Chapter 11

New Approaches in Multi-View Clustering

Fanghua Ye, Zitai Chen, Hui Qian, Rui Li,
Chuan Chen and Zibin Zheng

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.75598

Abstract

Many real-world datasets can be naturally described by multiple views. Due to this, multi-view learning has drawn much attention from both academia and industry. Compared to single-view learning, multi-view learning has demonstrated plenty of advantages. Clustering has long been serving as a critical technique in data mining and machine learning. Recently, multi-view clustering has achieved great success in various applications. To provide a comprehensive review of the typical multi-view clustering methods and their corresponding recent developments, this chapter summarizes five kinds of popular clustering methods and their multi-view learning versions, which include *k*-means, spectral clustering, matrix factorization, tensor decomposition, and deep learning. These clustering methods are the most widely employed algorithms for single-view data, and lots of efforts have been devoted to extending them for multi-view clustering. Besides, many other multi-view clustering methods can be unified into the frameworks of these five methods. To promote further research and development of multi-view clustering, some popular and open datasets are summarized in two categories. Furthermore, several open issues that deserve more exploration are pointed out in the end.

Keywords: clustering, multi-view clustering, multi-view *k*-means, multi-view spectral clustering, multi-view matrix factorization, tensor decomposition, deep learning

1. Introduction

Clustering is one of the most critical unsupervised learning techniques, which has been widely applied for data analysis, such as social network analysis, gene expression analysis, heterogeneous data analysis, and market analysis. The goal of clustering is to partition a dataset into several groups such that data samples in the same group are more similar than those in

different groups. Clustering plays an important role in mining the hidden patterns. However, most of the existing clustering algorithms are designed for single-view data.

With the rapid development of Internet and communication technology (ICT), the accesses to extract data are dramatically extended. That is, data can be collected from multiple sources or multiple facets. In such setting, each datum is associated with much richer information, which results in the requirement that to mine the intrinsic and valuable patterns hidden in the data, it is a necessity to take full advantage of the information contained in multiple sources. This issue is formally referred to as *multi-view learning*. To be more specific, each view corresponds to one source of information. For example, web pages can be described by both the page-contents (one view) and the hyperlink information (another view). Besides, different facets of a datum can also be treated as different views. For instance, an image can be characterized by its shape, color, and location.

Obviously, integrating the information contained in multiple views can bring great benefits for data clustering. The most straightforward way to utilize the information of all views is to concatenate the data features of each view together and then perform the traditional clustering methods such as *k*-means. However, such a method lacks the ability to distinguish the different significance of different views. That is, the important views and less important views are treated equally, which may degrade the ultimate performance severely. To take better advantage of the multi-view information, the ideal approach is to simultaneously perform the clustering using each view of data features and integrate their results based on their importance to the clustering task. Formally, this approach is known as *multi-view clustering*.

As an emerging and effective paradigm in data mining and machine learning, multi-view clustering refers to the clustering of the same class of data samples with multi-view representations, either from various information sources or from different feature generators. It is clear that if the clustering method cannot cope appropriately with multi-views, these views may even degrade the performance of multi-view clustering. To make use of multi-view information to improve clustering results, there are two main challenges to overcome. The first one is how to naturally ensemble the multiple clustering results of all the views. The second one is how to learn the importance of different views to the clustering task. In addition, these two issues should be figured out simultaneously. Thus, to achieve these goals, new clustering objective function should be designed, followed by the new solving method.

Multi-view clustering was first studied by Bickel and Scheffer [1] in 2004. They extended the classic *k*-means and expectation maximization (EM) clustering methods to the multi-view environment to deal with text data with two conditionally independent views. Based on this seminal work, a variety of multi-view clustering methods have been proposed over the past two decades [2–4]. Since covering all the proposed methods in one chapter is hard, to provide a comprehensive review of the typical multi-view clustering methods and their corresponding recent developments, we summarize five kinds of popular clustering methods and their multi-view learning versions, which include *k*-means, spectral clustering, matrix factorization, tensor decomposition, and deep learning. This is based on the consideration that these clustering methods are the most widely employed algorithms for single-view data, and lots of efforts have been devoted to extending them for multi-view clustering. Besides, many other multi-

view clustering methods can be unified into the frameworks of these five methods. Therefore, when readers become familiar with these five multi-view clustering methods, they can capture the core ideas of other multi-view clustering methods easily. This chapter is self-contained, which follows a line of introduction from the preliminaries of these clustering methods for single-view data to their variant forms for multi-view clustering.

The remainder of this chapter is organized as follows. Section 2 describes the benefits of multi-view clustering. Section 3 details the aforementioned five multi-view clustering methods. Section 4 summarizes two kinds of popular open datasets. Several open issues are illustrated in Section 5. Section 6 concludes this chapter.

2. Benefits of multi-view clustering

Compared with the clustering methods that are implemented on single-view data, multi-view clustering is expected to obtain more robust and novel partitioning results by exploiting the redundant and complementary information in different views [5], as stated in the following sections.

2.1. Benefit one: accurate description of data

It is obvious that single-view data may contain incomplete knowledge, while multi-view data usually contains complementary and redundant information, which results in a more accurate description of the data. For example, it may fail to identify the intrinsic community structures of a social network via just leveraging the friendships. However, if more information such as users' demographics can be obtained, it is more inclined to find out the implicit relationships between users.

2.2. Benefit two: reducing noises of data

Even when the information contained in single-view data is complete, there may exist some unavoidable noises. It is apparent that data cleaning is one critical issue in data analysis, which can tremendously affect the performance of clustering algorithms. It is quite hard and costly to remove all the noises of data, and thus single-view noisy data usually leads to unsatisfactory clustering results. On the other hand, multi-view clustering is able to circumvent the side effect of noises or corrupted data in each view and emphasize the common patterns shared by multi-view data.

2.3. Benefit three: wider range of applications

There is no doubt that all the multi-view clustering methods can be applied to single-view data. However, many clustering tasks are impossible to implement by single-view clustering due to its limitations. For example, data with multiple modalities is becoming more and more common and heterogeneous information networks are gaining increasing popularity as well.

These types of data naturally fit into multi-view learning, while cannot be settled by single-view learning methods appropriately. In all, the complementary property among multi-view data can overcome the limitations of single-view data and expand their application areas.

3. Multi-view clustering methods

Due to the widespread use of multi-view datasets in practice, many realistic applications are accomplished by multi-view learning methods, such as community detection in social networks, image annotation in computer vision, and cross-domain user modeling in recommendation systems [6]. Meanwhile, based on the seminal work of Bickel and Scheffer [1], plenty of multi-view clustering methods have been proposed [2, 3, 5]. As explained in Section 1, this chapter seeks to review five kinds of typical clustering methods and their multi-view versions, which include *k*-means, spectral clustering, matrix factorization, tensor decomposition, and deep learning. All of these five methods are popular methods for single-view clustering. Although there are some other multi-view clustering methods not contained in this chapter, such as the canonical correlation analysis (CCA)-based multi-view clustering methods [7], the DBSCAN-based multi-view clustering methods [8], and the lower dimensional subspace-based multi-view clustering methods [9], most of them can be unified into the frameworks of these five involved methods. For instance, a pair-wise sparse subspace representation model for multi-view clustering proposed in [10] can be unified into the framework of matrix factorization.

3.1. Multi-view clustering via *k*-means

k-means is one of the most popular clustering algorithms with a history of more than 50 years [11]. Except for its simplicity, *k*-means has a good potential to deal with large-scale datasets. Owing to these properties, *k*-means has been successfully used in various topics, including computer vision, social network analysis, and market segmentation, to name but a few. Although it has been studied deeply over the past few decades, many variants of *k*-means are put forward continuously [12–15].

3.1.1. Preliminaries of k-means

As a classic clustering algorithm, *k*-means employs K prototype vectors (i.e., centers or centroids of the K clusters) to characterize each data sample and minimizes a sum of squared loss function to find these prototypes. Consider a dataset denoted by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}$, where $\mathbf{x}_i \in \mathbb{R}^M$ represents the attribute (feature) vector of the i-th data sample $\mathbf{x}_i$. In order to partition the dataset $\mathbf{X}$ into K disjoint clusters, denoted by $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, *k*-means tries to optimize the following objective function:

$$\varepsilon = \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{ik} \|\mathbf{x}_i - \mathbf{v}_k\|_2^2, \quad \mathbf{v}_k = \frac{\sum_{i=1}^{N} \delta_{ik} \mathbf{x}_i}{\sum_{i=1}^{N} \delta_{ik}} = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}, \tag{1}$$

where δ_{ik} is an indicator variable with $\delta_{ik} = 1$ if $\mathbf{x}_i \in C_k$ and 0 otherwise and $\mathbf{v}_k$ is the k-th prototype vector, i.e., the k-th cluster center.

As can be seen, Eq. (1) adopts the Euclidean distance to measure the similarities between data samples. However, there are many data structures or data distributions in real world. Thus, it is not always suitable to apply this basic form of *k*-means to accurately identify the hidden patterns of datasets. What is more, some datasets may be not separable in the low-dimensional space. Recently, kernel method has been of wide concern in the field of machine learning. By introducing a kernel function, the original nonlinear datasets are mapped to a higher dimensional reproducing kernel Hilbert space. In the new space, the datasets become linearly separable. For this reason, the kernel *k*-means algorithm [16, 17] has been proposed. It is just a generalization of the standard *k*-means algorithm and has the following objective function:

$$\varepsilon = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}\|\phi(\mathbf{x}_i) - \mathbf{v}'_k\|_2^2, \quad \mathbf{v}'_k = \frac{\sum_{i=1}^{N}\delta_{ik}\phi(\mathbf{x}_i)}{\sum_{i=1}^{N}\delta_{ik}}, \tag{2}$$

where $\phi : \mathbf{X} \to \mathbf{H}$ is a nonlinear transformation function. Define a kernel function $\mathcal{K} : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ with $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. Then, Eq. (2) can be rewritten into the kernel form as below:

$$\varepsilon = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}\left(\mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\frac{\sum_{j=1}^{N}\delta_{jk}\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{j=1}^{N}\delta_{jk}} + \frac{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}\mathcal{K}(\mathbf{x}_j, \mathbf{x}_l)}{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}}\right). \tag{3}$$

With the aid of the kernel function, there is no need to explicitly provide the transformation function ϕ. This is because, for certain kernel function, the corresponding transformation function is intractable. However, the inner products in the kernel space can be easily obtained according to the kernel function.

3.1.2. Basic form of multi-view k-means

Both the *k*-means and the kernel *k*-means described above are designed for single-view data. To solve the multi-view clustering problem, some new objective functions should be developed. Assume that there are V views in total. Let $\mathcal{X} = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(V)}\right\}$ denote the data of all the views. It is obvious that different views should have different contributions according to their conveyed information. To achieve this goal, it is straightforward to modify the standard *k*-means to make it applicable in the multi-view environment with a new objective function as follows:

$$\varepsilon = \sum_{v=1}^{V}\mu_v^{\gamma}\varepsilon_v, \text{ s.t.}\mu_v \geq 0, \sum_{v=1}^{V}\mu_v = 1, \gamma > 1, \tag{4}$$

where μ_v is the weight factor for the v-th view, γ is a parameter used to control the weight distribution, and ε_v corresponds to the objective function (i.e., loss function) of the v-th view:

$$\varepsilon_v = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}\|\mathbf{x}_i^{(v)} - \mathbf{v}_k^{(v)}\|_2^2, \mathbf{v}_k^{(v)} = \frac{\sum_{i=1}^{N}\delta_{ik}\mathbf{x}_i^{(v)}}{\sum_{i=1}^{N}\delta_{ik}}. \tag{5}$$

Similarly, the objective function of the multi-view kernel *k*-means can be obtained, which is omitted here. Note that finding the optimal solution of Eq. (4) is an NP-hard problem; thus,

some iterative algorithms are developed according to the greedy strategy. One basic iterative algorithm works in a two-stage manner: (1) updating the clustering for given weights and (2) updating the weights for given clusters; see [18] for details.

Denote $\|\mathbf{X}\|_F$ as the Frobenius norm of a given matrix $\mathbf{X}$, i.e., $\|\mathbf{X}\|_F = \sqrt{\sum_{i,j=1}^{N} x_{ij}^2}$. Then, Eq. (4) can be easily transformed into a matrix form as shown in the following:

$$\min_{\mathbf{V}^{(v)},\,\mathbf{U},\,\mu_v} \sum_{v=1}^{V} \mu_v^{\gamma} \|\mathbf{X}^{(v)} - \mathbf{V}^{(v)}\mathbf{U}^T\|_F^2, \text{ s.t.} u_{ik} \in \{0,1\}, \sum_{k=1}^{K} u_{ik} = 1, \mu_v \geq 0, \sum_{v=1}^{V} \mu_v = 1, \gamma > 1, \tag{6}$$

where $\mathbf{V}^{(v)} \in \mathbb{R}^{M_v \times K}$ denotes the centroid matrix for the v-th view and $\mathbf{U} \in \mathbb{R}^{N \times K}$ denotes the clustering indicator matrix with the (i,k) element being δ_{ik}. Note that all the views share a common clustering indicator matrix $\mathbf{U}$.

3.1.3. Variants of multi-view k-means

The basic formulations of multi-view *k*-means shown in Eqs. (4) and (6) do have some drawbacks. For example, it assumes that all the views are sharing a common clustering indicator matrix $\mathbf{U}$. However, the structure information contained may be very limited or even lost in some views. In such case, the performance will be severely affected if all the views share a common clustering indicator matrix. To tackle the issues, many variants of multi-view *k*-means clustering algorithms have been proposed in recent years. Instead of the ℓ_2-norm, the structured sparsity-inducing norm, i.e., the $\ell_{2,1}$-norm, is adopted to strengthen the basic multi-view *k*-means, in the hope that the effect of outlier data samples will be reduced [19]. In [20], a *k*-means-based dual-regularized multi-view outlier detection method (DMOD) is proposed to identify the cluster outliers and the attribute outliers simultaneously, which is based on a novel cross-view outlier measurement criterion. Moreover, in the DMOD model, each view is associated with a particular clustering indicator matrix, and another alignment matrix is introduced to enforce the consistency between different views. An automated two-level variable weighting clustering algorithm, called TW-*k*-means, is developed in [21]. TW-*k*-means is able to compute weights for each view and each individual attribute simultaneously. More specifically, in this algorithm, to identify the compactness of the view, a view weight is assigned to each view, and an attribute weight is assigned to each attribute in the view to identify the importance of the attribute. Both view weights and attribute weights are employed in the distance function to determine the cluster structures of data samples. Similar strategies have also been taken in [22, 23] to learn more robust multi-view *k*-means models.

As aforementioned, it is NP-hard to find the optimal solution of the multi-view *k*-means clustering problem. The greedy iterative algorithm has a high risk of getting stuck in local optima during the optimization. Recently, the self-paced learning has been used to alleviate this problem. The general self-paced learning model consists of a weighted loss function on all data samples and a regularizer term imposed on the weights of data samples. By gradually increasing the penalty on the regularizer, more data samples are automatically added into consideration from "easy" to "complex" via a pure self-paced approach. In this, Xu et al. [24]

present a new multi-view self-paced learning (MSPL) algorithm for clustering based on multi-view *k*-means. MSPL learns the multi-view model by not only progressing from "easy" to "complex" data samples but also from "easy" to "complex" views. The objective function of MSPL is quite succinct, which is shown in Eq. (7).

$$\min_{\mathbf{V}^{(v)},\mathbf{U},\mathcal{U}} \sum_{v=1}^{V} \|\left(\mathbf{X}^{(v)} - \mathbf{V}^{(v)}\mathbf{U}^{T}\right)\mathrm{diag}\left(\sqrt{\mu^{(v)}}\right)\|_F^2 + f(\mathcal{U}), \text{s.t.} u_{ik} \in \{0,1\}, \sum_{k=1}^{K} u_{ik} = 1, \tag{7}$$

where $\mu^{(v)} = \left[\mu_1^{(v)}, \mu_2^{(v)}, \ldots, \mu_N^{(v)}\right] \in [0,1]^N$ denotes the weights of data samples in the v-th view, $\mathcal{U} = \left[\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(V)}\right]$, and $f(\mathcal{U})$ denotes the regularization term on demand.

3.2. Multi-view clustering via spectral clustering

Spectral clustering is built upon the spectral graph theory. In recent years, spectral clustering has become one of the most popular clustering algorithms and shown its effectiveness in various real-world applications ranging from statistics, computer sciences to bioinformatics. Due to its adaptation in data distribution, spectral clustering often outperforms traditional clustering algorithms such as *k*-means. In addition, spectral clustering is simple to implement and can be solved efficiently by standard linear algebra.

3.2.1. Preliminaries of spectral clustering

Spectral clustering is closely related to the minimum cut problem of graphs. It first performs dimensionality reduction on the original data space by leveraging the spectrum of the similarity matrix of data samples and then performs *k*-means on the low-dimensional space to partition data into different clusters. Therefore, for a set of data samples, a similarity matrix should be constructed at first. Typically, each data sample is treated as a node of a graph and each relationship between data samples is regarded as an edge in the graph. Besides, each edge is associated with a weight. It is obvious that the value of the edge weight between two far-away data samples should be low and the value between two close data samples should be high. For a given dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}$, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{S})$ be the generated undirected weighted graph, where $\mathcal{V}$ denotes the set of nodes representing the data samples and $\mathcal{E}$ denotes the set of edges representing the relationships between data samples. The similarity matrix $\mathbf{S}$ is a symmetric matrix with each element s_{ij} representing the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. There are three popular approaches to construct graph $\mathcal{G}$, that is, the ε-neighborhood graph, the *k*-nearest neighbor graph, and the fully connected graph (see details in [25]). To partition $\mathcal{G}$ into disjoint subgraphs (clusters), the minimum cut problem requires that the edge weights across different clusters are as small as possible, while the total edge weights within each cluster are as high as possible.

According to the above graph cut theory, two popular versions of spectral clustering are developed, i.e., the ratio cut (RatioCut) and the normalized cut (Ncut). The classical relaxed form of the RatioCut [26] is shown as below:

$$\min \ tr\left(\mathbf{U}^T\mathbf{L}\mathbf{U}\right), \text{s.t.}\mathbf{U}^T\mathbf{U} = \mathbf{I}, \tag{8}$$

where *tr* computes the trace of a matrix, $\mathbf{U} \in \mathbb{R}^{N\times K}$ is the clustering indicator matrix, $\mathbf{I}$ is an identity matrix, and $\mathbf{L}$ is the graph Laplacian matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{S}$. Here, $\mathbf{D}$ is a diagonal matrix with $d_{ii} = \sum_{j=1}^{N} s_{ij}$. The objective function of Ncut [27] is similar to Eq. (8) by replacing $\mathbf{L}$ by the normalized Laplacian matrix $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}$. Both RatioCut and NCut can be solved efficiently by the eigenvalue decomposition (EVD) of $\mathbf{L}$ or $\tilde{\mathbf{L}}$.

3.2.2. Basic form of multi-view spectral clustering

Multi-view spectral clustering is able to learn the latent cluster structures by fusing the information contained in multiple graphs. Similar to multi-view *k*-means, it is not hard to extend the basic spectral clustering to the multi-view environment. Given a dataset $\mathfrak{X} = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(\mathbf{V})}\right\}$ with V views, V graphs $\left\{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(\mathbf{V})}\right\}$ and the corresponding Laplacian matrices $\left\{\mathbf{L}^{(1)}, \mathbf{L}^{(2)}, \ldots, \mathbf{L}^{(\mathbf{V})}\right\}$ can be constructed.

Kumar et al. [28] firstly present a multi-view spectral clustering approach, which has a flavor of co-training idea widely used in semi-supervised learning. It follows the consistency of multi-view learning that each view gives the same labels for all data samples. So it can use the eigenvector of one view to "label" another view and vice versa. For example, via computing two views' eigenvectors, say $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$, the clustering result of $\mathbf{U}^{(1)}$ can be used to modify the graph similarity matrix $\mathbf{S}^{(2)}$, and then the clustering result of $\mathbf{U}^{(2)}$ can be used to modify the graph similarity matrix $\mathbf{S}^{(1)}$. For more than two views, the same strategy can be applied. Kumar et al. [29] further propose a multi-view spectral clustering approach using co-regularization idea that makes the clustering results of different views agree with each other. The co-regularization form is stated as the disagreement between clustering results of two views: $\Phi\left(\mathbf{U}^{(p)}, \mathbf{U}^{(q)}\right) = -tr\left(\mathbf{U}^{(p)}\mathbf{U}^{(p)T}\mathbf{U}^{(q)}\mathbf{U}^{(q)T}\right)$. Then the goal is to minimize the disagreement to achieve the consistency between views with the following objective function:

$$\min \sum_{v=1}^{V} tr\left(\mathbf{U}^{(v)T}\mathbf{L}^{(v)}\mathbf{U}^{(v)}\right) - \sum_{p,\, q=1}^{V} \lambda_{pq} tr\left(\mathbf{U}^{(p)}\mathbf{U}^{(p)T}\mathbf{U}^{(q)}\mathbf{U}^{(q)T}\right), \text{s.t.}\mathbf{U}^{(v)T}\mathbf{U}^{(v)} = \mathbf{I}, \tag{9}$$

where λ_{pq} represents the degree of disagreement between the p-th view and the q-th view. From another perspective, all the views sharing a common indicator matrix $\mathbf{U}^*$ is also rational according to the consistency requirement. So the model in Eq. (9) can be rewritten as

$$\min \sum_{v=1}^{V} tr\left(\mathbf{U}^{(v)T}\mathbf{L}^{(v)}\mathbf{U}^{(v)}\right) - \sum_{v=1}^{V} \lambda_v tr\left(\mathbf{U}^{(v)}\mathbf{U}^{(v)T}\mathbf{U}^*\mathbf{U}^{*T}\right), \text{s.t.}\mathbf{U}^{(v)T}\mathbf{U}^{(v)} = \mathbf{I}, \tag{10}$$

where λ_v controls the degree of disagreement between $\mathbf{U}^{(v)}$ and $\mathbf{U}^*$.

3.2.3. Variants of multi-view spectral clustering

The basic form of multi-view spectral clustering achieves the basic goals of multi-view learning. However, some issues have not yet been considered. For instance, the weight parameter λ in Eq. (9) needs to be set manually. To make up this issue, it is necessary to adaptively compute the weight of each view. Xia et al. [30] assume that each view has a weight μ_v representing its importance and the weight distribution should be sufficiently smooth. They further consider a unified indicator matrix $\mathbf{U}$ across all views, which can be fulfilled via exploring the complementary property of different views. To this end, they develop a novel model as follows:

$$\min \sum_{v=1}^{V} \mu_v^{\gamma} tr\left(\mathbf{U}^T \mathbf{L}^{(v)} \mathbf{U}\right), \text{s.t.} \sum_{v=1}^{V} \mu_v = 1, \mu_v > 0. \tag{11}$$

The model above needs a manually specified parameter γ to adjust the weights of different views, which is sometimes intractable. Thus, Nie et al. [31] propose a parameter-free auto-weighted multiple graph learning method (AMGL), wherein the weight parameter μ_v is replaced by $\alpha_v = \frac{1}{2}\sqrt{tr\left(\mathbf{U}^T \mathbf{L}^{(v)} \mathbf{U}\right)}$. Thus, AMGL does not require additional parameters, and α_v can be self-updated. To avoid the considerable noise in each view which often degrades the performance severely, Xia et al. [32] propose a robust multi-view spectral clustering (RMSC) method via low-rank and sparse decomposition. In RMSC, a novel Markov chain is designed for dealing with the noise. First, the similarity matrix $\mathbf{S}^{(v)}$ and the corresponding transition probability matrix $\mathbf{P}^{(v)} = \left(\mathbf{D}^{(v)}\right)^{-1} \mathbf{S}^{(v)}$ are computed. Then, the row-rank latent transition probability matrix $\widehat{\mathbf{P}}$ and the deviation error matrix $\mathbf{E}^{(v)}$ are constructed via low-rank and sparse decomposition. Finally, based on the transition probability matrix $\widehat{\mathbf{P}}$, the standard Markov chain method is applied for partitioning data into K clusters. Note that the methods above have a high cost in optimization computation. There are numerous variables that need to be updated and the derivation process is also extremely complex during the optimization. To overcome this limitation, Chen et al. [33] present a novel variant of the Laplacian matrix named block intra-normalized Laplacian defined as follows, without the linear combination of multiple Laplacian matrices.

$$\mathbf{B} = \mathbf{B}_w + \beta \mathbf{B}_a = \begin{pmatrix} \mathbf{L}^{(1)} & 0 & \cdots & 0 \\ 0 & \mathbf{L}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{L}^{(V)} \end{pmatrix} + \beta \begin{pmatrix} (V-1)\mathbf{I} & -\mathbf{I} & \cdots & -\mathbf{I} \\ -\mathbf{I} & (V-1)\mathbf{I} & \cdots & -\mathbf{I} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{I} & -\mathbf{I} & \cdots & (V-1)\mathbf{I} \end{pmatrix}, \tag{12}$$

where $\mathbf{B}_w$ denotes the within Laplacian matrix of V views and $\mathbf{B}_a$ denotes the across Laplacian matrix between different views. Based on $\mathbf{B}$, the block intra-normalized Laplacian matrix is then defined as $\widehat{\mathbf{B}} = \mathbf{D}^{-1/2} \mathbf{B}_w \mathbf{D}^{-1/2} + \beta \mathbf{B}_a$, where $\mathbf{D}$ is a block diagonal matrix with the v-th block being $\mathbf{D}^{(v)}$. By proving that the multiplicity of the zero eigenvalue of the constructed block Laplacian matrix is equal to the number of clusters K, the eigenvectors of the block

Laplacian matrix can be used for clustering via the classical form of spectral clustering. At the end, the lower and upper bounds of the optimal solution are also established. See [33] for more details.

3.3. Multi-view clustering via matrix factorization

In the fields of data mining and machine learning, matrix factorization (MF) is an effective latent factor learning model. Given a data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$, MF tries to find two low-rank factor matrices $\mathbf{V} \in \mathbb{R}^{M \times K}$ and $\mathbf{U} \in \mathbb{R}^{N \times K}$ whose multiplication can well approximate it, i.e., $\mathbf{X} \approx \mathbf{V}\mathbf{U}^T$. MF has shown many promising applications in real world, such as information retrieval, recommendation system, signal processing, document analysis, and so on. Usually, the nonnegativity constraints are enforced to the factor matrices to promote the interpretability of the MF models. Therefore, in this part, we focus on the introduction of the nonnegative MF (NMF)-related clustering models. For a comprehensive review of NMF-based models and applications, please refer to [34].

3.3.1. Preliminaries of matrix factorization

As is well known, there are many matrix factorization models, including the singular value decomposition, Cholesky decomposition, LU decomposition, QR decomposition, and Schur decomposition. These factorization models either have too strict restrictions on the factor matrices or lack the ability to be applied to data analysis. Due to the wide applications of NMF in recommending systems, NMF has drawn much attention in both academia and industry. In fact, NMF can be regarded as an extension of the standard *k*-means algorithm by relaxing the constraints imposed on the clustering indicator matrix. For a given dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}_+^{M \times N}$, NMF seeks to learn a basis matrix $\mathbf{V}$ and a coefficient matrix $\mathbf{U}$ via optimizing the following objective function:

$$\min_{\mathbf{V},\,\mathbf{U}} \|\mathbf{X} - \mathbf{V}\mathbf{U}^T\|_F^2, \text{ s.t.} \mathbf{V} \geq 0, \mathbf{U} \geq 0, \tag{13}$$

where $\mathbf{V} \in \mathbb{R}_+^{M \times K}$ can be considered as the cluster centroid matrix and $\mathbf{U} \in \mathbb{R}_+^{N \times K}$ can be treated as a "soft" clustering indicator matrix. The objective function above is not convex in $\mathbf{U}$ and $\mathbf{V}$; therefore, it is impractical to find the global optima. Typically, there are two methods to solve Eq. (13). The first one is the gradient descent method [35]. The other one is the multiplicative method [36] where the iterative updating rules are as follows:

$$\mathbf{V} \leftarrow \mathbf{V} \odot \frac{\mathbf{X}\mathbf{U}}{\mathbf{V}\mathbf{U}^T\mathbf{U}}, \quad \mathbf{U} \leftarrow \mathbf{U} \odot \frac{\mathbf{X}^T\mathbf{V}}{\mathbf{U}\mathbf{V}^T\mathbf{V}}, \tag{14}$$

where $\odot$ and $\frac{[\cdot]}{[\cdot]}$ denote the element-wise multiplication and division, respectively. It is noteworthy that there are many other criterions to measure the difference between $\mathbf{X}$ and $\mathbf{V}\mathbf{U}^T$, such as the ℓ_1-norm, the $\ell_{2,1}$-norm, and the Kullback-Leibler divergence (a.k.a. relative entropy). For these criterions, the updating rules can be derived similarly.

3.3.2. Basic form of multi-view matrix factorization

The hypothesis behind multi-view clustering is that different views should admit the same underlying clustering structures of the datasets. That is, the coefficient matrices learned from different views should be as consistent as possible. To this end, a soft regularization term is introduced to enforce the coefficient matrices of different views toward a common consensus [37]. For a given dataset $\mathfrak{X} = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(V)}\right\}$ with V views, the following objective function can be derived to partition $\mathfrak{X}$ into K clusters:

$$\min_{\mathbf{V}^{(v)},\, \mathbf{U}^{(v)}} \sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{V}^{(v)}\mathbf{U}^{(v)T}\|_F^2 + \sum_{v=1}^{V} \lambda_v \|\mathbf{U}^{(v)} - \mathbf{U}^*\|_F^2, \text{ s.t.} \mathbf{V}^{(v)} \geq 0, \mathbf{U}^{(v)} \geq 0, \mathbf{U}^* \geq 0, \tag{15}$$

where $\mathbf{U}^*$ is a consensus matrix that characterizes the intrinsic clustering structures of datasets among all views and λ_v is the parameter used to tune both the relative importance of different views and the contribution between the first reconstruction error term and the second disagreement term. Note that Eq. (15) does not require that all the views share a common $\mathbf{U}^*$; thus, this model is more robust to low-quality views, i.e., the effect of low-quality views is reduced by setting the corresponding λ_v to be small enough.

Instead of enforcing a rigid common consensus constraint on all the views as in Eq. (15), another form of basic multi-view NMF for clustering is the pair-wise CoNMF model [38], which imposes similarity constraints on each pair of views. Through the pair-wise co-regularization, it is expected that the coefficient matrices learned from two views can complement with each other during the factorization process. And therefore, high-quality clustering results can be yielded. The co-regularization objective function of the pair-wise CoNMF model is defined intuitively as follows:

$$\min_{\mathbf{V}^{(v)},\, \mathbf{U}^{(v)}} \sum_{v=1}^{V} \lambda_v \|\mathbf{X}^{(v)} - \mathbf{V}^{(v)}\mathbf{U}^{(v)T}\|_F^2 + \sum_{p,\, q=1}^{V} \lambda_{pq} \|\mathbf{U}^{(p)} - \mathbf{U}^{(q)}\|_F^2, \text{ s.t.} \mathbf{V}^{(v)} \geq 0, \mathbf{U}^{(v)} \geq 0, \tag{16}$$

where λ_v is the parameter employed to combine the factorization of different views and λ_{pq} is the parameter used to denote the weight of similarity constraint on $\mathbf{U}^{(p)}$ and $\mathbf{U}^{(q)}$. As the column vector of the coefficient matrix $\mathbf{U}$ represents a cluster, when adopting the vector-based ℓ_2-norm, each element of $\mathbf{U}^T\mathbf{U}$ gives the cosine similarity between two clusters. Obviously, in the multi-view environment, the cluster similarity between different views should also be consistent, which results in the cluster-wise CoNMF model. Cluster-wise CoNMF replaces the pair-wise regularization term in Eq. (16) by the following cluster-wise regularization term:

$$\sum_{p,\, q=1}^{V} \lambda_{pq} \|\mathbf{U}^{(p)T}\mathbf{U}^{(p)} - \mathbf{U}^{(q)T}\mathbf{U}^{(q)}\|_F^2. \tag{17}$$

Similar to the optimization of the standard single-view NMF model, all the three basic multi-view NMF clustering models can be optimized via the multiplicative updating rules.

3.3.3. Variants of multi-view matrix factorization

As the locality preserving learning and the manifold learning have been shown very important to promote the performance of clustering algorithms, Cai et al. [39] propose a graph (or manifold) regularized NMF model GNMF for single-view clustering with satisfying performance. Note that the aforementioned multi-view NMF models cannot preserve the local geometrical structures of the samples. To overcome this limitation, a multi-manifold regularized NMF model (MMNMF) is proposed in [40]. MMNMF incorporates consensus manifold and consensus coefficient matrix with multi-manifold regularization to preserve the local geometrical structures of the multi-view data space. The multi-manifold regularization has also been considered in [41]. Moreover, the correntropy-induced metric (CIM) is adapted to measure the reconstruction error, since CIM has achieved excellent performance in many applications. CIM is also insensitive to large errors that are mainly introduced from heavy noises. A much simpler formulation of the manifold regularized multi-view NMF model is developed in [42]. Without the explicit constraint that enforces a rigid common manifold consensus, an auxiliary matrix is involved to add constraints on the column sums of the basis matrix $\mathbf{V}^{(v)}$ such that the coefficient matrix $\mathbf{U}^{(v)}$ is comparable. A weighted extension of multi-view NMF is presented in [43] to address the image annotation problem. In this model, two weight matrices are introduced. One weight matrix is used to bias the factorization toward improved reconstruction for rare tags. The other weight matrix gives more weight to images containing rare tags and is applied to all views. A weighted extension of the pair-wise CoNMF model has also been developed in [44] to handle those attributes that are unobserved in each data sample so as to resolve the sparseness problem in all views' matrices. For the realistic cases that many views suffer from missing of some data samples resulting in many partial examples, Li et al. [45] firstly devise a partial multi-view clustering method to handle this problem. A multi-incomplete-view clustering method MIC [46] is also designed to deal with the incompleteness of the views. MIC is built upon the weighted NMF model with a $\ell_{2,1}$-norm regularization. Zhang et al. [47] further propose a constrained multi-view clustering algorithm for unmapped data in the framework of NMF. The proposed algorithm uses inter-view constraints to establish the connections between different views.

Due to its great interpretability and high efficacy, NMF has been widely employed for graph clustering [48]. In such setting, the data matrix $\mathbf{X}$ is replaced by the adjacency matrix $\mathbf{A}$. In many applications, graph data may be collected from heterogeneous domains or sources. Integrating multiple graphs has been shown to be a promising approach to improve the graph clustering accuracy. Clearly, multi-view NMF is suitable for multiple graph processing. In [49], a flexible and robust NMF-based framework, named co-regularized graph clustering (CGC), is developed to address the multi-domain graph clustering problem. CGC supports many-to-many cross-domain node relationships, and it also incorporates weights on cross-domain relationships. Besides, CGC allows partial cross-domain mapping so that graphs in different domains may have different sizes. Considering the fact that in many real-world applications, different graphs have different node distributions, the assumption that the multiple graphs share a common clustering structure does not hold. Given this, Ni et al. [50] develop a novel two-phase clustering method NoNClus, based on the NMF framework. At first, a main graph

is constructed via modeling the similarity between different domains. Then, the main graph is utilized to regularize the clustering structures in different domain-specific graphs. In the NonClus model, multiple underlying clustering structures can co-exist among domain-specific graphs, while for similar domains, the corresponding clustering structures should be as consistent as possible.

3.4. Multi-view clustering via tensor decomposition

In this part, we analyze multi-view clustering from a multilinear algebra perspective and present several novel multi-view clustering algorithms (note that the notations used in this part are self-contained). Tensor is known as a multidimensional matrix or multiway array [51]. In multi-view research field, data can be naturally modeled as a third-order tensor with objects, features, and view dimensions. An intuitive way is to compact different views along the view dimension of the tensor (see **Figure 1**). Another widely adopted way is to transform each feature matrix to a similarity matrix before compacting them.

3.4.1. Preliminaries of tensor decomposition

In the field of data mining and machine learning, tensor decomposition is an emerging and effective tool for processing multi-view data. In this section, some basic knowledge on tensors and tensor decomposition methods is provided. We refer the readers to [51, 52] for a comprehensive understanding of these topics.

3.4.1.1. Notations

Let $\mathcal{X}$ be an m-order tensor of size $I_1 \times I_2 \times \cdots \times I_m$. The mode-$p$ matricization of $\mathcal{X}$ is denoted as an $I_p \times \left(I_1 \cdots I_{p-1} I_{p+1} \cdots I_m\right)$ matrix $\mathbf{X}_{(p)}$, which is obtained by arranging the mode-p fibers to be the columns of the matrix $\mathbf{X}_{(p)}$. The p-mode multiplication $\mathcal{Y} = \mathcal{X} \times_p \mathbf{U}$ can be manipulated as matrix multiplication $\mathbf{Y}_{(p)} = \mathbf{U}\mathbf{X}_{(p)}$, where $\mathbf{U} \in \mathbb{R}^{J_p \times I_p}$ and $\mathcal{Y} \in \mathbb{R}^{I_1 \cdots I_{p-1} J_p I_{p+1} \cdots I_m}$. The Frobenius norm of a tensor $\mathcal{X}$ is the sum of the squares of all its elements $x_{i_1 i_2 \ldots i_m}$. The tensor $\mathcal{X}$ is a rank-one tensor if it can be written as the outer product of m vectors, i.e., $\mathcal{X} = \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \ldots \circ \mathbf{x}^{(m)}$, where $\circ$ represents the vector outer product.

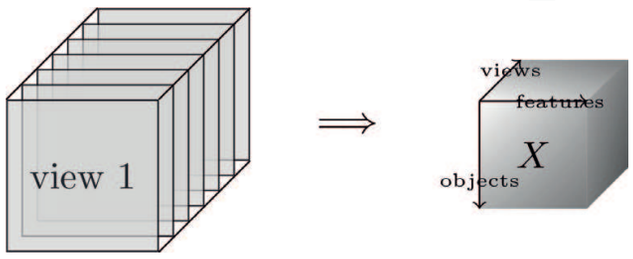

Figure 1. Visualization of the process of transforming the feature matrices to a third-order tensor.

3.4.1.2. CP decomposition

The idea of expressing tensor as the sum of a number of rank-one tensors comes from the study of Hitchcock [53]. Then, Cattell [54] proposed the idea of parallel proportional analysis. The popular CP decomposition comes from the ideas of Carroll and Chang [55] (canonical decomposition) and Harshman [56] (parallel factors). Taking a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ as an example, the CP decomposition tries to approximate tensor $\mathcal{X}$ with R components of rank-one tensor, i.e.,

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r, \tag{18}$$

where $\mathbf{u}_r \in \mathbb{R}^I$, $\mathbf{v}_r \in \mathbb{R}^J$, and $\mathbf{w}_r \in \mathbb{R}^K$. For simplicity, we denote $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_R]$, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_R]$, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_R]$, and $[\![\mathbf{U}, \mathbf{V}, \mathbf{W}]\!]$ as the CP decomposition of $\mathcal{X}$.

3.4.1.3. Tucker decomposition

The idea of Tucker decomposition is introduced by Tucker [57]. The Tucker decomposition is a form of higher-order singular value decomposition (HOSVD) [58]. It decomposes a tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ into a core tensor $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ multiplied by several orthogonal matrices along each mode, i.e.,

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{u}_p \circ \mathbf{v}_q \circ \mathbf{w}_r. \tag{19}$$

The cutting-edge technique for calculating the factor matrices is proposed in [59].

3.4.2. Tensor decomposition-based multi-view clustering

In multi-view clustering, the goal is to find out some meaningful group of objects from the data. The above CP decomposition naturally divides the multi-view data into several components, which can be seen as the clusters. Thus, it can be directly applied to solve multi-view clustering problems. For a given dataset $\mathcal{X} = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(V)}\right\}$ with V views, where $\mathbf{X}^{(v)}$ of each view takes value from $\mathbb{R}^{N \times M}$, $\mathcal{X}$ can be formulated as a third-order tensor $\mathcal{X} \in \mathbb{R}^{N \times M \times V}$. In this part, a variant of CP decomposition is introduced first, which is quite straightforward. Then we shed light on the relations between several classic multi-view spectral clustering methods and the Tucker decomposition.

3.4.2.1. Total variation based CP (TVCP)

In some clustering problems, a consecutive range of time points is non-negligible. For example, in the dataset with authors, publications, and a sequence of time points, we are interested in figuring out which group of authors work in the same topics during a period of time. Chen et al. [60] propose a total variation based tensor decomposition method (TVCP) for the

constraint on a period of consecutive time points. The total variation regularizes the time factor to obtain a piece-wise constant function w.r.t. time points. Owing to the piece-wise constant function, the decomposition can be relatively consistent in a cluster and separated between clusters. The TVCP model is formulated as follows:

$$\min_{[\![\mathbf{U},\mathbf{V},\mathbf{W}]\!]} \frac{1}{2}\|\mathcal{X} - \sum_{r=1}^{R} \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r\|_F^2 + \tau \sum_{r=1}^{R} \|\mathbf{F}\mathbf{w}_r\|_1, \tag{20}$$

where $\mathbf{F}$ is the first-order difference $(V-1)\times V$ matrix such that $f_{ii}=1$ and $f_{i(i+1)}=-1$ for $i=1,2,\cdots,V-1$, and the other elements are zeros, τ is a positive regularization parameter, and $\|\cdot\|_1$ denotes the ℓ_1-norm. The first term corresponds to the CP decomposition of $\mathcal{X}$, and the second term constrains the time mode ($\mathbf{w}$) to be a piece-wise constant function.

3.4.2.2. Relations between Tucker decomposition and spectral clustering

Liu et al. [61] propose a framework of multi-view clustering via tensor decomposition, mainly the Tucker decomposition. According to the framework, the common type of multi-view spectral clustering is equivalent to a Tucker decomposition problem as follows:

$$\min_{\mathbf{U}} \quad \sum_{v=1}^{V} tr\left(\mathbf{U}^T\mathbf{L}^{(v)}\mathbf{U}\right), \text{s.t.}\mathbf{U}^T\mathbf{U}=\mathbf{I}, \quad \Leftrightarrow \quad \max_{\mathbf{U}} \|\mathcal{X}\times_1\mathbf{U}^T\times_2\mathbf{U}^T\times_3\mathbf{I}^T\|_F^2. \tag{21}$$

Another form of multi-view spectral clustering can also be written as a Tucker problem:

$$\begin{aligned} &\min_{\mathbf{U},\,\mu} \; tr\left(\mathbf{U}^T\left(\sum_{v=1}^{V}\mu_v\mathbf{L}^{(v)}\right)\mathbf{U}\right), && \max_{\mathbf{U},\,\mu} \|\mathcal{X}\times_1\mathbf{U}^T\times_2\mathbf{U}^T\times_3\mu^T\|_F^2, \\ &\text{s.t.}\mathbf{U}^T\mathbf{U}=\mathbf{I}, \mu_v\geq 0, \sum_{v=1}^{V}\mu_v=1, && \text{s.t.}\mathbf{U}^T\mathbf{U}=\mathbf{I}, \mu_v\geq 0, \sum_{v=1}^{V}\mu_v=1. \end{aligned} \tag{22}$$

With this framework, variety of spectral clustering problems can be solved by a tensor decomposition algorithm. We can see the strong connection between them as well as the strong capability of tensor methodology.

Canonical correlation analysis is designed to inspect the linear relationship between two sets of variables [62]. In multi-view learning, a typical approach is to maximize the sum of pair-wise correlations between different views [63]. Without loss of high-order correlations, Luo et al. [64] propose a tensor canonical correlation analysis (TCCA), which is equivalent to CP decomposition of the correlation tensor. Khan et al. [65] propose a Bayesian extension of CP decomposition for multiple coupled tensors sharing common latent factors.

3.5. Multi-view clustering via deep learning

With the third wave of artificial intelligence, deep learning is gaining increasing popularity in recent years. Deep learning has demonstrated excellent performance in many real-world

applications, such as face recognition, image annotation, natural language processing, object detection, customer relationship management, and mobile advertising. Typically, deep learning models are composed of multiple nonlinear transformations and thus can learn a better feature representation than traditional shallow models [66]. However, deep learning requires labeled training data to learn the models, which limits its application in data clustering for the reason that training data with cluster labels are not available in many cases. Despite the hardness, there are some works devoted to adjusting shallow clustering models for deep learning. Here, we introduce two popular deep clustering models and their extensions to the multi-view environment.

3.5.1. Deep auto-encoder

An auto-encoder [67] is an artificial neural network adopted for unsupervised learning, the goal of which is to learn a representation for each data sample. An auto-encoder always consists of two parts: the encoder and the decoder. The encoder plays the role of a nonlinear mapping function that can map each data sample to a representation space. The decoder demands accurate data reconstruction from the representation generated by the encoder. Auto-encoder has been shown to be similar to spectral clustering in theory; however, it is more efficient and flexible in practice. The auto-encoder can be easily deepened via adding more encoder layers and corresponding decoder layers. **Figure 2 (a)** gives an example of the framework of the deep auto-encoder.

Although auto-encoder can learn a compact representation for each data sample, it contributes little to clustering since it does not require that the representation vectors of similar data samples should also be similar. To make the learned feature representation better capture the cluster structures, many variants of deep auto-encoder models have been proposed. In [68], a novel regularization term that is similar to the objective function of *k*-means is introduced to guide the learning of the mapping function. In this way, the learned feature representation is more stable and suitable for clustering. In [69], a deep embedded clustering method is proposed to simultaneously learn feature representations and cluster assignments using deep auto-encoders. These deep clustering models are designed for single-view data. For deep multi-view clustering, the learned feature representations should not only capture the cluster structure of each single view but also implement a consensus between different views. To this end, a common encoder is utilized to extract the shared feature representation for all views,

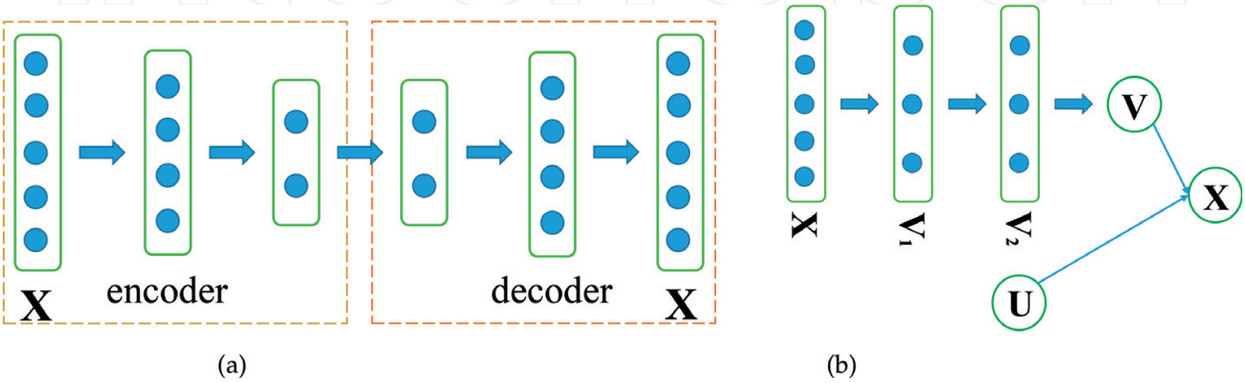

Figure 2. Frameworks of deep auto-encoder and deep matrix factorization (depth is 2).

and different decoders are used to reconstruct view-specific input data samples [70]. In [71], an extension of CCA based on deep neural networks is proposed to learn a shared representation of two views. In fact, the feature representations of the two views are not exactly the same, but their correlations are maximized. Following this line, the deep canonically correlated auto-encoder (DCCAE) is developed in [72]. DCCAE simultaneously optimizes the canonical correlation between the learned feature representations and the reconstruction errors of the auto-encoders. Benton et al. [73] further extend the deep CCA model for multiple views.

3.5.2. Deep matrix factorization

Another line of developing deep clustering models is deepening the MF models. As shown earlier, MF, especially NMF, has demonstrated outstanding performance in many applications. Thus, it is worth building a deep structure for MF in the hope that better feature representations can be obtained to facilitate clustering. **Figure 2(b)** illustrates an example of the framework of the deep MF models. Compared to the deep auto-encoders, both deep MF and deep auto-encoders are trying to minimize the reconstruction errors. However, unlike deep auto-encoders, the mapping function of deep MF is linear.

The first nonnegative deep network based on NMF is proposed in [74] for speech separation. This architecture can be discriminatively trained for optimal separation performance. Then Li et al. [75] propose a novel weakly supervised deep MF model to uncover the latent image representations and tag representations embedded in the latent subspace by collaboratively exploring the weakly supervised tagging information, the visual structure, and the semantic structure. In [76], a deep semi-NMF model is further developed for learning latent attribute representations. Semi-NMF is a popular variant of NMF by relaxing the factorized basis matrix to be real-valued. This practice makes semi-NMF have much wider applications than NMF since the datasets in real world may contain complex information, for instance, the attributes may be mix-signed. Considering the fact that these deep MF models are trying to factorize the basis matrix hierarchically alone, Qiu et al. [77] further propose a deep orthogonal NMF model which can decompose the coefficient matrix hierarchically. This model is able to learn higher-level representations for clusters. These deep MF models have achieved great success in data clustering for single-view data. However, they are seldom utilized for multi-view clustering. A recent work [78] attempts to extend the deep semi-NMF model for multi-view clustering, which can dissemble unimportant factors layer by layer and generate an effective consensus representation in the last layer. Another work [79] proposes to address the incomplete multi-view clustering problem via deep semantic mapping. The proposed model first projects all incomplete multi-view data to a unified representation in a common subspace, which is further executed by standard shallow NMF for clustering.

4. Open datasets

No one can make bricks without straw. In this section we will first list two kinds of open datasets that can be used in multi-view clustering, i.e., feature-based and graph-based datasets. Then we will discuss the performance of multi-view clustering on them briefly.

4.1. Feature-based datasets

Audio genre [80] consists of 1886 audio tracks classified into 9 music genres, which are Blues, Electronic, Jazz, Pop, Rap/HipHop, Rock, Folk/Country, Alternative, and Funk/Soul. Forty-nine low-level audio features have been extracted and they are grouped into 15 vector spaces.

NUS-WIDE [81] is a web image dataset composed of 269,648 images, 5018 related tags, and 81 ground-truth concepts. Six types of low-level features have been extracted: 64-D color histogram, 144-D color correlogram, 73-D edge direction histogram, 128-D wavelet texture, 225-D block-wise color moments extracted over 5×5 fixed grid partitions, and 500-D bag of words based on SIFT descriptions.

UCF101 [82] consists of 101 human action classes. These actions can be divided into five types: human-object interaction, body-motion only, human-human interaction, playing musical instruments, and sports. There are over 13,000 clips and 27 hours of video data in it.

Handwritten numerals [83] is composed of 2000 handwritten digits which are divided into 10 classes. Four types of feature sets have been extracted: Zernike moments, Karhunen-Loeve features, Fourier descriptors, and image vectors. For Zernike set, it has 47 rotation invariant Zernike moments and 6 morphological features. For Fourier set, it has 76 two-dimensional shape descriptors. Both Zernike and Fourier feature sets are rotation invariant. For Karhunen-Loeve set, it has 64 Karhunen-Loeve transform which corresponds to the projection of images onto the eigenvectors of a covariance matrix.

4.2. Graph-based datasets

DBLP coauthorship [84] is a coauthorship network composed of 10,305 authors. There are 617 layers in it, each layer representing different publication categories.

Facebook [85] is a three-layer social network composed of 1640 users with multiple types of ties. The first layer shows whether two users are friends. The second layer shows whether users are in a same group. The third layer shows whether users are in the same photos uploaded by users.

CiteSeer [86] consists of 3312 scientific publications classified into 6 classes, which are Agents, AI, DB, IR, ML, and HCI. It can be represented as an annotated network, where nodes represent scientific publications and links represent the citation relationships. For each node, there is a 3703-dimensional one-hot encoding vector representing the absence/presence of key words.

Enron e-mail [87] consists of 184 users and 44 layers. Although it is a temporal network, it can be considered as a multi-layer network. Each layer represents communication in different months.

4.3. Performance on different datasets

For feature-based datasets, when confronted with the situation where we need to reconstruct the views, the performance of classical methods, like deep learning, is not promising. But

multi-view clustering can give satisfactory results under this condition. In some cases, classical methods can also give good performance for feature-based datasets where all features are descriptions of the same object from different perspectives. For graph-based datasets, multi-view clustering naturally fits into them since different graphs can be processed by different views.

For both feature-based and graph-based datasets, when the scale of datasets becomes significantly large, most multi-view clustering methods have the potential to outperform other clustering methods on speed. For example, multi-view matrix factorization is quite suitable to parallel process.

5. Open issues

Although multi-view clustering has demonstrated its superiority over single-view clustering in many applications, there are still many open issues deserving much more attention from both academia and industry. Several vital open issues are summarized in this part.

5.1. View construction

Although there are many typical methods to construct views, they all have their own drawbacks. It is well known that if we cannot extract valuable information from the original data and put it into different views appropriately, the performance will be highly limited no matter how delicate the algorithm is. So it is important to find efficient ways of constructing and evaluating multiple views.

5.2. Incomplete view

When constructing different views, we may find that for some views, the information is not complete. In other words, even though we know how to construct views appropriately, we do not have enough information to do it, which is very common in practical problems. In real world, it is very difficult to ensure the completeness of data. This unbalanced relationship between complete views and incomplete views could cause huge problems. Moreover, these incomplete views may influence views with complete information. To solve it, one possible way is to construct these lost information from other views.

5.3. Single-view to multi-view

In multi-view learning, sometimes researchers will convert single-view data into multiple views and apply relevant algorithms on them. In practice, it may give good performance, but there are few theoretical researches on the proof of its reliability. Since the original data is single view, it is important to make it clear: is it necessary to complicate a simple task? We should not only focus on the final performance, the trade-off between cost and benefit is also important.

5.4. Deep leaning in multi-view

Deep learning has shown remarkable performance in many fields. One common way to deal with data composed of different types of sources is to combine them together and then feed them into a deep learning model. It often works well. Although multi-view learning seems to be a more reasonable way to deal with data composed of different types of sources, there is no evidence showing that multi-view learning has an obvious advantage over deep learning. Another issue is that when using deep learning in multi-view learning, we need to train different neural networks for different views separately. This method has two drawbacks. One is that the number of neural networks depends on the number of views. When there are many views, the calculation is huge. The other is that it fails to unify different views during training.

6. Conclusion

Multi-view clustering has demonstrated variety of real-world applications, such as community detection in social networks, image annotation in computer vision, cross-domain user modeling in recommendation systems, and protein interaction analysis in bioinformatics. This chapter provides a comprehensive review of the typical multi-view clustering methods and their corresponding recent developments by focusing on five most typical and popular clustering methods, which include *k*-means, spectral clustering, matrix factorization, tensor decomposition, and deep learning. The basic forms of these five clustering methods are introduced in detail, followed by a substantial overview of their recent developments. Several open datasets and open issues are discussed in the end, which deserves more attention to facilitate the future research of multi-view clustering.

In the field of multi-view clustering, there are many algorithms whose source codes are exposed by their authors. For example, the co-training[1] and co-regularization[2] methods of classical multi-view spectral clustering are open in GitHub with MATLAB. The variants MSE[3] and AMGL[4] are also implemented by MATLAB.

Author details

Fanghua Ye[1], Zitai Chen[1], Hui Qian[1], Rui Li[2], Chuan Chen[1]* and Zibin Zheng[1]

*Address all correspondence to: chenchuan@mail.sysu.edu.cn

1 School of Data and Computer Science, Sun Yat-sen University, China

2 School of Physics, Sun Yat-sen University, China

[1] https://github.com/areslp/matlab/tree/master/code_cospectral

[2] https://github.com/areslp/matlab/tree/master/code_coregspectral

[3] https://github.com/rciszek/mse

[4] http://www.escience.cn/people/fpnie/index.html;jsessionid = 253C211B5AEDB8C09865FFEAEAACFB73-n1

References

[1] Bickel S, Scheffer T. Multi-view clustering. ICDM. 2004;**4**:19-26

[2] Chang X, Tao D, Xu C. A survey on multi-view learning. arXiv preprint arXiv. 2013;**1304**: 5634

[3] Sun S. A survey of multi-view machine learning. Neural Computing and Applications. Feb 2013;**23**(7–8):2031-2038

[4] Zhao J, Xie X, Xin X, Sun S. Multi-view learning overview: Recent progress and new challenges. Information Fusion. 2017;**38**:43-54

[5] Liu X. Learning from Multi-View Data: Clustering Algorithm and Text Mining Application. Leuven, Belgium: KU Leuven; 2011

[6] Ali Mamdouh E, Yang S, Xiaodong H. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: WWW, International World Wide Web Conferences Steering Committee; 2015. pp. 278-288

[7] Blaschko MB, Lampert CH. Correlational spectral clustering. In: CVPR. IEEE; 2008. pp. 1-8

[8] Kailing K, Kriegel H-P, Pryakhin A, Schubert M. Clustering multi-represented objects with noise. In: PAKDD. Springer Berlin Heidelberg: Springer; 2004. pp. 394-403

[9] Chaudhuri K, Kakade SM, Livescu K, Sridharan K. Multi-view clustering via canonical correlation analysis. In: ICML. ACM; 2009. pp. 129-136

[10] Yin Q, Shu W, He R, Wang L. Multi-view clustering via pairwise sparse subspace representation. Neurocomputing. 2015;**156**:12-21

[11] Jain AK. Data clustering: 50 years beyond k-means. PRL. Jun 2010;**31**(8):651-666

[12] Maldonado S, Carrizosa E, Weber R. Kernel penalized k-means: A feature selection method based on kernel k-means. Information Sciences. 2015;**322**:150-160

[13] Liang D, Zhou P, Shi L, Wang H, Fan M, Wang W, Shen Y-D. Robust multiple kernel k-means using l21-norm. In: IJCAI; 2015

[14] Liu X, Li M, Wang L, Dou Y, Yin J, Zhu E. Multiple kernel k-means with incomplete kernels. In: AAAI: 2017. pp. 2259-2265

[15] Wang S, Gittens A, Mahoney MW. Scalable kernel k-means clustering with nystrom approximation: Relative-error bounds. arXiv preprint arXiv. 2017;**1706**:02803

[16] Zhang R, Rudnicky AI. A large scale clustering scheme for kernel k-means. In: ICPR. Vol. 4. IEEE; 2002. pp. 289-292

[17] Dhillon IS, Guan Y, Kulis B. Kernel k-means. In: SIGKDD. ACM, ACM Press; 2004. pp. 551-556

[18] Tzortzis G, Likas A. Kernel-based weighted multi-view clustering. In: ICDM. IEEE; 2012. pp. 675-684

[19] Cai X, Nie F, Huang H. Multi-view k-means clustering on big data. In: IJCAI; 2013. pp. 2598-2604

[20] Zhao H, Yun F. Dual-regularized multi-view outlier detection. In: IJCAI; 2015. pp. 4077-4083

[21] Chen X, Xiaofei X, Huang JZ, Ye Y. Tw-k-means: Automated two-level variable weighting clustering algorithm for multiview data. TKDE. 2013;**25**(4):932-944

[22] Yu-Meng X, Wang C-D, Lai J-H. Weighted multi-view clustering with feature selection. Pattern Recognition. 2016;**53**:25-35

[23] Bo J, Qiu F, Wang L. Multi-view clustering via simultaneous weighting on views and features. Applied Soft Computing. 2016;**47**:304-315

[24] Xu C, Tao D, Xu C. Multi-view self-paced learning for clustering. In: IJCAI; 2015. pp. 3974-3980

[25] von Luxburg U. A tutorial on spectral clustering. Statistics and Computing. 2007;**17**(4): 395-416

[26] Hagen L, Kahng AB. New spectral methods for ratio cut partitioning and clustering. TCAD. 1992;**11**(9):1074-1085

[27] Shi J, Malik J. Normalized cuts and image segmentation. TCAD. 2000;**22**(8):888-905

[28] Kumar A, Daumé H. A co-training approach for multi-view spectral clustering. In: ICML; 2011. pp. 393-400

[29] Kumar A, Rai P, Daume H. Co-regularized multi-view spectral clustering. In: NIPS; 2011. pp. 1413-1421

[30] Xia T, Tao D, Mei T, Zhang Y. Multiview spectral embedding. SMCB. 2010;**40**(6):1438-1446

[31] Nie F, Li J, Li X et al. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In: IJCAI; 2016. pp. 1881-1887

[32] Xia R, Pan Y, Lei D, Yin J. Robust multi-view spectral clustering via low-rank and sparse decomposition. In: AAAI; 2014, pp. 2149-2155

[33] Chen C, Ng MK, Zhang S. Block spectral clustering methods for multiple graphs. Numerical Linear Algebra with Applications. 2017;**24**:e2075. DOI: 10.1002/nla.2075

[34] Wang Y-X, Zhang Y-J. Nonnegative matrix factorization: A comprehensive review. TKDE. 2013;**25**(6):1336-1353

[35] Kivinen J, Warmuth MK. Additive versus exponentiated gradient updates for linear prediction. In Proceedings of the twenty-seventh annual ACM symposium on Theory of computing

(STOC '95). ACM, New York, NY, USA.1995. pp. 209-218. http://dx.doi.org/10.1145/225058.225121

[36] Lee DD, Sebastian Seung H. Learning the parts of objects by non-negative matrix factorization. Nature. 1999;**401**(6755):788-791

[37] Liu J, Wang C, Gao J, Han J. Multi-view clustering via joint nonnegative matrix factorization. In: ICDM. SIAM; 2013. pp. 252-260

[38] He X, Kan M-Y, Xie P, Chen X. Comment-based multi-view clustering of web 2.0 items. In: WWW. ACM, ACM Press; 2014. pp. 771-782

[39] Cai D, He X, Han J, Huang TS. Graph regularized nonnegative matrix factorization for data representation. PAMI. 2011;**33**(8):1548-1560

[40] Zong L, Zhang X, Zhao L, Hong Y, Zhao Q. Multi-view clustering via multi-manifold regularized non-negative matrix factorization. Neural Networks. 2017;**88**:74-89

[41] Weihua O, Shujian Y, Li G, Jian L, Zhang K, Xie G. Multi-view non-negative matrix factorization by patch alignment framework with view consistency. Neurocomputing. 2016;**204**:116-124

[42] Hidru D, Goldenberg A. Equinmf: Graph regularized multiview nonnegative matrix factorization. arXiv preprint arXiv. 2014;**1409**:4018

[43] Kalayeh MM, Idrees H, Shah M. Nmf-knn: Image annotation using weighted multi-view non-negative matrix factorization. In: CVPR; 2014. pp. 184-191

[44] Gong X, Wang F, Huang L. Weighted nmf-based multiple sparse views clustering for web items. In: PAKDD. Springer; 2017. pp. 416-428

[45] Li S-Y, Jiang Y, Zhou Z-H. Partial multi-view clustering. In: AAAI; 2014

[46] Shao W, He L, Philip SY. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2015. pp. 318-334

[47] Zhang X, Zong L, Liu X, Yu H. Constrained nmf-based multi-view clustering on unmapped data. In: AAAI; 2015. pp. 3174-3180

[48] Wang F, Li T, Wang X, Zhu S, Ding C. Community discovery using nonnegative matrix factorization. DMKD. 2011;**22**(3):493-521

[49] Cheng W, Zhang X, Guo Z, Yubao W, Sullivan PF, Wang W. Flexible and robust co-regularized multi-domain graph clustering. In: SIGKDD. ACM; 2013. pp. 320-328

[50] Ni J, Tong H, Fan W, Zhang X. Flexible and robust multi-network clustering. In: SIGKDD. ACM; 2015. pp. 835-844

[51] Kolda TG, Bader BW. Tensor decompositions and applications. SIAM Review. Aug 2009; **51**(3):455-500

[52] Sidiropoulos ND, De Lathauwer L, Xiao F, Huang K, Papalexakis EE, Faloutsos C. Tensor decomposition for signal processing and machine learning. SP. 2017;**65**(13):3551-3582

[53] Hitchcock FL. The expression of a tensor or a polyadic as a sum of products. Journal of Mathematical Physics. Apr 1927;**6**(1–4):164-189

[54] Cattell RB. "Parallel proportional profiles" and other principles for determining the choice of factors by rotation. Psychometrika. Dec 1944;**9**(4):267-283

[55] Douglas Carroll J, Chang J-J. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-young" decomposition. Psychometrika. Sep 1970;**35**(3):283-319

[56] Harshman RA. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. UCLA Working Papers in Phonetics. 1970;**16**: 1-84

[57] Tucker LR. Some mathematical notes on three-mode factor analysis. Psychometrika. 1966; **31**(3):279-311

[58] De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications. July 2000;**21**:1253-1278

[59] De Lathauwer L, De Moor B, Vandewalle J. On the best rank-1 and rank-(r1 ,r2 ,…,rn) approximation of higher-order tensors. SIAM Journal on Matrix Analysis and Applications. 2000;**21**(4):1324-1342

[60] Chen C, Li X, Ng MK, Yuan X. Total variation based tensor decomposition for multi-dimensional data with time dimension. Numerical Linear Algebra with Applications. May 2015;**22**(6):999-1019

[61] Liu X, Ji S, Glänzel W, De Moor B. Multiview partitioning via tensor methods. TKDE. 2013; **25**(5):1056-1069

[62] Hardoon DR, Szedmak S, Shawe-Taylor J. Canonical correlation analysis: An overview with application to learning methods. Neural Computation. Dec 2004;**16**(12):2639-2664

[63] Vía J, Santamaría I, Pérez J. A learning algorithm for adaptive canonical correlation analysis of several data sets. Neural Networks. 2007;**20**(1):139-152

[64] Luo Y, Tao D, Ramamohanarao K, Xu C, Wen Y. Tensor canonical correlation analysis for multi-view dimension reduction. TKDE. Nov 2015;**27**(11):3111-3124

[65] Khan SA, Kaski S. Bayesian multi-view tensor factorization. In: Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg: Springer; 2014. pp. 656-671

[66] Zhao L, Chen Z, Yang Z, Yueming H, Obaidat MS. Local similarity imputation based on fast clustering for incomplete data in cyber-physical systems. IEEE Systems Journal. 2016; **PP**(99):1-11

[67] Bengio Y. Learning deep architectures for Al. Foundations and trends® in Machine Learning. 2009;**2**(1):1-127

[68] Song C, Huang Y, Liu F, Wang Z, Liang W. Deep auto-encoder based clustering. Intelligent Data Analysis. 2014;**18**(6S):S65-S76

[69] Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis. In: ICML; 2016. pp. 478-487

[70] Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY. Multimodal deep learning. In: ICML; 2011. pp. 689-696

[71] Andrew G, Arora R, Bilmes J, Livescu K. Deep canonical correlation analysis. In: ICML; 2013. pp. 1247-1255

[72] Wang W, Arora R, Livescu K, Bilmes J. On deep multi-view representation learning: Objectives and optimization. arXiv preprint arXiv. 2016;**1602**:01024

[73] Benton A, Khayrallah H, Gujral B, Reisinger D, Zhang S, Arora R. Deep generalized canonical correlation analysis. arXiv preprint arXiv. 2017;**1702**:02519

[74] Le Roux J, Hershey JR, Weninger F. Deep nmf for speech separation. In: ICASSP. IEEE; 2015, pp. 66-70

[75] Li Z, Tang J. Weakly supervised deep matrix factorization for social image understanding. IP. Jan 2017;**26**(1):276-288

[76] Trigeorgis G, Bousmalis K, Zafeiriou S, Schuller BW. A deep matrix factorization method for learning attribute representations. PAMI. 2017;**39**(3):417-429

[77] Qiu Y, Zhou G, Xie K. Deep approximately orthogonal nonnegative matrix factorization for clustering. arXiv preprint arXiv. 2017;**1711**:07437

[78] Zhao H, Ding Z, Fu Y. Multi-view clustering via deep matrix factorization. In: AAAI; 2017. pp. 2921-2927

[79] Zhao L, Chen Z, Yi Y, Jane Wang Z, Leung VCM. Incomplete multi-view clustering via deep semantic mapping. Neurocomputing. 2018;**275**:1053-1062

[80] Homburg H, Mierswa I, Moller B, Morik K, Wurst M. A benchmark dataset for audio classification and clustering. In: Ismir 2005, Proceedings of the International Conference on Music Information Retrieval, 11–15 September 2005; London. Uk; 2005. pp. 528-531

[81] Chua TS, Tang J, Hong R, Li H, Luo Z, Zheng Y. Nus-wide: a real-world web image database from national university of singapore. In: ACM International Conference on Image and Video Retrieval; 2009. p. 48

[82] Soomro K, Zamir AR, Shah M. Ucf101: A dataset of 101 human actions classes from videos in the wild. CRCV-TR-12-01, November, 2012

[83] Van Breukelen M, Duin RPW, Tax DMJ, Den Hartog JE. Handwritten digit recognition by combined classifiers. Kybernetika. 1998;**34**(4):381-386

[84] Ng KP, Li X, Ye Y. Multirank: co-ranking for objects and relations in multi-relational data. In: SIGKDD; 2011. pp. 1217-1225

[85] Lewis K, Kaufman J, Gonzalez M, Wimmer A, Christakis N. Tastes, ties, and time: A new social network dataset using facebook.Com. Social Networks. 2008;**30**(4):330-342

[86] Sen P, Namata G, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T. Collective classification in network data articles. AI Magazine. 2008;**29**(3):93-106

[87] Bader BW, Harshman RA, Kolda TG. Temporal analysis of semantic graphs using asalsan. In: ICDM; 2007. pp. 33-42