Information Sciences 610 (2022) 114–125

Contents lists available at ScienceDirect

Information Sciences

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

Diversity embedding deep matrix factorization for multi-view clustering

Zexi Chen [a,b], Pengfei Lin [a,b], Zhaoliang Chen [a,b], Dongyi Ye [a,b], Shiping Wang [a,b,*]

[a] College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China
[b] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China

ARTICLE INFO

Article history:
Received 21 March 2022
Received in revised form 28 July 2022
Accepted 30 July 2022
Available online 5 August 2022

Keywords:
Multi-view clustering
Deep matrix factorization
Deep learning
Diversity embedding

ABSTRACT

Multi-view clustering has attracted increasing attention by reason of its ability to leverage the complementarity of multi-view data. Existing multi-view clustering methods have explored nonnegative matrix factorization to decompose a matrix into multiple matrices for feature representations from multi-view data, which are not discriminative enough to deal with the natural data containing complex information. Moreover, most of multi-view clustering methods prioritize the consensus information among multi-view data, leaving a large amount of information redundant and the clustering performance deteriorated. To address these issues, this paper proposes a multi-view clustering framework that adopts a diversity loss for deep matrix factorization and reduces feature redundancy while obtaining more discriminative features. We then bridge the relation between deep auto-encoder and deep matrix factorization to optimize the objective function. This method avoids the challenges in the optimization process. Extensive experiments demonstrate that the proposed method is superior to state-of-the-art methods.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Clustering is an important field in unsupervised learning. Natural data usually contains a lot of redundant information, therefore researches in massive numbers optimize clustering performance by learning more discriminative features.

Many studies have verified the effectiveness of matrix factorization in clustering tasks [1,2]. Matrix factorization compresses the original high-dimensional data by finding a set of basis to improve the performance of many machine learning tasks, such as matrix completion [3,4], recommender systems [5,6], information retrieval [7], community detection [8,9] and image recognition [10]. Nonnegative matrix factorization (NMF) is one of the most widely utilized dimensionality reduction techniques, which decomposes a nonnegative data matrix into two nonnegative matrices. However, some characteristics limit the performance of NMF. One of the characteristics is that NMF requires the original data matrix to be nonnegative, whose application scenarios are relatively limited. To address this limitation, semi-nonnegative matrix factorization (Semi-NMF) yields nonnegative factors but does not require that the input data matrix is nonnegative [11]. Data representations achieved in this way are usually mixed with complex information, which may become an obstacle when applying

* Corresponding author at: College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China.
E-mail addresses: chenzexi@vip.126.com (Z. Chen), linpfmail@163.com (P. Lin), chenzl23@outlook.com (Z. Chen), yiedy@fzu.edu.cn (D. Ye), shipingwangphd@163.com (S. Wang).

https://doi.org/10.1016/j.ins.2022.07.177
0020-0255/© 2022 Elsevier Inc. All rights reserved.

Semi-NMF or NMF to clustering problems. Deep matrix factorization (DMF) has been proposed to handle this issue [12,13], which employs a multi-layer structure to learn a nonnegative hidden representation of the original data.

In recent years, a mass of data described in multiple feature representations has appeared. For example, heterogeneous features can be extracted from social images. Plenty of researches have indicated that taking advantage of the complementarity among multiple views could improve the performance of many machine learning tasks, such as semi-supervised learning [14,15], transfer learning [16,17], and discriminant analysis [18,19]. Multi-view clustering by leveraging the complementarity among data in multiple views generally outperforms a single-view clustering method [20,21]. Therefore, multi-view clustering has caused widespread attention [22,23]. Recently, NMF has been applied to multi-view clustering [24,25]. Some prior multi-view clustering methods leveraged deep matrix factorization to provide more informative hidden representations for clustering. Huang et al. [26] developed a deep NMF model to learn the hidden representations associate with different implicit lower-level attributes. Zhao et al. [27] utilized the consensus matrix as a constraint to enforce multi-view data so as to share the same representation after multi-layer factorization, and constructed multiple graphs to preserve the geometric structure in each view. Wei et al. [28] generated cluster assignments in each layer and achieved diversity by minimizing a redundancy quantification term. Chen et al. [29] obtained partition representations of each view and jointly exerted them with the most desirable representation for multi-view clustering tasks. Nevertheless, most of multi-view matrix factorization methods for clustering methods leveraged the consensus representation matrix to obtain similarities among multiple views. These approaches put more emphases on the shared information among multiple views, which may lead to several limitations. For instance, data representations achieved by these methods contain mutually redundant information and lack diversity. Wang et al. [30] used a diversity term to reduce the redundancy and enhance diversity among multi-view representations.

At present, matrix factorization has revealed its effectiveness in clustering problems, but there are still some limitations in the objective function optimization. A major limitation of NMF is that it fails to effectively handle non-linear data. The methods based on NMF and DMF suffer from some challenges during the optimization process, such as nonnegativity and orthogonality. Rising to this challenge, we optimize our objective function through deep auto-encoders. Deep auto-encoder is a significant branch of deep learning to utilize artificial neural networks with representation learning, and it is suitable for modeling complex non-linear relations. By adding non-linear activation functions [31], auto-encoders achieved a more powerful representation ability. Therefore, deep auto-encoders and their variants have been widely applied to learning effective latent features [32,33].

This paper proposes a novel multi-view clustering method named diversity embedding deep matrix factorization for multi-view clustering, abbreviated as DDMF. The proposed method performs deep matrix factorization on the data matrix to achieve a more discriminative latent data representation for each view. At the same time, a diversity term is adopted among the learned data representations of different views to increase diversity and reduce redundancy. For more effective optimization, deep matrix factorization is transformed into deep neural networks with reconstruction error and constraints on the hidden layer, and activation functions are utilized to ensure nonnegative constraints. Taking two views as an example, the overall framework of the proposed method is illustrated in Fig. 1. The main contributions of this paper can be summarized as follows.

- A deep matrix factorization based multi-view clustering model is proposed, which adopts diversity loss to reduce redundancy and enhance diversity of data representations.
- By bridging the relation between deep matrix factorization and deep auto-encoders, the objective function is optimized while preserving nonnegative constraints of outputs.
- Extensive experiments were conducted on six real world datasets to demonstrate the effectiveness of the proposed method against existing state-of-the-arts.

2. Related work

Multi-view clustering aims to exploit complementary information from multi-view data to achieve better clustering performance. Among plenty of models, graph-based and subspace-based models have generated substantial publicity [34]. Graph-based models aim to apply the similarity matrix and typical spectral clustering to obtain the final clusters. Wang et al. [35] proposed a framework that learned each view graph matrix and a unified matrix in a mutual reinforcement manner, and automatically weighted each data graph matrix to derive the unified matrix, where a rank constraint is imposed on the Laplacian matrix. Kang et al. [36] performed graph fusion and spectral clustering simultaneously, and the fusion graph approximated the original graph of each individual view while maintaining an explicit cluster structure. Huang et al. [37] performed a multi-view clustering task while simultaneously learning similarity relationships with multiple kernel learning ability. Qiang et al. [38] efficiently generated representative anchors and constructed anchor graphs on multiple views to directly solve the spectral clustering problem. Constructing a more accurate similarity graph, automatically allocating optimal weight for each view, and finding the cluster indicator are simultaneously accomplished.

Furthermore, subspace-based models learning a new data representation are more applicable to clustering tasks from multiple data representations. Wang et al. [39] harnessed the complementary information between distinct representations by introducing a position-aware exclusivity term. Chen et al. [40] adopted the least square regression to learn global consensus information shared by multiple views and the column-sparsity norm to measure the residual information. Lv et al. [41]

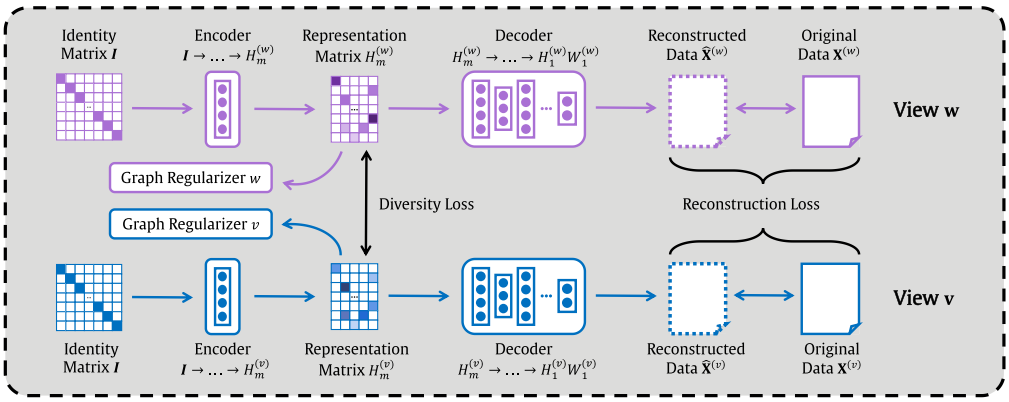

Fig. 1. An overview of the proposed framework. An identity matrix is utilized as the input of the network of each view, and the data representation of the view is output in the first layer. After that, the graph regularizer of each view is constructed, and the diversity loss between the two distinct views is constructed. The output of the decoder is utilized to define the reconstruction loss.

learned a view-specific representation from data by exploiting the local structure within each view, then generated a low-rank tensor representation from the view-specific representation to capture a high-order correlation across multiple views. Wang et al. [42] provided a method to solve the problem of higher-order matrix decomposition by transforming it into a task with unitary invariance. The matrix factorization based model is a particular category in the multi-view subspace clustering. Liu et al. [43] added the measurement of inconsistency between the latent representation of each view and the consensus matrix to the objective function to realize multi-view information sharing. Zong et al. [25] preserved the locally geometrical structure of the multi-view data space by leveraging consensus manifold and coefficient matrix with multi-manifold regularization. Wang et al. [44] extended concept factorization to deep concept factorization. Huang et al. [45] improved the way of weight assignments so that the weight of each view is automatically assigned without introducing extra hyperparameters. Nevertheless, most of the above methods preferred to learn an ordinary representation, which might not efficiently exploit complementary information among various views. Multi-view clustering methods reducing redundancy and obtaining more complementary data representations need to be developed urgently.

3. The proposed method

In this section, we will first review the deep matrix factorization, then a multi-view clustering method called diversity embedding deep matrix factorization, named DDMF is proposed. The method intends to leverage deep matrix factorization to learn latent data representations that retain information more applicable for clustering while enhancing the diversity of data representations of distinct views. We propose an auto-encoder based method to recover the objective function with nonnegative constraints. To facilitate understanding the following calculation process, we summarize the main notations in Table 1.

3.1. Revisiting deep matrix factorization

NMF aims to find two nonnegative matrices $\mathbf{U}$ and $\mathbf{V}$, and minimizes the following objective function:

$$\min_{\mathbf{U}\geqslant 0,\mathbf{V}\geqslant 0} \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2. \tag{1}$$

Thus, NMF tries to search for a compressed approximation of the original data matrix, i.e., $\mathbf{X}\approx\mathbf{U}\mathbf{V}^T$. Semi-NMF extends NMF by relaxing the basis matrix $\mathbf{U}$ to real values while requiring $\mathbf{V}$ to be nonnegative.

Natural data contains a variety of information, and a single Semi-NMF is affected by other information, therefore it fails to extract category information well. Applying deep semi-NMF helps to solve this problem relatively well and achieved acceptable results in data representation [12]. The multi-layer decomposition process is defined as follows:

$$\begin{aligned}
\mathbf{X} &\approx \mathbf{Z}_1\mathbf{H}_1^+,\\
\mathbf{X} &\approx \mathbf{Z}_1\mathbf{Z}_2\mathbf{H}_2^+,\\
&\vdots\\
\mathbf{X} &\approx \mathbf{Z}_1\cdots\mathbf{Z}_m\mathbf{H}_m^+,
\end{aligned} \tag{2}$$

Table 1
Main notations of the DDMF method.

Symbols	Description
$\mathbf{X}^{(v)}$	the data matrix in the v-th view
$\mathbf{Z}_i^{(v)}$	the i-th layer weight matrix of DMF of the v-th view
$\mathbf{H}_i^{(v)}$	the i-th layer output matrix of DMF of the v-th view
$\mathbf{H}^{(v)}$	the representation matrix of DMF of the v-th view
$\mathbf{h}_i^{(v)}$	the i-th data point of DMF of the v-th view
$\sigma(\cdot)$	activation function (relu/tanh/...)
$\mathbf{E}^{(v)}$	the weight of the v-th view encoder
$\mathbf{C}^{(v)}$	the output of the v-th view decoder
$\mathbf{W}_i^{(v)}$	the i-th layer weight of the v-th view decoder
$\mathbf{C}_i^{(v)}$	the i-th layer output of the v-th view decoder

where $\mathbf{Z}_i$ denotes the i-th layer basis matrix, and $\mathbf{H}_i^{+}$ is the i-th layer representation matrix. Trigeorgis et al. [12] experimented that deep matrix factorization could learn better latent representations than single-layer Semi-NMF according to the attribute with the lowest variability in the data.

3.2. The objective function

Given a V-view multi-view dataset $\mathscr{X} = \left\{\mathbf{X}^{(v)}\right\}_{v=1}^{V}$ with $\mathbf{X}^{(v)} \in \mathbb{R}^{d^{(v)} \times n}$, where $d^{(v)}$ is the feature dimension of the v-th view and n denotes the number of samples. Deep matrix factorization can be extended to multi-view setting, where $\mathbf{H}^{(v)} \in \mathbb{R}^{d \times n}$ denotes the representation matrix in the v-th view. Deep matrix factorization is performed on feature matrices of distinct views. The i-th layer weight matrix of the v-th view is denoted as $\mathbf{Z}_i^{(v)}$. And combining these decomposition processes, the objective function can be formulated as follows:

$$\min_{\mathbf{Z}_i^{(v)}, \mathbf{H}_i^{(v)}, \mathbf{H}^{(v)}} \sum_{v=1}^{V} \left\| \mathbf{X}^{(v)} - \mathbf{Z}_1^{(v)} \mathbf{Z}_2^{(v)} \cdots \mathbf{Z}_m^{(v)} \mathbf{H}^{(v)} \right\|_F^2, s.t. \mathbf{H}_i^{(v)} \geq 0, \mathbf{H}^{(v)} \geq 0, \tag{3}$$

where the objective function can be regarded as a reconstruction error between the original data matrix $\mathbf{X}^{(v)}$ and the reconstructed data matrix $\mathbf{Z}_1^{(v)} \mathbf{H}_1^{(v)}$ in each view. And $\mathbf{H}_i^{(v)}$ is the output of the i-th layer and $\mathbf{H}^{(v)} = \mathbf{H}_m^{(v)}$ as Eq. (2).

To exploit more comprehensive information, it is essential to guarantee the diversity of the representation matrices extracted from the multi-view data, which requires the orthogonality of the feature vectors of the same sample from distinct views. Orthogonality means that the dot product of two vectors is zero, which can be guaranteed by minimizing the following function:

$$\mathbf{h}_i^{(v)^T} \cdot \mathbf{h}_i^{(w)} = \sum_{j=1}^{d} h_{ij}^{(v)} \cdot h_{ij}^{(w)}, \tag{4}$$

where $h_{ij}^{(v)}$ is the j-th column element of $\mathbf{h}_i^{(v)}$. Given two distinct views, the following item is applied to ensuring the orthogonality of the representation vector:

$$tr\left(\mathbf{H}^{(v)^T} \mathbf{H}^{(w)}\right) = \sum_{i=1}^{n} \mathbf{h}_i^{(v)^T} \cdot \mathbf{h}_i^{(w)}, \tag{5}$$

For a dataset with more than two views, the diversity term is expressed as follows:

$$\mathscr{L}_d = \sum_{v \neq w} tr\left(\mathbf{H}^{(v)^T} \mathbf{H}^{(w)}\right) \tag{6}$$

Retaining geometric information in each view as much as possible is based on the manifold assumption that if two data points x_i and x_j are close in the original feature space, the representations of these two data points should be close to each other. This can be achieved by minimizing the following function:

$$\mathscr{L}_g = \frac{1}{2} \sum_{i,j}^{n} \left(a_{ij}^{(v)} \left\| \mathbf{h}_i^{(v)} - \mathbf{h}_j^{(v)} \right\|_F^2 \right) = tr\left(\mathbf{H}^{(v)} \mathbf{L}^{(v)} \mathbf{H}^{(v)^T}\right). \tag{7}$$

For each view, a normalized laplacian matrix is constructed by the graph in k-nearest neighbors for the v-th view, denoted as $\mathbf{L}^{(v)} \in \mathbb{R}^{n \times n}$. The weight matrix $\mathbf{A}^{(v)}$ can be constructed by a Gaussian kernel. $\mathbf{L}^{(v)}$ is defined as

$\mathbf{L}^{(v)} = \mathbf{D}^{(v)^{-\frac{1}{2}}}\left(\mathbf{D}^{(v)} - \mathbf{A}^{(v)}\right)\mathbf{D}^{(v)^{-\frac{1}{2}}}$, where $d_{ii}^{(v)} = \sum_j a_{ij}^{(v)}$ is the diagonal element of $\mathbf{D}^{(v)}$.

To avoid over-fitting of multi-view data, a smoothness regularizer $\|\mathbf{H}^{(v)}\|_F^2$ is combined with the objective function. By incorporating reconstruction error $\mathscr{L}_r$, graph regularizer $\mathscr{L}_g$, diversity loss $\mathscr{L}_d$ and smoothness regularizer $\mathscr{L}_s$, we obtain the following objective function:

$$\begin{aligned}\min_{\mathbf{Z}_i^{(v)},\mathbf{H}_i^{(v)},\mathbf{H}^{(v)}} \quad & \sum_{v=1}^{V}\Bigg(\underbrace{\left\|\mathbf{X}^{(v)} - \mathbf{Z}_1^{(v)}\mathbf{Z}_2^{(v)}\cdots\mathbf{Z}_m^{(v)}\mathbf{H}^{(v)}\right\|_F^2}_{\text{Reconstruction}(\mathscr{L}_r)} + \lambda_1\underbrace{tr\left(\mathbf{H}^{(v)}\mathbf{L}^{(v)}\mathbf{H}^{(v)^T}\right)}_{\text{GraphRegularizer}\left(\mathscr{L}_g\right)} \\ & +\lambda_2\underbrace{\left\|\mathbf{H}^{(v)}\right\|_F^2}_{\text{Smoothness}(\mathscr{L}_s)}\Bigg) + \lambda_3\underbrace{\sum_{v\neq w} tr\left(\mathbf{H}^{(v)^T}\mathbf{H}^{(w)}\right)}_{\text{Diversity}(\mathscr{L}_d)}, s.t. \mathbf{H}_i^{(v)} \geq 0, \mathbf{H}^{(v)} \geq 0,\end{aligned} \tag{8}$$

where λ_1, λ_2 and λ_3 are trade-off parameters to control weights of four loss terms.

After solving the objective function (8), the optimized representation matrix in each view $\mathbf{H}^{(v)}$ is obtained. The average value of the representation matrices $\mathbf{H}^{(v)}$ learned from all views is regarded as the global representation matrix $\mathbf{H}^*$:

$$\mathbf{H}^* = \frac{\sum_{v=1}^{V}\mathbf{H}^{(v)}}{V}. \tag{9}$$

Typical clustering method K-means is performed on $\mathbf{H}^*$ to achieve the final clustering results. Due to complex constraints of the objective function, we employ an auto-encoder based method for its optimization.

3.3. Optimization method

There is some consistency between deep matrix factorization and deep auto-encoders. A deep auto-encoder inputs original data, learns a data representations with a multi-layer encoder, as well as recovers input data in the output layer of a multi-layer decoder. They all reconstruct the original data through a multi-layer structure. The output of each layer constructs the output of the next layer with a weight matrix, while constraining some properties of the output, such as nonnegativity. Deep matrix factorization obtains the hidden features at the last decomposition, and a deep auto-encoder yields the output of the encoder as the hidden features. For the V-view case, we employed V deep auto-encoders to optimize the objective function (8). We input an identity matrix $\mathbf{I} \in \mathbb{R}^{n\times n}$ into the v-th one-layer encoder whose dimension is set to d. Therefore, the output of the v-th encoder $\mathbf{C}^{(v)} = \sigma\left(\mathbf{E}^{(v)}\right)$ is utilized as the data representation in the v-th view corresponding to $\mathbf{H}^{(v)}$. And data representations of the i-th layer of the v-th view decoder $\mathbf{C}_i^{(v)}$ corresponding to $\mathbf{H}_i^{(v)}$ are defined as:

$$\mathbf{C}_i^{(v)} = \sigma\left(\mathbf{C}_{i+1}^{(v)}\mathbf{W}_{i+1}^{(v)}\right), \tag{10}$$

where $\sigma(\cdot)$ is utilized to ensure that the output of each layer satisfies nonnegative constraints in deep matrix factorization. Here $\mathbf{C}^{(v)}$ is regarded as the last layer output of the deep matrix factorization, and the reconstruction error is evaluated on $\mathbf{C}_1^{(v)}\mathbf{W}_1^{(v)}$.

In this way, the relation between the objective function (8) and deep auto-encoders is established. During the optimization, we exploit the output of the final layer of the decoder to approximate the original $\mathbf{X}^{(v)}$. Therefore, the loss function of the objective function (8) can be rewritten as the following form:

$$\begin{aligned}\mathscr{J}\left(\mathbf{E}^{(v)}, \mathbf{W}_i^{(v)}\right) = \quad & \sum_{v=1}^{V}\Big(\mathscr{L}_r\left(\mathbf{X}^{(v)}; \mathbf{C}_1^{(v)}\mathbf{W}_1\right) + \lambda_1 tr\left(\mathbf{C}^{(v)}\mathbf{L}^{(v)}\mathbf{C}^{(v)^T}\right) \\ & +\lambda_2\left\|\mathbf{C}^{(v)}\right\|_F^2\Big) + \lambda_3\sum_{v\neq w} tr\left(\mathbf{C}^{(v)^T}\mathbf{C}^{(w)}\right).\end{aligned} \tag{11}$$

Then, we leverage the gradient descent method in deep matrix factorization to solve the objective function. The weight matrix of the encoder $\mathbf{E}^{(v)}$ is updated by the following rule:

$$\mathbf{E}^{(v)(t)} = \mathbf{E}^{(v)(t-1)} - \alpha\nabla_{\mathbf{E}^{(v)(t-1)}}\mathscr{J}\left(\mathbf{E}^{(v)}, \mathbf{W}_i^{(v)}\right), \tag{12}$$

where $\mathbf{E}^{(v)(t)}$ denotes the t-th iteration of $\mathbf{E}^{(v)}$, α denotes the learning rate and $\nabla_{\mathbf{E}^{(v)}}\mathscr{J}\left(\mathbf{E}^{(v)}, \mathbf{W}_i^{(v)}\right)$ is the gradient of $\mathbf{E}^{(v)}$, calculated by:

$$\begin{aligned}\nabla_{\mathbf{E}^{(v)}}\mathscr{J}\left(\mathbf{E}^{(v)},\mathbf{W}_i^{(v)}\right) &= \frac{\partial \mathscr{J}\left(\mathbf{E}^{(v)},\mathbf{W}_i^{(v)}\right)}{\partial \mathbf{E}^{(v)}} = \frac{\partial \mathscr{J}}{\partial \mathbf{C}^{(v)}}\frac{\partial \mathbf{C}^{(v)}}{\partial \mathbf{E}^{(v)}} \\ &= \frac{\partial \mathbf{C}^{(v)}}{\partial \mathbf{E}^{(v)}}\left(\frac{\partial \mathscr{L}_r}{\partial \mathbf{C}^{(v)}} + 2\lambda_1 \mathbf{L}^{(v)}\mathbf{C}^{(v)} + 2\lambda_2 \mathbf{C}^{(v)} + \lambda_3 \sum_{v\neq w}\mathbf{C}^{(w)}\right).\end{aligned} \tag{13}$$

The weight of each layer $\mathbf{W}_i^{(v)}$ in the decoder is iteratively updated by:

$$\begin{aligned}\mathbf{W}_i^{(v)(t)} &= \mathbf{W}_i^{(v)(t-1)} - \alpha\nabla_{\mathbf{W}_i^{(v)(t-1)}}\mathscr{J}\left(\mathbf{E}^{(v)}\mathbf{W}_i^{(v)}\right) \\ &= \mathbf{W}_i^{(v)(t-1)} - \alpha\frac{\partial \mathscr{L}_r}{\partial \mathbf{W}_i^{(v)}}\end{aligned} \tag{14}$$

After training process, data representation matrix of each view $\mathbf{H}^{(v)}$ is restored by the output of the encoder $\mathbf{C}^{(v)}$. The whole process of DDMF is summarized in Algorithm 1.

Algorithm 1 Diversity embedding deep matrix factorization for multi-view clustering (DDMF)

Input: Multi-view data $\mathscr{X}$, trade-off parameters λ_1, λ_2 and λ_3, number of clusters k, dimension of latent data representations d.
Output: Cluster labels of all samples.
1: Initialize the weight of each layer in the decoder $\mathbf{W}^{(v)}$ and the encoder $\mathbf{E}^{(v)}$ of each view.
2: **repeat**
3: Input an identity matrix into the encoder of each view to obtain latent and reconstructed features, then calculate the loss by Equation (11).
4: Update the weights of encoders $\left\{\mathbf{E}^{(v)}\right\}_{v=1}^{V}$ by Equation (12).
5: Compute the weight of each layer in decoder $\left\{\mathbf{W}_i^{(v)}\right\}_{v=1}^{V}$ by Equation (14).
6: **until** Convergence
7: Input an identity matrix into the encoders of all views to obtain a data representation matrix of each view $\mathbf{H}^{(v)}$.
8: Calculate the global data representation $\mathbf{H}^{*}$ by the mean of data representations of all views by Equation (9).
9: Perform K-means on $\mathbf{H}^{*}$.
10: **return** Cluster labels of all samples.

4. Experiments and analysis

In this section, we conduct experiments on several real-world multi-view data-sets to verify the effectiveness of the proposed method by comparing it with several typical and state-of-art multi-view clustering methods.

4.1. Datasets

Experiments are carried out on six publicly available datasets to verify the effectiveness. Detailed descriptions of these datasets are as follows:

ALOI[1] is a color image collection of 1,000 small objects. Four kinds of features are extracted from each image, including RGB color histograms, HSV color histograms, color similarities and Haralick texture features.

Caltech101[2] consists of 9,144 images of 102 categories belonging to 101 object categories and a background clutter category. Each image is represented by 6 kinds of features: Gabor, wavelet moments, centrist, HoG, GIST, and LBP.

Notting-Hill[3] dataset is derived from the movie "Notting-Hill". The dataset consists of 4,660 face images of 5 major casts in 76 tracks.

ORL[4] consists of 400 images from 40 individuals. Ten images for each individual vary in lighting, facial expressions and facial details.

YaleB[5] contains 38 facial image individuals, and each individual has 64 images. For the first 10 objects, 640 images are chosen for the experiment.

[1] http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView.
[2] http://www.vision.caltech.edu/ImageDatasets/Caltech101.
[3] https://rdrr.io/github/schochastics/networkdata/.
[4] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html.
[5] http://vision.ucsd.edu/content/yale-face-database.

Youtube[6] is a dataset of video games of 2,000 samples. Each sample is described in both audio features (mfcc, volume stream, and spectrogram stream) and visual features (cuboids histogram, hist motion estimate, and hog features).

The statistics and brief descriptions of six experimental datasets are summarized in Table 2.

4.2. Compared algorithms

To fairly validate the effectiveness of the proposed method, a typical clustering method **K-means** and the following five state-of-the-art methods are chosen for comparison.

MLAN modifies the similarity graph of all view data in each iteration until achieving the final optimal graph, and learns each view weight coefficient to perform clustering and classification tasks [46].

SwMC automatically assigns appropriate weights to various views and explores the Laplacian rank constraint graph, so that an additional clustering method is not required [47].

MSC-IAS intends to construct an intactness-aware similarity matrix under the assumption that the similarity should have maximum dependence with the corresponding points in the intact space [48].

MCGC imposes a rank constraint on the Laplacian matrix to learn a consensus graph with k connected components and minimizes the divergence between different views [49].

CGD captures the underlying manifold geometry structure of original data and learns an optimized unified graph by cross-view graph diffusion for multi-view data clustering [50].

4.3. Results and analysis

All methods used for the comparison were run on each dataset with default parameters. The balancing parameters λ_1, λ_2 and λ_3 range in $\left\{10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}\right\}$. The dimension of data representation is set to 300 in each view. And each decoder consists of two fully connected layers activated by the sigmoid function with the neuron numbers of (100, 100). Due to the initialization sensitivities of most methods, each method is performed ten times on each dataset, and the average values and standard deviations of the three evaluation metrics are reported. The results evaluated by ACC, NMI and ARI are presented in Tables 3–5.

Each multi-view clustering method outperforms K-means, which demonstrates that clustering performance can be improved by exploiting the complementarity among heterogeneous features. The experimental result indicates that the proposed method achieves the best performance among compared state-of-art methods on all test datasets. Especially, the performance on ALOI and Notting-Hill dataset is improved with a large margin, while the performance on Caltech101 is also comparable. This can be interpreted that various features in ALOI and Notting-Hill have low similarities, therefore representations can be extracted by exploring the advantage of diversity loss. Clustering in the Notting-Hill dataset serves as a more challenging task due to plenty of changes in lighting conditions and the angle of the casts. Therefore, the strength is more obvious in the dataset where the information in each view is more complex and the similarity among different views is lower. Furthermore, it helps to explain why the performance on Caltech101 and YouTube is less impressive. These indicate that through deep matrix factorization, and by enhancing the diversity of data representations, the global data representation learned from multiple views exhibits more discriminative power. To more intuitively demonstrate the advantages of our method over other methods, we report the visualization results on ALOI in Fig. 2.

4.4. Convergence analysis

DDMF leverages an auto-encoder based method to minimize the objective function (8). In order to validate the convergence of DDMF, Fig. 3 illustrates the variations of objective function values with the number of iterations on six datasets. To illustrate the convergence curves more clearly, we change the objective function values to the scale of $\log_{10}$. We set a low learning rate to prevent over-fitting on a specific view, the objective function values decrease stably as the number of iterations increases. And the optimization methods takes around 5,000 iterations for the objective function to converge.

4.5. Parameter sensitivity analysis

For the sake of verifying the influence of various parameter settings on model performance, we conduct parameter sensitivity analysis in this subsection. In DDMF, there are two essential parameters λ_1 and λ_3 to balance the weights between the graph regularizer and the diversity term. We employ a grid search strategy to find the best choices for λ_1 and λ_3 in each dataset, and both of them range in $\left\{1, 10, \cdots, 10^{5}\right\}$. We fix the weight of the smoothing term λ_2 as 0.1. The performance is measured by clustering accuracy, and all results are illustrated in Fig. 4. When the values of λ_1 and λ_3 are relatively large, the clustering performance is comparatively superior. This can be explained by that small λ_1 and λ_3 fail to tolerate reconstruction error. When λ_1 than λ_3 are relatively close, DDMF can achieve favorable clustering performance. DDMF comes with the best

[6] https://archive.ics.uci.edu/ml/datasets/YouTube+Multiview+Video+Games+Dataset.

Table 2
A description of experimental datasets.

Dataset	# Views	# Samples	# Classes	# Dimensions of each view
ALOI	6	1,079	10	64/64/77/13
Caltech101	6	9,144	102	48/40/254/1,984/512/928
Notting-Hill	3	4,660	40	6,750/2,000/3,304
ORL	3	400	40	4,096/3,304/6,750
YaleB	3	650	10	2,500/3,304/6,750
Youtube	6	2,000	10	2,000/1,024/64/512/64/647

Table 3
Accuracy (%) for compared algorithms on multiple datasets. The best results are highlighted in bold (The higher the better).

Dataset	ALOI	Caltech101	Notting-Hill	ORL	YaleB	Youtube
K-means	47.49 (3.31)	13.37 (0.45)	63.71 (6.60)	59.03 (2.41)	17.98 (1.32)	24.21 (1.62)
MLAN	58.94 (5.18)	19.47 (0.64)	59.21 (0.00)	77.75 (0.00)	34.31 (0.00)	16.33 (1.01)
SwMC	47.73 (0.00)	14.11 (0.00)	33.88 (0.00)	74.75 (0.00)	37.08 (0.00)	19.05 (0.00)
MSC-IAS	59.39 (4.25)	18.36 (0.28)	65.45 (9.77)	73.32 (2.16)	53.18 (2.65)	28.45 (0.77)
MGCG	55.51 (0.00)	23.63 (0.00)	52.34 (0.00)	81.00 (0.00)	29.54 (0.00)	30.00 (0.00)
CGD	70.99 (0.00)	10.65 (0.00)	31.24 (0.00)	78.75 (0.00)	32.92 (0.00)	21.30 (0.00)
DDMF	**96.08 (0.76)**	**24.24 (0.49)**	**93.56 (2.07)**	**84.83 (1.64)**	**62.03 (1.04)**	**33.03 (1.08)**

Table 4
Normalized mutual information (%) for compared algorithms on multiple datasets. The best results are highlighted in bold (The higher the better).

Dataset	ALOI	Caltech101	Notting-Hill	ORL	YaleB	Youtube
K-means	47.34 (2.14)	30.30 (0.17)	57.95 (4.85)	77.87 (1.35)	09.14 (2.37)	15.08 (0.58)
MLAN	59.37 (4.31)	30.36 (2.46)	55.50 (0.00)	88.46 (0.00)	31.60 (0.00)	06.14 (1.12)
SwMC	48.66 (0.00)	25.87 (1.56)	08.77 (0.00)	89.04 (0.00)	36.06 (0.00)	11.07 (0.00)
MSC-IAS	70.05 (1.79)	38.11 (0.22)	51.15 (9.12)	86.75 (1.39)	52.25 (2.36)	15.73 (0.48)
MCGC	55.41 (0.00)	26.27 (0.00)	52.34 (0.00)	90.30 (0.00)	26.69 (0.00)	17.41 (0.00)
CGD	68.72 (0.00)	27.06 (0.00)	05.23 (0.00)	89.17 (0.00)	30.86 (0.00)	11.04 (0.00)
DDMF	**96.31 (1.13)**	**46.41 (0.25)**	**87.57 (5.02)**	**92.98 (0.62)**	**68.80 (1.27)**	**22.30 (1.63)**

Table 5
Adjusted rand index (%) for compared algorithms on multiple datasets. The best results are highlighted in bold (The higher the better).

Dataset	ALOI	Caltech101	Notting-Hill	ORL	YaleB	Youtube
K-means	32.98 (2.89)	07.99 (0.35)	49.68 (6.85)	46.34 (2.81)	03.00 (1.07)	07.91 (0.94)
MLAN	34.54 (5.55)	00.39 (0.12)	51.05 (0.00)	66.86 (0.00)	08.99 (0.00)	01.98 (0.63)
SwMC	20.41 (0.00)	00.56 (0.00)	02.88 (0.00)	56.42 (0.00)	12.56 (0.00)	03.61 (0.00)
MSC-IAS	53.24 (3.49)	10.60 (0.25)	49.31 (9.38)	62.66 (3.28)	33.04 (2.15)	09.50 (0.32)
MCGC	35.42 (0.00)	00.38 (0.00)	41.25 (0.00)	69.96 (0.00)	12.25 (0.00)	08.80 (0.00)
CGD	44.32 (0.00)	03.78 (0.00)	02.21 (0.00)	66.63 (0.00)	12.88 (0.00)	06.35 (0.00)
DDMF	**91.25 (1.84)**	**17.01 (0.90)**	**87.08 (5.82)**	**78.09 (1.60)**	**53.63 (1.64)**	**14.09 (1.56)**

performance on most datasets. In general, the balancing parameters λ_1 and λ_3 within certain ranges can promote DDMF to obtain superior clustering performance.

4.6. Ablation studies

The ablation studies of graph regularizer and diversity loss on four datasets are presented in Table 6. The clustering performance is unacceptable while neither of both constraints exists. Reconstruction term $\mathscr{L}_r$ is the basement of our model to guarantee the main performance. Both diversity term $\mathscr{L}_d$ to enhance the diversity and graph regularizer term $\mathscr{L}_g$ to mine geometric information of each view can improve the experimental results. The addition of smoothness term $\mathscr{L}_s$ to prevent overfitting has no clear effect on ALOI, but significant optimizations on Notting-Hill and ORL. Any one of the three constraints can optimize the clustering performance. When all three constraints are combined, the clustering performance is improved more significantly than that achieved by reconstruction term $\mathscr{L}_r$ or adding any one of the other three terms ($\mathscr{L}_g, \mathscr{L}_s$ and $\mathscr{L}_d$). In conclusion, the experimental results verify that data representation preserving geometric information with diversity embedding can largely improve the clustering performance of multi-view deep matrix factorization.

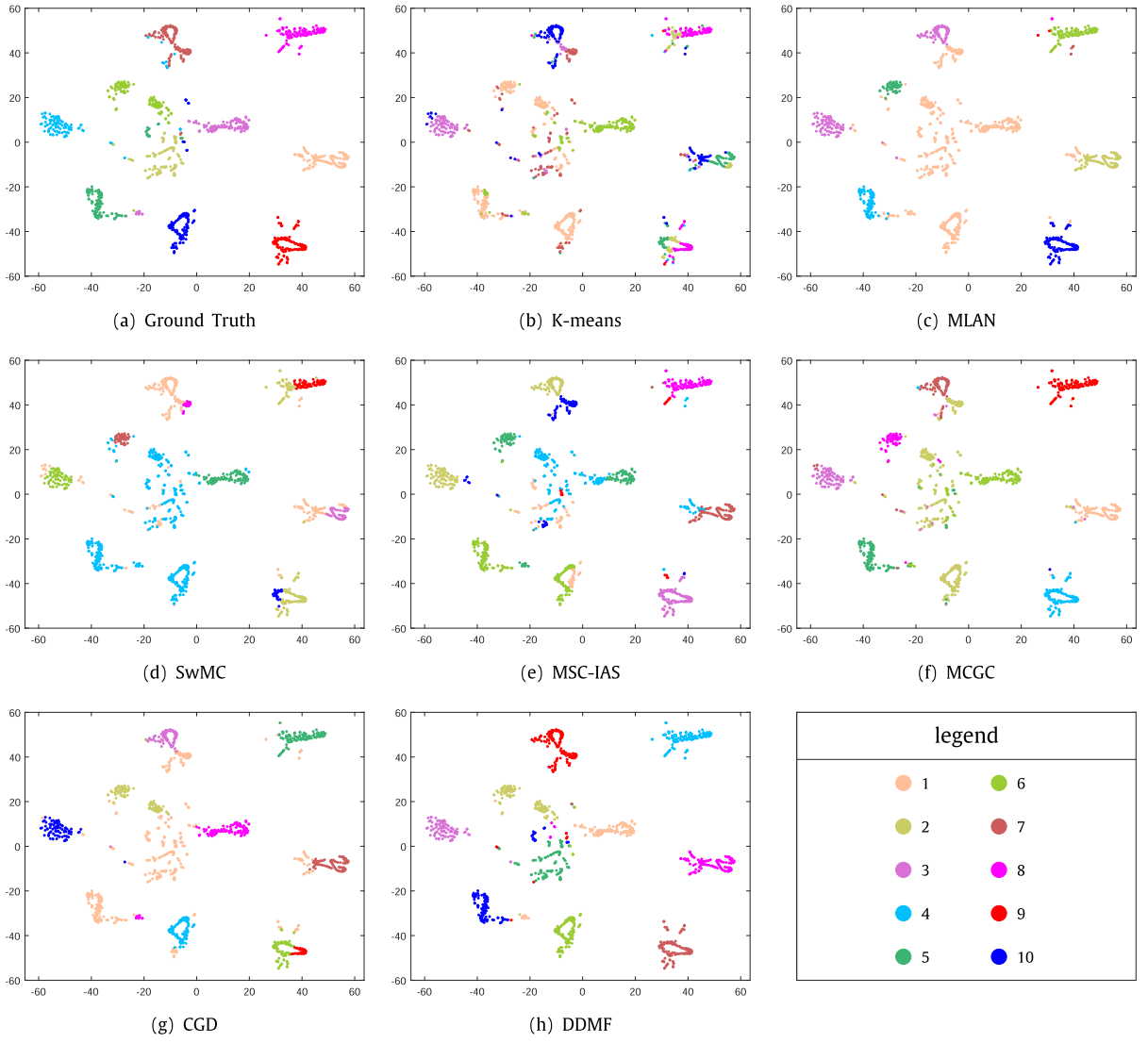

Fig. 2. Visualizations of compared multi-view clustering methods and ground-truth on ALOI.

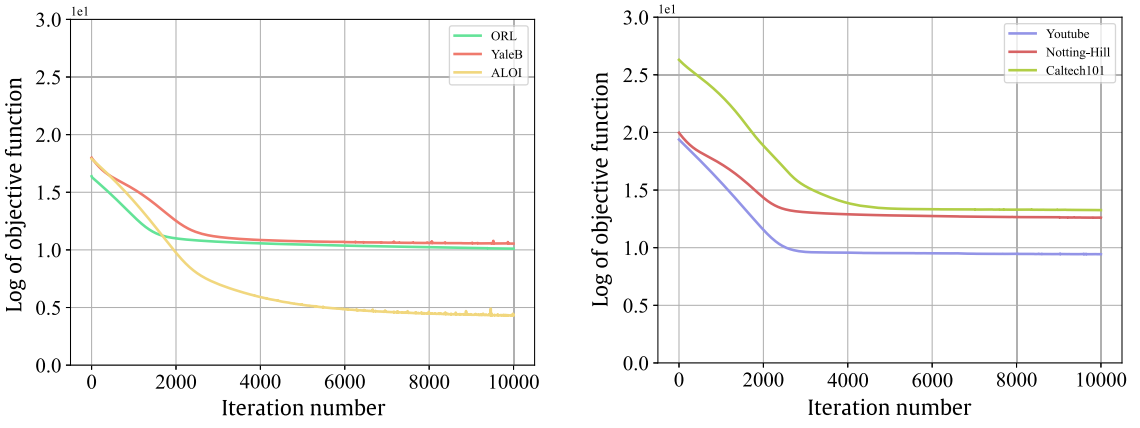

Fig. 3. Convergence curve of the objective function, where the proposed method converges around 5,000 iterations.

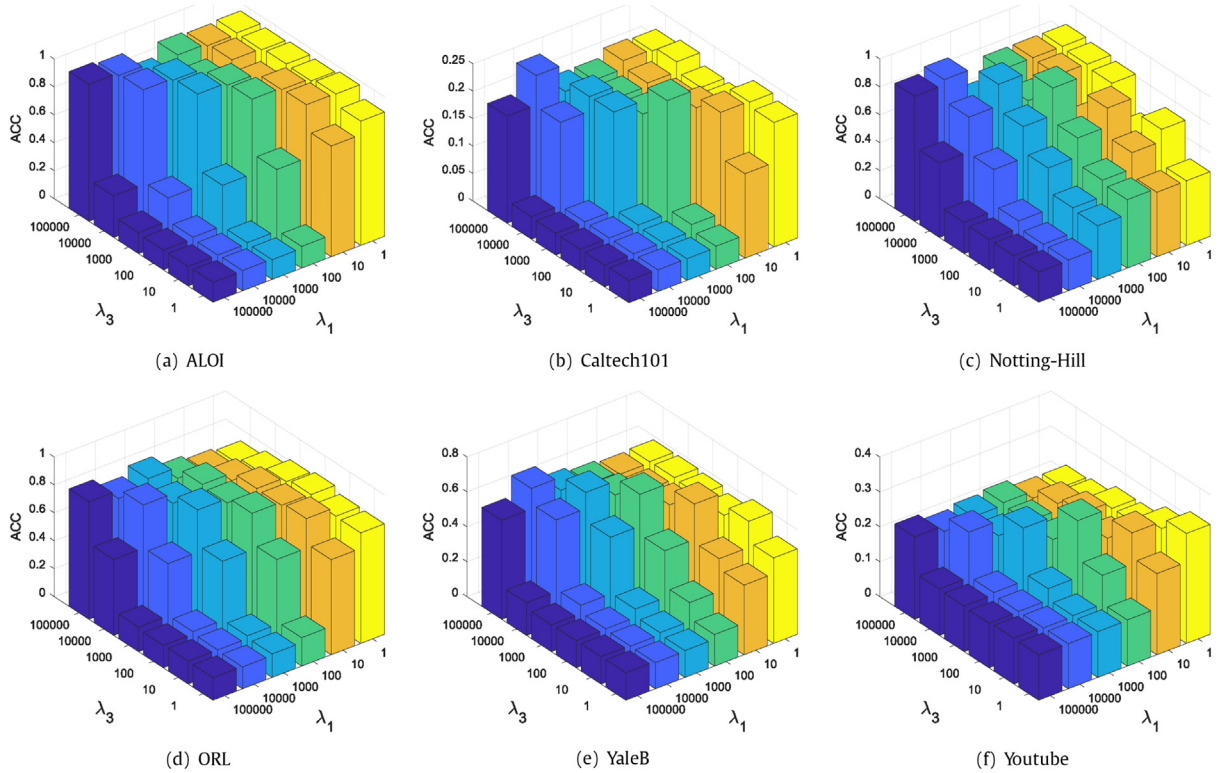

Fig. 4. The clustering accuracy of DDMF on six datasets with λ_1 and λ_3 chosen with grid search strategy, varying in $\left\{1, 10, \cdots, 10^5\right\}$.

Table 6
Ablation studies of the proposed method on three datasets evaluated by ACC (%), NMI (%) and ARI (%). The values in brackets are standard deviations.

Dataset	Methods	ACC	NMI	ARI
ALOI	$\mathscr{L}_r$	89.81 (1.02)	83.47 (1.94)	76.63 (2.22)
	$\mathscr{L}_r + \mathscr{L}_g$	92.81 (0.61)	87.99 (0.90)	83.53 (1.38)
	$\mathscr{L}_r + \mathscr{L}_d$	90.31 (1.27)	83.73 (1.96)	77.91 (2.96)
	$\mathscr{L}_r + \mathscr{L}_s$	89.58 (0.88)	82.87 (1.43)	76.30 (1.60)
	$\mathscr{L}_r + \mathscr{L}_g + \mathscr{L}_d$	**97.85 (2.15)**	**96.22 (3.62)**	**95.43 (4.59)**
Notting-Hill	$\mathscr{L}_r$	62.28 (4.10)	47.05 (0.40)	38.52 (7.69)
	$\mathscr{L}_r + \mathscr{L}_g$	75.17 (6.77)	60.56 (8.31)	61.93 (9.69)
	$\mathscr{L}_r + \mathscr{L}_d$	65.76 (1.25)	45.49 (0.99)	41.26 (1.45)
	$\mathscr{L}_r + \mathscr{L}_s$	68.98 (9.76)	51.40 (5.14)	49.12 (8.82)
	$\mathscr{L}_r + \mathscr{L}_g + \mathscr{L}_d$	**94.16 (1.06)**	**89.85 (2.01)**	**89.20 (1.97)**
ORL	$\mathscr{L}_r$	57.85 (2.21)	76.36 (2.17)	41.63 (3.90)
	$\mathscr{L}_r + \mathscr{L}_g$	64.40 (2.78)	80.94 (1.69)	50.75 (2.39)
	$\mathscr{L}_r + \mathscr{L}_d$	71.13 (2.12)	86.30 (1.13)	60.69 (2.31)
	$\mathscr{L}_r + \mathscr{L}_s$	62.20 (1.86)	78.91 (1.26)	45.46 (2.50)
	$\mathscr{L}_r + \mathscr{L}_g + \mathscr{L}_d$	**86.21 (1.97)**	**93.17 (0.77)**	**79.77 (2.20)**

5. Conclusion

In this paper, we proposed a deep matrix factorization model with diversity embedding named DDMF for multi-view clustering tasks. Deep matrix factorization learns more effective latent features from complex information while diversity loss is adopted to reduce the redundancy among the data representations of various views. DDMF incorporated manifold information with the graph regularizer. To deal with the issue, we constructed the model with neural networks to optimize the objective function. By comparing with five state-of-the-art methods, comprehensive experiments on six datasets were conducted to validate the effectiveness of DDMF. Experiment results showed that DDMF boosted more significant advan-

tages when there exists low similarity among various view features. Therefore, in the future work, we will take the consensus information into consideration, which is also a vital factor for multi-view clustering.

CRediT authorship contribution statement

Zexi Chen: Conceptualization, Methodology, Writing – review & editing. **Pengfei Lin:** Data curation, Software, Visualization. **Zhaoliang Chen:** Software, Visualization, Investigation. **Dongyi Ye:** Supervision. **Shiping Wang:** Validation, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China (Grant No. U21A20472), and the Natural Science Foundation of Fujian Province (Grant No. 2020J01130193).

References

[1] Bin Shen, Luo Si, Non-negative matrix factorization clustering on multiple manifolds, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010, pp. 575–580.
[2] Deng Cai, Xiaofei He, Jiawei Han, Thomas S Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2010) 1548–1560.
[3] Zhaoliang Chen, Wei Zhao, Shiping Wang, Kernel meets recommender systems: A multi-kernel interpolation for matrix completion, Expert Syst. Appl. 168 (2021) 114436.
[4] Bin Gao, P.-A. Absil, A riemannian rank-adaptive method for low-rank matrix completion, Comput. Optim. Appl. 81 (1) (2022) 67–90.
[5] Defu Lian, Xing Xie, Enhong Chen, Discrete matrix factorization and extension for fast item recommendation, IEEE Trans. Knowl. Data Eng. 33 (2021) 1919–1933.
[6] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, Chun Chen, Fast adaptively weighted matrix factorization for recommendation with implicit feedback, in: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 3470–3477.
[7] Di Wang, Quan Wang, Yaqiang An, Xinbo Gao, Yumin Tian, Online collective matrix factorization hashing for large-scale cross-media retrieval, in: Proceedings of the Forty-Third International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1409–1418.
[8] Fanghua Ye, Chuan Chen, Zibin Zheng, Deep autoencoder-like nonnegative matrix factorization for community detection, in: Proceedings of the Twenty-Seventh ACM International Conference on Information and Knowledge Management, 2018, pp. 1393–1402.
[9] Tiantian He, Yang Liu, Tobey H. Ko, Keith C.C. Chan, Yew-Soon Ong, Contextual correlation preserving multiview featured graph clustering, IEEE Trans. Cybern. 50 (10) (2019) 4318–4331.
[10] Siyuan Peng, Wee Ser, Badong Chen, Zhiping Lin, Robust semi-supervised nonnegative matrix factorization for image clustering, Pattern Recogn. 111 (2021) 107683.
[11] Chris HQ Ding, Tao Li, Michael I Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2008) 45–55.
[12] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, Björn W. Schuller, A deep semi-nmf model for learning hidden representations, in: Proceedings of the Thirty-First International Conference on Machine Learning, 2014, pp. 1692–1700.
[13] Guosheng Cui, Ye Li, Nonredundancy regularization based nonnegative matrix factorization with manifold learning for multiview data representation, Inf. Fusion 82 (2022) 86–98.
[14] Shiping Wang, Zhewen Wang, Wenzhong Guo, Accelerated manifold embedding for multi-view semi-supervised classification, Inf. Sci. 562 (2021) 438–451.
[15] Shu Li, Wen-Tao Li, Wei Wang, Co-gcn for multi-view semi-supervised learning, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2020, pp. 4691–4698.
[16] Jingjing Tang, Yiwei He, Yingjie Tian, Dalian Liu, Gang Kou, Fawaz E. Alsaadi, Coupling loss and self-used privileged information guided multi-view transfer learning, Inf. Sci. 551 (2021) 245–269.
[17] Pei Yang, Wei Gao, Multi-view discriminant transfer learning, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013, pp. 1848–1854.
[18] Saroj S. Shivagunde, Ashwani Nadapana, V. Vijaya Saradhi, Multi-view incremental discriminant analysis, Inf. Fusion 68 (2021) 149–160.
[19] Xinge You, Jiamiao Xu, Wei Yuan, Xiao-Yuan Jing, Dacheng Tao, Taiping Zhang, Multi-view common component discriminant analysis for cross-view classification, Pattern Recogn. 92 (2019) 37–51.
[20] Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, Xu. Zenglin, Deep embedded multi-view clustering with collaborative training, Inf. Sci. 573 (2021) 279–290.
[21] Shiping Wang, Shunxin Xiao, William Zhu, Yingya Guo, Multi-view fuzzy clustering of deep random walk and sparse low-rank embedding, Inf. Sci. 586 (2022) 224–238.
[22] Shiping Wang, Zhaoliang Chen, Shide Du, Zhouchen Lin, Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification, IEEE Trans. Pattern Anal. Mach. Intell. (2021), https://doi.org/10.1109/TPAMI.2021.3082632.
[23] Xuanlong Ma, Xueming Yan, Jingfa Liu, Guo Zhong, Simultaneous multi-graph learning and clustering for multiview data, Inf. Sci. 593 (2022) 472–487.
[24] Lihua Zhou, Du. Guowang, Kevin Lü, Lizhen Wang, A network-based sparse and multi-manifold regularized multiple non-negative matrix factorization for multi-view clustering, Expert Syst. Appl. 174 (2021) 114783.
[25] Linlin Zong, Xianchao Zhang, Long Zhao, Hong Yu, Qianli Zhao, Multi-view clustering via multi-manifold regularized non-negative matrix factorization, Neural Networks 88 (2017) 74–89.
[26] Shudong Huang, Zhao Kang, Xu. Zenglin, Quanhui Liu, Robust deep k-means: An effective and simple method for data clustering, Pattern Recogn. 117 (2021) 107996.
[27] Handong Zhao, Zhengming Ding, Fu. Yun, Multi-view clustering via deep matrix factorization, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2921–2927.
[28] Shaowei Wei, Jun Wang, Guoxian Yu, Carlotta Domeniconi, Xiangliang Zhang, Multi-view multiple clusterings using deep matrix factorization, in: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 6348–6355.

[29] Chen Zhang, Siwei Wang, Jiyuan Liu, Sihang Zhou, Pei Zhang, Xinwang Liu, En Zhu, Changwang Zhang, Multi-view clustering via deep matrix factorization and partition alignment, in: Proceedings of the Twenty-Ninth ACM International Conference on Multimedia, 2021, pp. 4156–4164.
[30] Jing Wang, Feng Tian, Hongchuan Yu, Chang Hong Liu, Kun Zhan, Xiao Wang, Diverse non-negative matrix factorization for multiview data representation, IEEE Trans. Cybern. 48 (2017) 2620–2632.
[31] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, Roberto Prevete, A survey on modern trainable activation functions, Neural Networks 138 (2021) 14–32.
[32] Xuelong Li, Rui Zhang, Qi Wang, Hongyuan Zhang, Autoencoder constrained clustering with adaptive neighbors, IEEE Trans. Neural Networks Learn. Syst. 32 (2020) 443–449.
[33] Xifeng Guo, Long Gao, Xinwang Liu, Jianping Yin, Improved deep embedded clustering with local structure preservation, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 1753–1759.
[34] Lele Fu, Jinghua Yang, Chuan Chen, Chuanfu Zhang, Low-rank tensor approximation with local structure for multi-view intrinsic subspace clustering, Inf. Sci. 606 (2022) 877–891.
[35] Hao Wang, Yan Yang, Bing Liu, GMC: graph-based multi-view clustering, IEEE Trans. Knowl. Data Eng. 32 (2020) 1116–1129.
[36] Zhao Kang, Guoxin Shi, Shudong Huang, Wenyu Chen, Xiaorong Pu, Joey Tianyi Zhou, Zenglin Xu, Multi-graph fusion for multi-view spectral clustering, Knowl.-Based Syst. 189 (2020) 105102.
[37] Shudong Huang, Zhao Kang, Ivor W. Tsang, Xu. Zenglin, Auto-weighted multi-view clustering via kernelized graph learning, Pattern Recogn. 88 (2019) 174–184.
[38] Qianyao Qiang, Bin Zhang, Fei Wang, Feiping Nie, Fast multi-view discrete clustering with anchor graphs, in: Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, 2021, pp. 9360–9367.
[39] Xiaobo Wang, Xiaojie Guo, Zhen Lei, Changqing Zhang, Z. Li. Stan, Exclusivity-consistency regularized multi-view subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1–9.
[40] Yongyong Chen, Shuqin Wang, Fangying Zheng, Yigang Cen, Graph-regularized least squares regression for multi-view subspace clustering, Knowl.-Based Syst. 194 (2020) 105482.
[41] Juncheng Lv, Zhao Kang, Boyu Wang, Luping Ji, Zenglin Xu, Multi-view subspace clustering via partition fusion, Inf. Sci. 560 (2021) 410–423.
[42] Shiping Wang, Zhaoliang Chen, William Zhu, Fei-Yue Wang, Deep random walk of unitary invariance for large-scale data representation, Inf. Sci. 554 (2021) 1–14.
[43] Jialu Liu, Chi Wang, Jing Gao, Jiawei Han, Multi-view clustering via joint nonnegative matrix factorization, in: Proceedings of the SIAM International Conference on Data Mining, 2013, pp. 252–260.
[44] Shuai Chang, Hu. Jie, Tianrui Li, Hao Wang, Bo Peng, Multi-view clustering via deep concept factorization, Knowl.-Based Syst. 217 (2021) 106807.
[45] Shudong Huang, Zhao Kang, Xu. Zenglin, Auto-weighted multi-view clustering via deep matrix decomposition, Pattern Recogn. 97 (2020) 107015.
[46] Feiping Nie, Guohao Cai, Xuelong Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2408–2414.
[47] Feiping Nie, Jing Li, Xuelong Li, et al, Self-weighted multiview clustering with multiple graphs, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 2564–2570.
[48] Xiaobo Wang, Zhen Lei, Xiaojie Guo, Changqing Zhang, Hailin Shi, Stan Z. Li, Multi-view subspace clustering with intactness-aware similarity, Pattern Recogn. 88 (2019) 50–63.
[49] Kun Zhan, Feiping Nie, Jing Wang, Yi Yang, Multiview consensus graph clustering, IEEE Trans. Image Process. 28 (2019) 1261–1270.
[50] Chang Tang, Xinwang Liu, Xinzhong Zhu, En Zhu, Zhigang Luo, Lizhe Wang, Wen Gao, Cgd: Multi-view clustering via cross-view graph diffusion, in: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 5924–5931.