbb{R}^{n\times k_1}$ and $\mathbf{U}_1^{(m)} \in \mathbb{R}^{f^{(m)}\times k_1}$. Then the representation matrix is further decomposed as $\mathbf{V}_1^{(m)} \approx \mathbf{U}_2^{(m)}\mathbf{V}_2^{(m)^T}$, where $\mathbf{V}_2^{(m)} \in \mathbb{R}^{n\times k_2}$ and $\mathbf{U}_2^{(m)} \in \mathbb{R}^{k_1\times k_2}$. Here we define k_1 and k_2 as the dimensionalities of layer 1 and layer 2, respectively.[1] Note that the pre-training process can be accomplished by simply using k-means, as the relaxed k-means is equivalent to the matrix factorization method [33]. The process will be repeated till all layers are pre-trained.

First we rewrite Eq. (9) as follows

$$\begin{aligned} &\min_{\mathbf{U}_i^{(m)},\mathbf{V}_i^{(m)}} \sum_{m=1}^{M} \alpha^{(m)} \mathrm{Tr}\left(\mathbf{Y}^{(m)}\mathbf{D}^{(m)}\mathbf{Y}^{(m)^T}\right) \\ &\text{s.t.} \mathbf{V}_i^{(m)} \geq 0, i \in [1, \ldots, r-1], (\mathbf{V}_r)_{.c} = \{0, 1\}, \\ &\sum_{c=1}^{C} (\mathbf{V}_r)_{.c} = 1, \end{aligned} \tag{10}$$

where

$$\mathbf{Y}^{(m)} = \left(\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\right), \tag{11}$$

and $\mathbf{D}^{(m)}$ denotes a diagonal matrix:

$$\mathbf{D}^{(m)} = \frac{1}{2\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_2}. \tag{12}$$

Computation of $\mathbf{U}_i^{(m)}$.

Optimizing Eq. (10) w.r.t. $\mathbf{U}_i^{(m)}$ is equivalent to minimizing the problem as follows

$$J_U = \min_{\mathbf{U}_i^{(m)}} \sum_{m=1}^{M} \alpha^{(m)} \mathrm{Tr}\left(\mathbf{Y}^{(m)}\mathbf{D}^{(m)}\mathbf{Y}^{(m)^T}\right). \tag{13}$$

Taking the partial derivation of J_U w.r.t. $\mathbf{U}_i^{(m)}$, we have

$$\frac{\partial J_U}{\partial \mathbf{U}_i^{(m)}} = -2\mathbf{\Phi}^T\mathbf{X}^{(m)}\widetilde{\mathbf{D}}^{(m)}\widetilde{\mathbf{V}}_i^{(m)} + 2\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{U}_i^{(m)}\widetilde{\mathbf{V}}_i^{(m)^T}\widetilde{\mathbf{D}}^{(m)}\widetilde{\mathbf{V}}_i^{(m)}, \tag{14}$$

where

$$\widetilde{\mathbf{D}}^{(m)} = \alpha^{(m)}\mathbf{D}^{(m)}, \tag{15}$$

and $\mathbf{\Phi}^{(m)} = \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_{i-1}^{(m)}$ and $\widetilde{\mathbf{V}}_i^{(m)}$ is the reconstruction of the i-th layer's feature matrix.

Setting Eq. (14) $= 0$ leads to the following update rule

$$\mathbf{U}_i^{(m)} = \left(\mathbf{\Phi}^{(m)^T}\mathbf{\Phi}^{(m)}\right)^{-1}\mathbf{\Phi}^{(m)^T}\mathbf{X}^{(m)}\widetilde{\mathbf{D}}^{(m)}\widetilde{\mathbf{V}}_i^{(m)}\left(\widetilde{\mathbf{V}}_i^{(m)^T}\widetilde{\mathbf{D}}^{(m)}\widetilde{\mathbf{V}}_i^{(m)}\right)^{-1} \tag{16}$$

Computation of $\mathbf{V}_i^{(m)}$ $(i < r)$.

Optimizing Eq. (10) w.r.t. $\mathbf{V}_i^{(m)}$ is equivalent to minimizing

$$\begin{aligned} J_V = &\min_{\mathbf{V}_i^{(m)}} \sum_{m=1}^{M} \alpha^{(m)} \mathrm{Tr}\left(\mathbf{Y}^{(m)}\mathbf{D}^{(m)}\mathbf{Y}^{(m)^T}\right) \\ &\text{s.t.} \mathbf{V}_i^{(m)} \geq 0, i \in [1, \ldots, r-1]. \end{aligned} \tag{17}$$

Taking the partial derivation of J_V w.r.t. $\mathbf{V}_i^{(m)}$:

$$\frac{J_V}{\partial \mathbf{V}_i^{(m)}} = 2\widetilde{\mathbf{D}}^{(m)}\mathbf{V}_i^{(m)}\mathbf{U}_i^{(m)^T}\mathbf{\Phi}^{(m)^T}\mathbf{\Phi}^{(m)}\mathbf{U}_i^{(m)} - 2\widetilde{\mathbf{D}}^{(m)}\mathbf{X}^{(m)^T}\mathbf{\Phi}^{(m)}\mathbf{U}_i^{(m)} \tag{18}$$

Setting $\frac{\mathcal{L}}{\partial \mathbf{V}_i^{(m)}} = 0$, we have

$$\mathbf{V}_i^{(m)} = \left(\mathbf{X}^{(m)^T}\mathbf{\Phi}^{(m)}\mathbf{U}_i^{(m)}\right)\left(\mathbf{U}_i^{(m)^T}\mathbf{\Phi}^{(m)^T}\mathbf{\Phi}^{(m)}\mathbf{U}_i^{(m)}\right)^{-1}. \tag{19}$$

Computation of $\mathbf{V}_r$ (i.e., $\mathbf{V}_i^{(m)}$, $(i = r)$).

Optimizing Eq. (10) w.r.t. $\mathbf{V}_r$ is equivalent to solving

$$\begin{aligned} &\min_{\mathbf{V}_r} \sum_{m=1}^{M} \alpha^{(m)} \mathrm{Tr}\left(\mathbf{Y}^{(m)}\mathbf{D}^{(m)}\mathbf{Y}^{(m)^T}\right) \\ = &\min_{\mathrm{v}} \sum_{m=1}^{M}\sum_{i=1}^{n} \widetilde{d}_i^{(m)} \|\mathrm{x}_i^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathrm{v}_i\|_2^2 \end{aligned}$$

[1] For convenience of discussion, we simply denote the dimensionalities (layer size) from layer 1 to layer r as $p = [k_1, k_2, \cdots, k_r]$ in the experiments. E.g., if we define $p = [100\ 50\ C]$, it means there are three layers in the deep framework, and $\mathbf{X}^{(m)}$ is decomposed as $\mathbf{X}^{(m)} \approx \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)}\mathbf{U}_3^{(m)}\mathbf{V}_3^{(m)^T}$, where $\mathbf{U}_1^{(m)} \in \mathbb{R}^{f^{(m)}\times 100}$, $\mathbf{U}_2^{(m)} \in \mathbb{R}^{100\times 50}$, $\mathbf{U}_3^{(m)} \in \mathbb{R}^{50\times C}$ and $\mathbf{V}_3^{(m)} \in \mathbb{R}^{n\times C}$.

$$= \min_{\mathrm{v}} \sum_{i=1}^{n} \left(\sum_{m=1}^{M} \widetilde{d}_i^{(m)} \|\mathrm{x}_i^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)} \mathrm{v}_i\|_2^2 \right)$$

$$\text{s.t.} (\mathbf{V}_r)_{.c} = \{0, 1\}, \sum_{c=1}^{C} (\mathbf{V}_r)_{.c} = 1, \tag{20}$$

where $\mathrm{x}_i^{(m)}$ is the i-th data point of $\mathbf{X}^{(m)}$, v_i is the i-th column of $\mathbf{V}_r^T$ and $\widetilde{d}_i^{(m)}$ is the i-th diagonal element of $\widetilde{D}^{(m)}$. The optimization problem in Eq. (20) can be solved by decoupling the data and assigning the cluster indicator for them one by one independently. In other words, for a fixed specific j, we solve the following problem

$$\min_{\mathrm{v}} \sum_{m=1}^{M} \widetilde{d}^{(m)} \|\mathrm{x}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)} \mathrm{v}\|_2^2$$

$$\text{s.t.} \nu_c = \{0, 1\}, \sum_{c=1}^{C} \nu_c = 1, \tag{21}$$

here we define $\mathrm{v} = [\nu_1, \nu_2, \cdots, \nu_C]^T \in \mathbb{R}^{C\times 1}$. Since v satisfies 1-of-C coding scheme, there are C candidates to be the solution of Eq. (21). Thus we can do an exhaustive search to find out the solution of Eq. (21) as

$$\mathrm{v}^* = \mathrm{v}_c, \tag{22}$$

where c is obtained by setting

$$c = \arg\min_{\bar{c}} \sum_{m=1}^{M} \widetilde{d}^{(m)} \|\mathrm{x}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)} \mathrm{v}_{\bar{c}}\|_2^2. \tag{23}$$

We summarize the proposed algorithm in Algorithm 1.

Algorithm 1 Deep MF for multi-view data representation.

```
Input:
  Data set with $M$ views $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(M)}\}$ and $\mathbf{X}^{(m)} \in \mathbb{R}^{f^{(m)}\times n}$;
  The layer size $p$;
  The number of hidden structures $C$;
  Initialize the weight factor $\alpha^{(m)} = \frac{1}{M}$ for each view;
Output: $\mathbf{U}_i^{(m)}$ and $\mathbf{V}_i^{(m)}$ for each of the layers;
  Initialize:
  for all layers do
    $\mathbf{U}_i^{(m)}, \mathbf{V}_i^{(m)} \leftarrow k$-means$(\mathbf{V}_{i-1}$,layers$(i))$
  end for
  Initialize $\mathbf{V}_r \in \mathbb{R}^{n\times C}$, such that $\mathbf{V}_r$ satisfies the 1-of-$C$ coding scheme;
  Initialize $\mathbf{D}$ as defined in Eq. (12).
  repeat
    for all layers do
      1. Update$\widetilde{\mathbf{V}}_i^{(m)} = \begin{cases} \mathbf{V}_r & \text{if } i = r, \\ \mathbf{U}_{i+1}^{(m)}\widetilde{\mathbf{V}}_{i+1}^{(m)T} & \text{otherwise.} \end{cases}$
      2. Compute $\mathbf{\Phi} = \prod_{j=1}^{i} \mathbf{U}_j^{(m)}$.
      3. Compute the diagonal matrices $\widetilde{\mathbf{D}}^{(m)}$ according to Eq. (15);
      4. Update basis matrices $\mathbf{U}_i^{(m)}$ according to Eq. (16);
      5. Update $\mathbf{V}_i^{(m)}$ $(i < r)$ according to Eq. (19);
      6. Update each row of the last layer representation matrix one by one according to Eq. (23);
      7. Update the weight factor $\alpha^{(m)}$ according to Eq. (8);
    end for
  until Converges
```

3.4. Convergence analysis

To prove the convergence of the proposed algorithm, we need to utilize the lemma introduce by [34]:

Lemma 1. *For any positive real number a and b, the following inequality holds:*

$$\sqrt{a} - \frac{a}{2\sqrt{b}} \le \sqrt{b} - \frac{b}{2\sqrt{b}}. \tag{24}$$

Theorem 1. *In Algorithm 1, updated* $\mathbf{U}_i^{(m)}$ *and* $\mathbf{V}_i^{(m)}$ *will monotonically decrease the objective in Eq. (9) (i.e., Eq. (5)) until converge.*

Proof. Suppose the alternatively updated $\mathbf{U}_i^{(m)}$ and $\mathbf{V}_i^{(m)}$ are $\overline{\mathbf{U}}_i^{(m)}$ and $\overline{\mathbf{V}}_i^{(m)}$ in each iteration, respectively. It is clear that both $\overline{\mathbf{U}}_i^{(m)}$ and $\overline{\mathbf{V}}_i^{(m)}$ are closed form solutions according to Eqs. (16) and (23), considering the fixed weights $\alpha^{(m)} = 1\Big/\left(2\sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}\right)$ in current iteration, we have

$$\sum_{m=1}^{M} \frac{\|\mathbf{X}^{(m)} - \overline{\mathbf{U}}_1^{(m)}\overline{\mathbf{U}}_2^{(m)} \cdots \overline{\mathbf{U}}_r^{(m)}\overline{\mathbf{V}}_r^T\|_{2,1}}{2\sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}}$$
$$\le \sum_{m=1}^{M} \frac{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}{2\sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}}. \tag{25}$$

According to Lemma 1, we have

$$\sum_{m=1}^{M} \sqrt{\|\mathbf{X}^{(m)} - \overline{\mathbf{U}}_1^{(m)}\overline{\mathbf{U}}_2^{(m)} \cdots \overline{\mathbf{U}}_r^{(m)}\overline{\mathbf{V}}_r^T\|_{2,1}}$$
$$- \sum_{m=1}^{M} \frac{\|\mathbf{X}^{(m)} - \overline{\mathbf{U}}_1^{(m)}\overline{\mathbf{U}}_2^{(m)} \cdots \overline{\mathbf{U}}_r^{(m)}\overline{\mathbf{V}}_r^T\|_{2,1}}{2\sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}}$$
$$\le \sum_{m=1}^{M} \sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}$$
$$- \sum_{m=1}^{M} \frac{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}{2\sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}}. \tag{26}$$

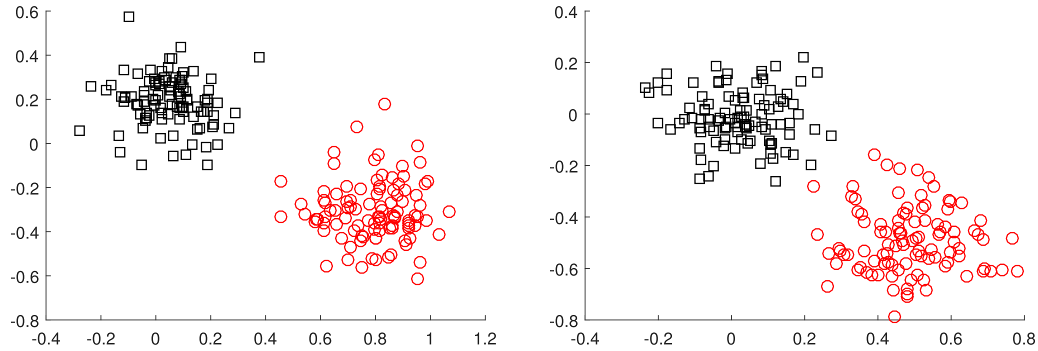
By summing over Eqs. (25) and (26) in the two sides, we obtain

$$\sum_{m=1}^{M} \sqrt{\|\mathbf{X}^{(m)} - \overline{\mathbf{U}}_1^{(m)}\overline{\mathbf{U}}_2^{(m)} \cdots \overline{\mathbf{U}}_r^{(m)}\overline{\mathbf{V}}_r^T\|_{2,1}}$$
$$\le \sum_{m=1}^{M} \sqrt{\|\mathbf{X}^{(m)} - \mathbf{U}_1^{(m)}\mathbf{U}_2^{(m)} \cdots \mathbf{U}_r^{(m)}\mathbf{V}_r^T\|_{2,1}}. \tag{27}$$

As a result, we prove that the objective value of current iteration is less than or equal to that of previous iteration. That is to say, the object is monotonically decreasing with iterations, and thus the convergence is certainly guaranteed. Finally, we complete the prove. □

3.5. Time complexity analysis

For the proposed method, it is composed of two stages including pre-training and fine-tuning. For convenience of analysis, we assume the dimensions of all the layers (i.e., layer size) are the same, namely p. The original feature dimensions for all the views are the same, namely f. M is the number of views. k is the number of layers. In pre-training stage, the time complexity of k-means process is $O(kt_1M(fnp + np^2 + pf^2))$, where t_1 is the iteration number of k-means. Similarly, the time complexity for the



(a) view-1

(b) View-2

Fig. 2. Illustration of the original toy data.

fine-tuning stag is $O(kt_2M(fnp+np^2+pf^2))$, where t_2 is the iteration number in this fine-tuning stage.

4. Experiments

Recent studies have focused on the case when dealing with multi-view learning tasks in the unsupervised setting, particularly, multi-view clustering. As mentioned before, the target data representation matrix $\mathbf{V}_r$ satisfies the 1-of-C coding scheme, thus the clustering partition result can be directly obtained without any post process, which is a major advantage for unsupervised learning. In our experiments, we will show the performance of the proposed method on both toy data and real-world benchmark datasets.

4.1. Experiments on toy data

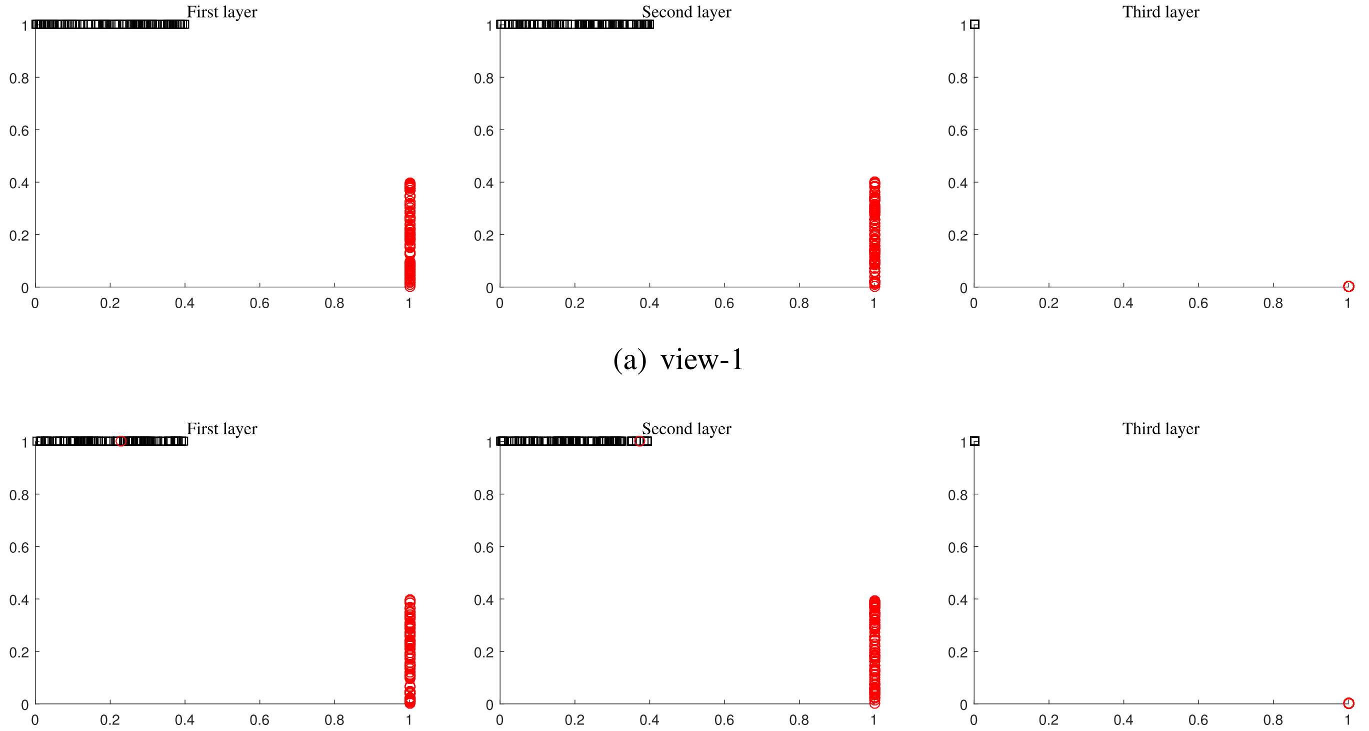
In this subsection, a randomly generated toy data is used to demonstrate our algorithms ability to make the data points more distinctive layer by layer. The toy dataset is comprised of two views and each view has 200 data points, which is grouped into two clusters, as shown in Fig. 2(a) and (b). The data points in each cluster are sampled from a Gaussian distribution with distinct center and variance.

We perform a three layer experiment on the toy dataset as an example. For convenience of demonstration, the layer size of all layers is set to two (i.e., p =[2 2 2]), that is, $\{V_1^{(1)}, V_2^{(1)}, V_3^{(1)}, V_1^{(2)}, V_2^{(2)}, V_3^{(2)}\} \in \mathbb{R}^{200\times2}$. Noting that $V_1^{(1)}$ denotes the first layer's representation matrix of the first view, as shown in the left subfigure of Fig. 3(a). As we can see, data points from the same class are forced to be closer layer by layer, for both of the two views. This demonstrates the effectiveness of the proposed method, which is able to uncover the hierarchical semantics of the input data in a layer-wise way. It is worth mentioning that, on view-2 (as shown in Fig. 3(b)), there is a data point misclassified. Fortunately, by leveraging the weight of each view as described in Eq. (8), the data point could be correctly classified, as the clustering capacity of view-1 is better than that of view-2. In our model, the representation matrix of last layer satisfies the 1-of-C coding scheme, thus ideally, the data points that belong to the same class would share the position with the same coordinate, as shown in the right subfigures in Fig. 3(a) and (b).

4.2. Experiments on benchmark data

To demonstrate the effectiveness of our method in terms of clustering performance, we compare the following state-of-the-art multi-view clustering methods:

- Co-trained multi-view clustering **(Co-train)** [11]: Co-train is proposed based on the assumption that the underlying common structure would assign a instance to the same class irrespective of the view. That is, if two instances are assigned in different clusters in one view, it should be so in all the views, and vice versa. It makes use of compatibility assumption of co-training.
- Co-regularized multi-view clustering **(Co-reg)** [10]: Co-reg enforces the view-specific eigenvectors of different views to look similar by regularizing them towards a common consensus. The objective combines the graph Laplacians of all views such that the latent structures resulting from each Laplacian look consistent.
- Multi-view k-means clustering **(MVKKM)** [35]: It performs the multi-view clustering task by utilizing unsupervised multiple kernel learning. All views in MVKKM are expressed by a given kernel matrices, and a weighted combination of the kernels is learned in parallel to the partitioning.
- Robust multi-view k-means clustering **(RMKMC)** [33]: This method is designed by extending the original single-view k-means clustering algorithm to a robust multi-view k-means clustering method.
- Multi-view NMF with local graph regularization **(MultiNMF)** [36]: MultiNMF is proposed based on manifold learning and multi-view NMF. A local graph regularization is constructed by taking the inner-view relatedness into consideration. With this local graph regularization, it is expected to integrate local geometrical information of each view.
- Multi-view clustering with adaptive structure concept factorization **(MVCF)** [37]: As an unsupervised multifeature learning method, MVCF aims to make the best utilization of the correlation among multiple features in an adaptive way. An affinity graph for each view is constructed, and the weight of affinity graph is learned by solving optimization problem.
- Self-weighted multi-view clustering with soft capped norm **(SCaMVC)** [5]: It is proposed to deal with different level noises and outliers by using soft capped norm. In SCaMVC, the residual of outliers is capped as a constant value and provides a probability for certain data point being an outlier.
- Latent multi-view subspace clustering **(LMSC)** [17]: Unlike most existing single-view subspace clustering methods, which directly reconstruct data points using original features, LMSC explores underlying complementary information from multiple views and simultaneously seeks the underlying latent representation. Using the complementarity of multiple views, the latent representation depicts data more comprehensively than each individual view, accord-



(a) view-1

(b) view-2

Fig. 3. Illustrations of each layer of each view. (a) view-1, (b) view-2.

ingly making subspace representation more accurate and robust.

- Multi-view clustering via deep semi-NMF **(DMVC)** [32]: DMVC applies the semi-nonnegative matrix factorization to exploit the hierarchical semantics of the input multi-view data. It introduces a graph regularizer into the objective by considering the intrinsic geometric information. This work is also one of the most recent multi-view clustering approaches based on a deep structure framework. Similar to our method, the goal of DMVC is to find a common representation matrix. Then it employs an additional clustering algorithm (e.g., spectral clustering algorithm) on the common representation to obtain the final clustering results.

In addition, the classic clustering algorithm, *k*-means (KM), is included as baseline. We apply KM on every dataset using each view of features (e.g., KM(1) means performing KM on the 1st view). We also apply KM using concatenated features of all views with equal weight (i.e., All features, AllFea in short), which assumes that all the views are of the same importance to the clustering task.

4.3. Data sets

Several benchmark datasets including image datasets and document datasets are used to assess the performance of our method. The details of these datasets are given below:

- **Handwritten (HW):** [2] This dataset contains 2000 instances for 0–9 ten digit classes, which is collected from two sources, i.e., USPS Handwritten Digits and MNIST Handwritten Digits.
- **3 sources:** [3] It consists of 169 news collected from BBC, Reuters and The Guardian. The topic of each news is associated with one of the following six topical labels: technology, health, business, politics, entertainment, and sport.
- **BBC** [4] [38]: BBC is derived from the BBC news corpora. It contains of 685 documents, and each document is split into four segments. The topic of each news is associated with one of five topical labels.
- **BBCSport** [5] [38]: It consists of 544 documents collected from the BBC Sport website. Each document was split into two segments. Similar to BBC, the topic of each news is associated with one of five topical labels.
- **CiteSeer** [6] [39]: This dataset contains 3312 documents over the 6 labels. Each document is desribed by content view and citations view. In detail, the documents are described by 3703 words in the content view, and 4732 links between them in the inbound, outbound and cites views.
- **Reuters** [7] [39]: The archive contains 6 samples of 1200 documents. Each document is described by five views of 2000 words each. The documents were initially in English, other four views correspond to the words of their traductions in Italian, French, German, and Spanish, respectively.

4.4. Experiment setup

For each data set, the number of clusters is set to the true number of classes for all methods. In our model, inspired by Zhao et al. [32], the layer sizes are empirically searched in [100 *C*], [50 *C*],

[2] https://cs.nyu.edu/roweis/data.html.

[3] http://mlg.ucd.ie/datasets/3sources.html.

[4] http://mlg.ucd.ie/datasets/segment.html.

[5] http://mlg.ucd.ie/datasets/segment.html.

[6] http://lig-membres.imag.fr/grimal/data.html.

[7] http://lig-membres.imag.fr/grimal/data.html.

Table 1
Clustering performance on HW (%).

Method	ACC	NMI	Purity
KM(1)	54.89(3.19)	48.92(1.02)	57.29(2.30)
KM(2)	44.75(2.42)	44.24(1.63)	47.76(2.09)
KM(AllFea)	61.05(5.26)	59.67(3.32)	65.20(4.23)
Co-train	50.33(0.46)	38.38(0.83)	50.93(0.18)
Co-reg	55.15(4.31)	47.91(3.27)	57.93(3.36)
MVKKM	49.82(2.79)	48.58(0.76)	55.33(1.38)
RMKMC	78.30(1.49)	76.71(1.50)	83.55(0.57)
MultiNMF	68.95(2.97)	62.65(2.66)	68.98(2.93)
MVCF	64.78(5.90)	60.72(0.04)	65.58(4.77)
SCaMVC	75.35(0.78)	74.49(1.27)	79.40(1.23)
LMSC	73.40(0.45)	63.19(1.02)	75.85(0.65)
DMVC	52.17(3.57)	50.20(6.24)	61.55(2.93)
Ours	**80.35(7.74)**	**83.48(8.38)**	**85.25(7.71)**

Table 2
Clustering performance on 3Sources (%).

Method	ACC	NMI	Purity
KM(1)	37.81(2.97)	14.35(4.08)	41.01(4.58)
KM(2)	40.41(7.56)	16.67(9.06)	43.14(8.33)
KM(3)	40.89(6.86)	18.34(8.63)	44.02(7.42)
KM(AllFea)	41.78(7.99)	20.71(9.87)	46.92(9.40)
Co-train	33.14(0.01)	10.04(0.37)	34.91(0.01)
Co-reg	32.55(1.68)	12.56(0.76)	36.69(0.84)
MVKKM	41.22(5.17)	18.10(8.61)	43.39(6.49)
RMKMC	34.71(0.68)	14.23(1.04)	36.29(0.35)
MultiNMF	45.76(4.52)	37.67(4.80)	59.76(3.29)
MVCF	53.61(4.60)	40.40(7.76)	55.45(4.60)
SCaMVC	49.70(4.53)	28.54(3.03)	57.99(2.34)
LMSC	41.42(2.37)	23.90(1.12)	51.48(1.90)
DMVC	43.79(0.42)	32.62(0.73)	**61.55(1.25)**
Ours	**55.03(0.83)**	**41.82(4.27)**	60.36(2.09)

Table 3
Clustering performance on BBC (%).

Method	ACC	NMI	Purity
KM(1)	38.74(6.63)	18.63(8.93)	39.74(6.73)
KM(2)	35.40(3.29)	13.81(3.72)	36.20(3.19)
KM(3)	36.76(4.37)	16.39(5.18)	38.22(4.24)
KM(4)	35.87(4.87)	15.21(5.16)	36.35(4.69)
KM(AllFea)	37.01(3.35)	16.10(5.96)	37.46(3.17)
Co-train	32.70(0.01)	10.93(0.01)	33.14(0.01)
Co-reg	39.27(1.34)	11.04(0.24)	34.03(0.21)
MVKKM	40.53(4.39)	17.39(3.57)	41.36(4.99)
RMKMC	32.60(1.22)	11.64(0.47)	33.48(0.34)
MultiNMF	46.13(2.13)	24.45(2.92)	46.28(1.97)
MVCF	**61.75(4.00)**	37.77(5.03)	61.90(3.94)
SCaMVC	50.22(1.73)	19.27(0.91)	50.38(2.18)
LMSC	38.39(0.85)	11.47(0.34)	48.03(3.63)
DMVC	48.18(1.30)	18.91(1.25)	46.06(2.32)
Ours	57.34(7.70)	**39.53(6.21)**	**68.85(8.70)**

Table 4
Clustering performance on BBCSport (%).

Method	ACC	NMI	Purity
KM(1)	37.94(4.30)	16.71(3.67)	40.18(3.45)
KM(2)	37.81(5.12)	16.04(4.71)	40.29(4.55)
KM(AllFea)	41.56(6.41)	20.24(7.40)	43.22(6.14)
Co-train	38.42(0.76)	15.69(0.79)	41.54(2.14)
Co-reg	29.41(0.21)	12.78(0.40)	36.21(0.10)
MVKKM	38.79(1.66)	17.04(2.05)	36.40(1.21)
RMKMC	41.73(4.20)	18.29(5.98)	42.28(3.66)
MultiNMF	53.74(3.77)	32.71(5.25)	54.72(4.51)
MVCF	61.40(1.84)	37.93(2.52)	61.58(1.84)
SCaMVC	41.54(2.13)	18.27(2.09)	42.28(1.98)
LMSC	38.79(3.47)	20.16(1.43)	43.38(1.85)
DMVC	41.47(2.34)	23.35(2.69)	49.15(2.21)
Ours	**65.33(5.43)**	**41.52(5.30)**	**62.62(3.37)**

Table 5
Clustering performance on Reuters (%).

Method	ACC	NMI	Purity
KM(1)	24.22(5.46)	17.52(4.46)	24.48(5.38)
KM(2)	20.65(5.77)	14.09(5.35)	20.92(5.74)
KM(3)	19.12(3.44)	12.45(2.98)	19.32(3.45)
KM(4)	23.38(6.88)	15.67(5.49)	23.68(6.82)
KM(5)	23.08(6.58)	15.55(5.34)	23.48(6.72)
KM(AllFea)	23.11(6.93)	16.15(5.61)	23.45(6.89)
Co-train	23.44(2.36)	15.58(2.66)	23.48(2.31)
Co-reg	22.51(0.59)	12.58(0.07)	22.81(0.48)
MVKKM	21.53(1.61)	11.37(3.11)	22.46(1.64)
RMKMC	34.33(3.69)	15.29(4.35)	34.33(3.69)
MultiNMF	41.17(1.79)	23.45(0.87)	45.83(2.25)
MVCF	–	–	–
SCaMVC	30.00(5.69)	20.62(4.34)	30.17(5.73)
LMSC	–	–	–
DMVC	–	–	–
Ours	**46.18(1.93)**	**27.83(1.14)**	**49.82(1.65)**

Table 6
Clustering performance on CiteSeer (%).

Method	ACC	NMI	Purity
KM(1)	21.88(1.12)	10.92(1.93)	22.06(1.08)
KM(2)	36.13(5.30)	12.98(3.43)	37.18(5.41)
KM(AllFea)	39.08(6.94)	15.84(4.37)	39.95(5.75)
Co-train	25.99(0.45)	12.13(0.12)	27.87(0.77)
Co-reg	21.26(0.03)	10.31(0.13)	21.32(0.00)
MVKKM	23.15(0.60)	11.62(0.23)	23.74(0.79)
RMKMC	22.47(0.30)	11.25(0.18)	21.66(0.10)
MultiNMF	35.11(5.11)	17.76(2.34)	38.76(3.16)
MVCF	46.58(0.63)	20.30(0.80)	48.34(0.43)
SCaMVC	22.74(0.73)	11.73(0.56)	23.13(0.83)
LMSC	28.14(0.04)	16.25(0.17)	32.67(1.25)
DMVC	24.57(0.26)	12.23(0.78)	26.73(1.41)
Ours	**51.33(1.08)**	**29.51(1.03)**	**53.15(3.68)**

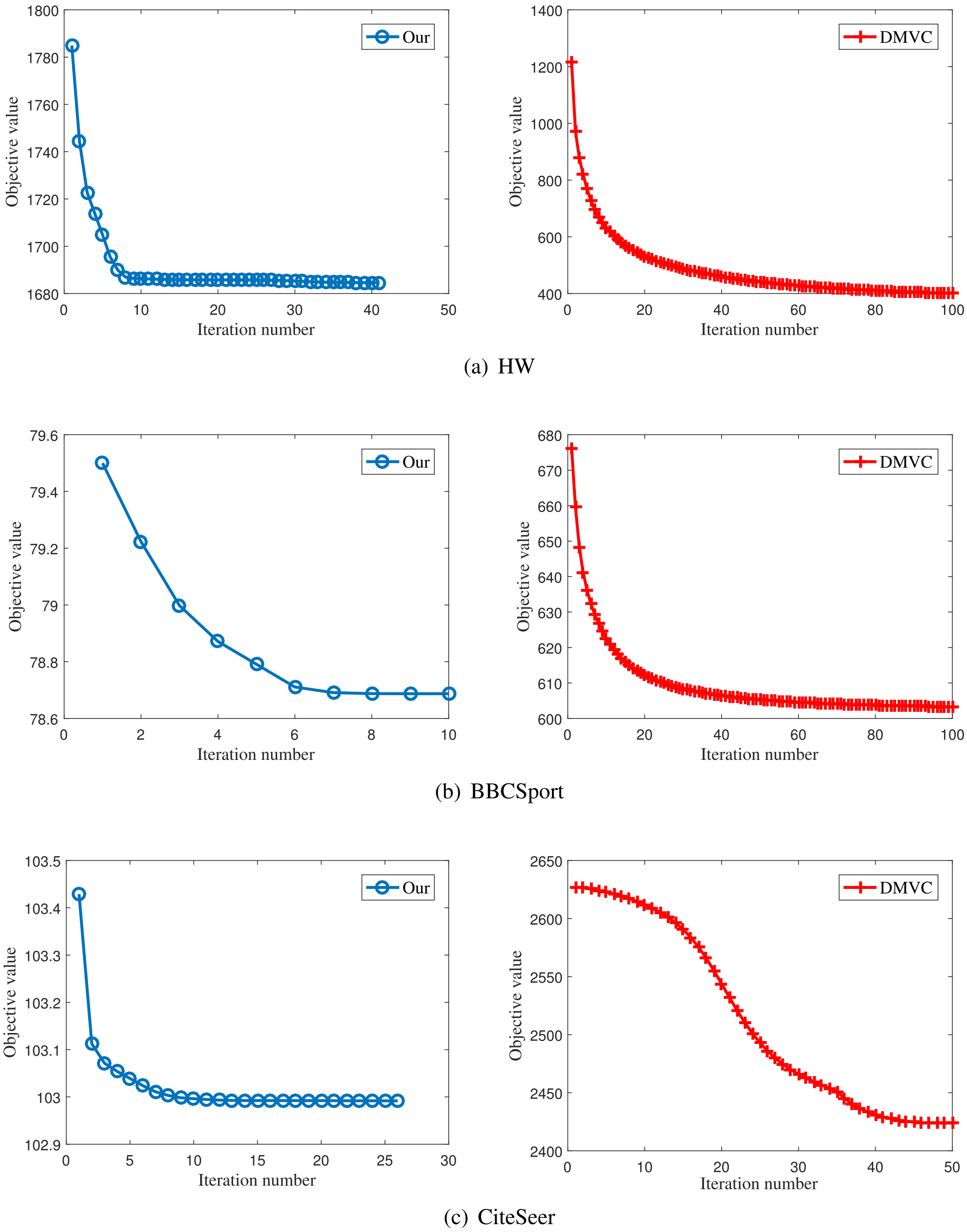
[100 50 C] for simplicity, where C is the number of clusters. For compared methods, we obtained the source code from its authors' website, and we set their parameters to the optimal value if they have. In order to randomize the experiments, we repeat the experiments 10 times, and the means and standard deviations are reported.

Three standard metrics accuracy (ACC), normalized mutual information (NMI) and Purity are used to evaluate the experimental performance. These metrics are positive correlated, a larger value represents a better results. The best mean and standard deviation of the results are highlighted in boldface.

4.5. Experimental results

The clustering performance measured by ACC, NMI and Purity are shown in Tables 1–6. As we can see, our method is very competitive, outperforms other compared methods in most cases. Specifically, the superiority of the proposed method arises the following aspects:

- It can be observed that the clustering performance of KM(AllFea) (i.e., performing k-means on the concatenated features of all views) is better than that obtained on a single view for most of the times. This indicates that the clustering performance can be enhanced by considering all information hidden in data, even just simply concatenates the features of all views with equal weight.
- The clustering performance obtain by multi-view clustering approaches, including ours, generally outperforms baseline method on both each individual view and the concatenated features. This states that the clustering quality can be further improved by leveraging the consistent and complemen-



(a) HW

(b) BBCSport

(c) CiteSeer

Fig. 4. The convergence curve of DMVC and our method. (a) HW, (b) BBCSport, (c) CiteSeer.

tary information shared by different views. With the help of multi-view clustering formulations, the learned weight for a good view is large while a relatively small weight will be assigned to a weak view. This is the core reason why multi-view learning methods are more efficient than traditional single-view methods.

- The results of the proposed method are better than the other compared methods on most datasets, thus demonstrates the effectiveness of our method, which is based on deep matrix decomposition framework. Considering the mapping between the obtained representation and the original data contains rather complex hierarchical information with implicit lower-level hidden attributes, our model is able to learn a better representation by uncovering the hierarchical semantics of the input data in a layerwise way. By introducing the deep framework, we expect to fully extracted the hierarchical information in general case. Noting that the dataset Resters is very sparse (contains only a few nonzero elements in data matrix). This may be the reason why MVCF, LMSC and DMVC cannot handle this dataset, as recorded in Table 5.

4.6. Parameter tuning

Our model has only one parameter, i.e., the layer size p. In this subsection, we investigate the sensitivity of our model with respect to this parameter, as shown in Table 7.

As we can see, our model is very stable when p is searched in [100 C], [50 C], [100 50 C]. This indicates the proposed model is not sensitive to p. Relatively speaking, tuning the layer size to

Table 7
The clustering accuracy with respect to different layer size.

Data set	HW	BBCSport	Reuters	CiteSeer
[50 C]	80.35	65.33	44.92	48.69
[100 C]	75.95	53.89	37.83	51.33
[100 50 C]	80.10	57.21	46.18	44.98

[50 C] may be a better choice, as the model consistently achieves good performance under this parameter setting.

4.7. Convergence study

We have proven that the updating rules for optimizing the proposed objective are convergent, as decribed in last section. Here we empirically investigate how fast these rules can converge. For comparison, the convergence curve of another deep multi-view clustering method, DMVC, is also recorded. For our model, we record the value of our objective as introduced in Eq. (9). While for DMVC, we report the value of Eq. (3) as described in [32].

The convergence curves of our model and DMVC are shown in Fig. 4. Noting that when the optimization process converges or the number of iterations reaches the predefined maximum value (which is empirically set to 100 in our experiment), the proposed algorithm stops the iterative process. It can be seen that the updating rules for our model converge very fast. Compared with DMVC, our model needs less iteration to achieve the convergence, which demonstrates the effectiveness of proposed optimization algorithm. Note that the measurement of residual calculation is different in our model and DMVC (we use $l_{2,1}$-norm, while DMVC uses the square of l_2-norm), thus it is natural that the value of the two objectives may differ significantly. Fortunately, the convergence speed is irrelevant to the objective value, as it is typically depended on update rules of the optimization algorithm.

5. Conclusion

In this paper, a novel deep multi-view clustering model is proposed by uncovering the hierarchical semantics of the input data in a layerwise way. By utilizing the deep matrix decomposition framework, the hidden representations are learned with respect to different attributes. We derive an efficient updating algorithm to solve the optimization problem and its convergence is also guaranteed theoretically. Furthermore, our model is able to automatically assign the optimal weight to each view without introducing extra hyperparameter as previous methods do. Experimental results on both toy and benchmark data demonstrate the efficacy of the proposed algorithm. Considering that our model is essentially an unsupervised algorithm that does not make use of the priori information, it would be interesting to extend the proposed framework to other machine learning areas such as semi-supervised learning and classification problems. In the future, we are also interested in studying how to extend the multi-view learning idea to nonlinear latent subspace cases.

Acknowledgments

This work was partially supported by NSF China (project no. 61572111) and Research Fund for the Central Universities of China (project nos. ZYGX2016Z003, ZYGX2017KYQD177).

References

[1] T. Ojala, Pietik, M. Inen, Topi, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach.Intell. 24 (7) (2002) 971–987.

[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[3] D.G. Lowe, D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.

[4] S. Huang, Y. Ren, Z. Xu, Robust multi-view data clustering with multi-view capped-norm k-means, Neurocomputing 311 (2018) 197–208.

[5] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, Knowl.-Based Syst. 158 (2018) 1–8.

[6] Z. Kang, Z. Guo, S. Huang, S. Wang, W. Chen, Y. Su, Z. Xu, Multiple partitions aligned clustering, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 2701–2707.

[7] Y. Li, M. Yang, Z.M. Zhang, A survey of multi-view representation learning, IEEE Trans. Knowl. Data Eng. (2018) 1–20.

[8] S. Hou, L. Chen, D. Tao, S. Zhou, W. Liu, Y. Zheng, Multi-layer multi-view topic model for classifying advertising video, Pattern Recognit. 68 (2017) 66–81.

[9] S. Huang, Z. Kang, I.W. Tsang, Z. Xu, Auto-weighted multi-view clustering via kernelized graph learning, Pattern Recognit. 88 (2019) 174–184.

[10] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Advances in Neural Information Processing Systems, 2012, pp. 1413–1421.

[11] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 393–400.

[12] J. Liu, C. Wang, J. Gao, J. Han, Multi-view clustering via joint nonnegative matrix factorization, in: Proceedings of the 2013 SIAM International Conference on Data Mining, 2013, pp. 252–260.

[13] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, B. Du, Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding, Pattern Recognit. 48 (10) (2015) 3102–3112.

[14] Z. Ding, Y. Fu, Low-rank common subspace for multi-view learning, in: Proceedings of the IEEE International Conference on Data Mining, 2014, pp. 110–119.

[15] P. Yang, W. Gao, Information-theoretic multi-view domain adaptation: a theoretical and empirical study, J. Artif. Intell. Res. 49 (2014) 501–525.

[16] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 129–136.

[17] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, D. Xu, Generalized latent multi--view subspace clustering, IEEE Trans. Pattern Anal. Mach.Intell. (2018) 1–14.

[18] S.Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, in: Proceedings of the European Conference on Computer Vision, 2002, pp. 67–81.

[19] X. Zhang, J. Cheng, C. Xu, H. Lu, S. Ma, Multi-view multi-label active learning for image classification, in: Proceedings of the IEEE International Conference on Multimedia and Expo, 2009, pp. 258–261.

[20] P. Dhillon, D.P. Foster, L.H. Ungar, Multi-view learning of word embeddings via CCA, in: Advances in Neural Information Processing Systems, 2011, pp. 199–207.

[21] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562.

[22] S. Huang, P. Zhao, Y. Ren, T. Li, Z. Xu, Self-paced and soft-weighted nonnegative matrix factorization for data representation, Knowl.-Based Syst. 164 (2019) 29–37.

[23] S. Huang, Z. Xu, F. Wang, Nonnegative matrix factorization with adaptive neighbors, in: Proceedings of the International Joint Conference on Neural Networks, 2017.

[24] S. Huang, H. Wang, T. Li, T. Li, Z. Xu, Robust graph regularized nonnegative matrix factorization for clustering, Data Min. Knowl. Discov. 32 (2) (2018) 483–503.

[25] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 126–135.

[26] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B. Schuller, A deep semi-nmf model for learning hidden representations, in: Proceedings of the International Conference on Machine Learning, 2014, pp. 1692–1700.

[27] D. Kong, C. Ding, H. Huang, Robust nonnegative matrix factorization using l21-norm, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 673–682.

[28] I. Daubechies, R. DeVore, M. Fornasier, C.S. Güntürk, Iteratively reweighted least squares minimization for sparse recovery, Commun. Pure. Appl. Math. 63 (1) (2010) 1–38.

[29] F. Nie, J. Li, X. Li, Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 1881–1887.

[30] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 1247–1255.

[31] W. Wang, R. Arora, K. Livescu, J. Bilmes, On deep multi-view representation learning, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 1083–1092.

[32] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in: Proceedings of AAAI Conference on Artificial Intelligence, 2017, pp. 2921–2927.

[33] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2013, pp. 2598–2604.

[34] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization, in: Advances in Neural Information Processing Systems, 2010, pp. 1813–1821.
[35] G. Tzortzis, A. Likas, Kernel-based weighted multi-view clustering, in: Proceedings of the 12th IEEE International Conference on Data Mining, 2012, pp. 675–684.
[36] Z. Wang, X. Kong, H. Fu, M. Li, Y. Zhang, Feature extraction via multi-view non-negative matrix factorization with local graph regularization, in: Proceedings of the IEEE International Conference on Image Processing, 2015, pp. 3500–3504.
[37] K. Zhan, X. Chang, J. Guan, L. Chen, Z. Ma, Y. Yang, Adaptive structure discovery for multimedia analysis using multiple features, IEEE Trans. Cybern. (2018) 1–9.
[38] D. Greene, P. Cunningham, Practical solutions to the problem of diagonal dominance in kernel document clustering, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 377–384.
[39] G. Bisson, C. Grimal, An architecture to efficiently learn co-similarities from multi-view datasets, in: Proceedings of the International Conference on Neural Information Processing, 2012, pp. 184–193.

Shudong Huang Mr. Huang is currently a Ph.D. student in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. He received the M.Sc. degree in the School of Information Science and Technology from Southwest Jiaotong University, China, in 2015. His research interests include semi-supervised learning and ensemble learning. He is a student member of CCF.

Zhao Kang Dr. Kang received his Ph.D. degree from the Department of Computer Science at Southern Illinois University in May 2017. At present, Dr. Kang focuses on the research of theory and method design in machine learning, as well as applying those methods on practical problems in compute vision, social network, information retrieval and data mining. He published more than 20 papers in top conferences and journals in the area of artificial intelligence and data mining, including AAAI, IJCAI, SIGKDD, TISTM, TKDD, Neurocomputing. At the same time, he has been invited to serve as reviewer and member of the Procedural Committee for the top journals and conferences in related fields for times.

Zenglin Xu Dr. Xu is currently a full professor in University of Electronic Science & Technology of China. He received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong. He has been working at Michigan State University, Cluster of Excellence at Saarland University and Max Planck Institute for Informatics, and later Purdue University. Dr. Xu's research interests include machine learning and its applications in information retrieval, health informatics, and social network analysis. He currently serves as an associate editor of Neural Networks, Neurocomputing and Big Data Analytics. He is the recipient of the outstanding student paper honorable mention of AAAI 2015, the best student paper runner up of ACML 2016, and the 2016 young researcher award from APNNS.