Knowledge-Based Systems 163 (2019) 1009–1019

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A study of graph-based system for multi-view clustering☆

Hao Wang [a,b], Yan Yang [a,*], Bing Liu [b], Hamido Fujita [c]

[a] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[b] Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA
[c] Faculty of Software and Information Science, Iwate Prefectural University, Takizawa 020-0693, Japan

ARTICLE INFO

Article history:
Received 21 June 2018
Received in revised form 11 October 2018
Accepted 13 October 2018
Available online 28 October 2018

Keywords:
Multi-view clustering
Graph-based technology
Data fusion
Laplacian matrix
Rank constraint

ABSTRACT

This paper studies clustering of multi-view data, known as multi-view clustering. Among existing multi-view clustering methods, one representative category of methods is the graph-based approach. Despite its elegant and simple formulation, the graph-based approach has not been studied in terms of (a) the generalization of the approach or (b) the impact of different graph metrics on the clustering results. This paper extends this important approach by first proposing a general *G*raph-*B*ased *S*ystem (GBS) for multi-view clustering, and then discussing and evaluating the impact of different graph metrics on the multi-view clustering performance within the proposed framework. GBS works by extracting data feature matrix of each view, constructing graph matrices of all views, and fusing the constructed graph matrices to generate a unified graph matrix, which gives the final clusters. A novel multi-view clustering method that works in the GBS framework is also proposed, which can (1) construct data graph matrices effectively, (2) weight each graph matrix automatically, and (3) produce clustering results directly. Experimental results on benchmark datasets show that the proposed method outperforms state-of-the-art baselines significantly.

1. Introduction

In many real-world problems data are from different sources or different views. For instance, the same news may be reported by different news organizations, an image may be represented by different types of features, and a video may be encoded in different amounts of images and sounds. All these are referred to as multi-view data, which bred a new learning paradigm, called *multi-view learning*. This learning paradigm is nature because we humans seem to learn in a similar way. We often look at problems from different views. That is why we can approach a problem in a comprehensive manner. The learning paradigm has been studied extensively in the past [1,2]. This paper makes a focused contribution to multi-view unsupervised learning, particularly, *multi-view clustering*. Multi-view clustering considers the diversity of different views and fuses these views to produce a more accurate and robust partition than single-view clustering [3,4]. We will discuss related work in the next section.

Recently, graph-based multi-view clustering has produced several state-of-the-art multi-view clustering methods [5–14]. These methods typically work as follows: (1) constructing a data graph of each view, (2) learning a fusion graph across the constructed graphs of all views, and (3) producing the clustering results on the fusion graph. In these methods, the data graph of each view is usually generated from a data similarity matrix, where each matrix entry is the similarity of two data points. We call this graph matrix the *similarity-induced graph* (SIG) matrix. Although such methods have achieved state-of-the-art performances, most are not comprehensive. First, there is not a general graph-based system for multi-view clustering. We propose a general approach in this paper. Second, the impact of different graph metrics on multi-view clustering has not been discussed. We will show in the experiment section (i.e., Section 6) that the performances of such clustering methods heavily depend on the constructed graphs. Additionally, the current methods, e.g., [5–7], do not give sufficient consideration to weights of different views in fusion or require an additional clustering step after fusion to produce the final clustering results. Most existing methods also need to tune parameters, e.g., [8,10,12,14]. Our proposals in GBS can tackle these problems. All these prompted us to make a new attempt to study multi-view graph-based clustering.

In this paper, we propose a general system for multi-view clustering, named Graph-Based System (GBS), as shown in Fig. 1. GBS first extracts features for data matrices of m views, constructs a SIG matrix $\mathbf{S}^v$ for each view (v), and then generates a unified graph matrix $\mathbf{U}$, which gives the clustering results directly in fusion without an additional clustering step. Specially, we propose a new

☆ For the purpose of reproducibility, the code and datasets are released at: https://github.com/cswanghao/gbs.

* Corresponding author.
E-mail address: yyang@swjtu.edu.cn (Y. Yang).

https://doi.org/10.1016/j.knosys.2018.10.022
0950-7051/

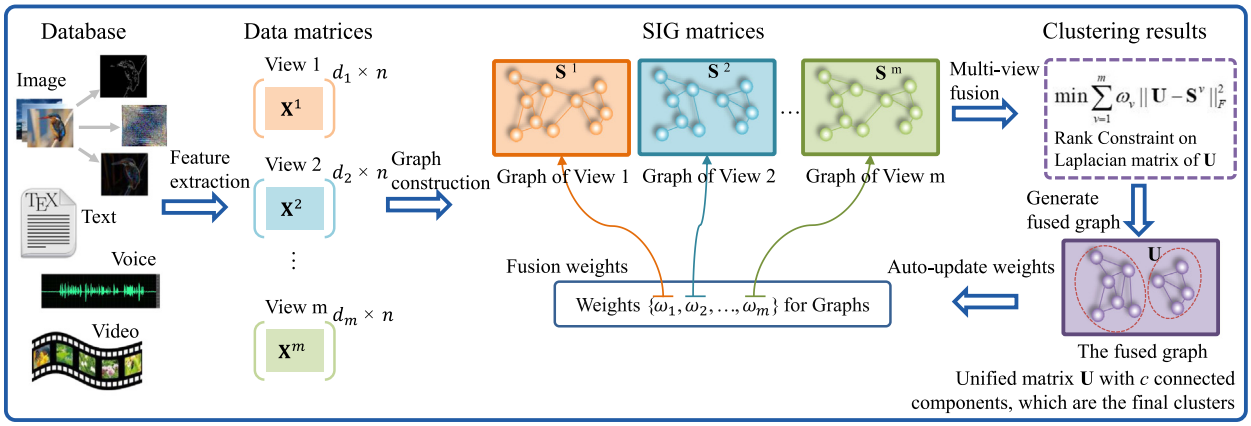

Fig. 1. The proposed GBS for multi-view clustering. The SIG matrices $\mathbf{S} = \{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^m\}$ are constructed from data matrices of m views $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^m\}$. The fused graph (i.e., the unified matrix $\mathbf{U}$) is generated by using the proposed auto-weighted fusion technique, where the weights $\mathbf{w} = \{w_1, w_2, \ldots, w_m\}$ correspond to $\mathbf{S} = \{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^m\}$. With a rank constraint on the Laplacian matrix of $\mathbf{U}$, the learned unified matrix $\mathbf{U}$ has c connected components, where c is the user-specified number of clusters.

graph construction method based on manifold learning [15,16] and sparse representation [17]. A novel multi-view fusion approach can automatically weight each SIG matrix to generate the unified graph matrix. A rank constraint is imposed on the Laplacian matrix of the unified graph matrix, which helps partition the data points naturally into the required number of clusters. To the best of knowledge, this is the first attempt to propose a general graph-based system for multi-view clustering.

In summary, this paper makes the following contributions:

1. It proposes a general Graph-Based System (GBS) for multi-view clustering and discusses the impact of graph metrics on multi-view clustering within the proposed GBS.
2. It also proposes a novel multi-view clustering method on GBS to tackle the problems confronted in the existing methods. The proposed method can effectively construct SIG matrices, automatically weight each SIG to learn the unified graph and directly produce the final clusters on the unified graph.
3. Extensive experiments show the robust of the proposed system and the effectiveness of the proposed clustering method. It outperforms state-of-the-art baseline methods markedly.

The rest of this paper is organized as follows. Section 2 describes some notations and gives a brief review on related works. Section 3 introduces the proposed multi-view clustering method for GBS. Section 5 presents our optimization algorithm and provides a convergence proof of our optimization scheme. Experimental results are given in Section 6. Finally, we give concluding remarks and future work in Section 7.

2. Related work

In this section, we first describe some notational conventions, and then discuss the related works to our proposed method.

2.1. Notational conventions

Throughout the paper, matrices, and vectors and scalars are written in boldface capital letters (e.g., $\mathbf{X}$), boldface lowercase letters (e.g., $\mathbf{x}$) and lowercase letters (e.g., x), respectively. $\mathbf{I}$ denotes the identity matrix, and $\mathbf{1}$ denotes a column vector with all the entries as one. For a matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, the jth column vector and the ij-th entry are denoted by $\mathbf{x_j}$ and x_{ij}, respectively. The trace and the Frobenius norm of $\mathbf{X}$ are denoted by $Tr(\mathbf{X})$ and $\|\mathbf{X}\|_F$, respectively. For a vector $\mathbf{x} \in \mathbb{R}^{d\times 1}$, the jth entry is denoted by x_j, and l_p-norm is denoted by $\|\mathbf{x}\|_p$.

2.2. Graph-based clustering

Given a data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, where d is the dimensionality and n is the number of data points, traditional graph-based clustering methods partition the n data points into c clusters as follows:

Step 1. Constructing the data graph matrix $\mathbf{G} \in \mathbb{R}^{n\times n}$, where each entry g_{ij} in $\mathbf{G}$ represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$;

Step 2. Computing the graph Laplacian matrix $\mathbf{L}_G = \mathbf{D}_G - (\mathbf{G}^T + \mathbf{G})/2$, where $\mathbf{D}_G$ is a diagonal matrix whose ith diagonal element is $\sum_j (g_{ij} + g_{ji})/2$;

Step 3. Computing the embedding matrix $\mathbf{E} \in \mathbb{R}^{n\times c}$ by solving

$$\min_{\mathbf{E}\in\mathbb{R}^{n\times c}} Tr(\mathbf{E}^T\mathbf{L}_G\mathbf{E});$$

Step 4. Clustering $\mathbf{E}$ into c groups with an additional clustering algorithm (e.g., K-means).

If we further constrain $\mathbf{E}^T\mathbf{D}_G\mathbf{E} = \mathbf{I}$, it becomes the typical normalized cut [18]. Note that spectral clustering methods usually perform similar steps. Our method performs in a new paradigm, which produces clusters on the graph matrix (i.e., the unified graph matrix $\mathbf{U}$, which will be given in the next section), not the embedding matrix. The theorem in graph theory [19] motivates us to constrain the rank of $\mathbf{L}_U$ (the Laplacian matrix of $\mathbf{U}$) to be equal to $n - c$. In such a way, the connected components in the unified graph are exactly c, which are the final clusters.

2.3. Multi-view clustering

Based on the prior work, existing multi-view clustering methods can be classified into five categories: multi-view graph-based clustering, multi-view spectral clustering, co-training style clustering, multi-kernel clustering, and multi-view subspace clustering. Our work is clearly related to multi-view graph-based clustering [5,11,6–10,12–14]. [5] proposed a 3-stage graph-based multi-view clustering method that utilizes a graph representation of subspaces. However, it does not consider the weights of different views, and also requires a hierarchical agglomerative clustering algorithm to produce the final clustering results. To address the first issue, weighted multi-view graph-based clustering was studied in [6,7]. These two methods first construct each view graph matrix, combine the constructed graph matrices to learn a unified graph matrix, and then employ K-means on the unified graph matrix to produce the final clusters. That is, they both require an additional clustering step to produce the final results. More

advanced methods were presented in [8–10,6,12–14]. Although these methods generate the final clusters after fusion without additional clustering steps, none of them is a general framework or has studied the impact of different graph metrics on multi-view clustering. In addition, there are a number of parameters in [8,10,12,14], which are hard to set in practice. Proposals in this paper are made to handle these problems. In Section 6, we will compare with these methods experimentally.

Our work is also related to multi-view spectral clustering [20–27]. As discussed earlier, spectral clustering behaves similar to graph-based clustering. Multi-view spectral clustering methods typically find a low-dimensional embedding matrix of the data first and then employ an additional clustering algorithm on this embedding matrix to produce the final clusters. However, additional clustering steps may bring about additional PAC (Probably Approximately Correct) bounds [28]. Our method produces the final clusters on the graph matrix of the data without any additional clustering steps. We will also compare with representative methods of this category.

There are also some other multi-view clustering methods, [29–34,16,35–41] to name a few. More methods are surveyed in [4].

3. Clustering method for our graph-based system

This section presents the proposed multi-view clustering method on our system GBS. From Fig. 1, we can see that GBS consists of four components: *feature extraction*, *graph construction*, *graph fusion* and *data clustering*. The first component is not our main concern as we compare with existing methods on benchmark datasets. In practice, we can use Bag-of-Words, N-grams or Word2Vector for text data [42], HOG, SIFT or LBP for image data [43], and STFT or FFT for audio data [44]. This section introduces the last three components in detail. We also give some insights about our method.

3.1. Graph construction

For a multi-view dataset with m views, let $\mathbf{X}^1, \ldots, \mathbf{X}^m$ be the data matrices of the m views and $\mathbf{X}^v = \{\mathbf{x}_1^v, \ldots, \mathbf{x}_n^v\} \in \mathbb{R}^{d_v \times n}$ be the data matrix of the vth view, where d_v is the dimensionality of the vth view and n is the number of data points. The most commonly used methods for transforming the data matrix $\mathbf{X}^v$ to a graph are as follows:

1. Complete graph. For each data point x_i^v, it puts edges between x_i^v and all the other data points.
2. k-nearest graph. For each data point x_i^v, it puts edges between x_i^v and its k nearest neighbors.

Let $\mathbf{S}^v \in \mathbb{R}^{n \times n}$ denote the constructed graph, where each node corresponds a data point. If nodes i and j are connected by an edge, the similarity $\mathbf{s}_{ij}^v$ on the graph is usually defined as follows:

1. Binary (0–1) similarity: $\mathbf{s}_{ij}^v = 1$.
2. Cosine similarity: $\mathbf{s}_{ij}^v = \frac{(x_i^v)^T x_j^v}{\|x_i^v\| \|x_j^v\|}$.
3. Gaussian kernel similarity: $\mathbf{s}_{ij}^v = exp(\frac{-\|x_i^v - x_j^v\|}{2\sigma})$, where σ is a scaling parameter.

There are some limitations in these metrics. For example, binary similarity is the simplest method but the result is weak. Cosine similarity cannot consider the local geometric structure of the data. Gaussian kernel similarity is a distance based measure, which is sensitive to noise and outliers in the data. We will compare these similarity metrics on both complete graph and k-nearest graph in Section 6.

We now introduce a novel graph construction method based on manifold learning [15,16] and sparse representation [17]. An intuitive explanation of manifold learning is that if two data points are close, they are also close to each other in the embedding graph. Recent study [17] has shown that sparse representation is robust to noise and outliers. Specially, we compute SIG matrix of each view (e.g., v) by solving the following problem:

$$\begin{aligned} \min_{\mathbf{S}^v} & \sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \alpha \sum_{i=1}^{n} \|\mathbf{s}_i^v\|_1 \\ & s.t.\ s_{ii}^v = 0, s_{ij}^v \geq 0. \end{aligned} \tag{1}$$

When we normalize $\mathbf{s}_i^v$ with $\mathbf{1}^T \mathbf{s}_i^v = 1$, it exactly makes the second term constant. That is, the normalization $\mathbf{1}^T \mathbf{s}_i^v = 1$ is equivalent to the sparse constraint on $\mathbf{s}_i^v$. Then, problem (1) is turned into

$$\begin{aligned} \min_{\mathbf{S}^v} & \sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v \\ & s.t.\ s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T \mathbf{s}_i^v = 1. \end{aligned} \tag{2}$$

Here we denote that problem (2) has a trivial solution, i.e., only one data point with the smallest distance to $\mathbf{x}_i^v$ has the value 1, while all the other data points have the value 0. Now we add a prior to problem (2), which is formulated as

$$\begin{aligned} \min_{\mathbf{S}^v} & \sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta \sum_{i=1}^{n} \|\mathbf{s}_i^v\|_2^2 \\ & s.t.\ s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T \mathbf{s}_i^v = 1. \end{aligned} \tag{3}$$

The prior can be seen as the similarity value of each data point to $\mathbf{x}_i^v$, which is $\frac{1}{n}$, if we only focus on the second term of Eq. (3) (Note that we use the terms problem $(\cdot)$ and Eq. $(\cdot)$ interchangeably as each problem is modeled as an equation.). Now we can construct a SIG matrix for each view. In the next subsection, we present our proposed graph fusion method.

3.2. Graph fusion

This subsection presents the proposed fusion method, which can automatically weight each SIG matrix to find a unified graph matrix. Mathematically, we compute the unified matrix $\mathbf{U} \in \mathbb{R}^{n \times n}$ from the SIG matrices $\mathbf{S}^1, \ldots, \mathbf{S}^m$ by solving the problem below:

$$\begin{aligned} \min_{\mathbf{U}} & \sum_{v=1}^{m} w_v \|\mathbf{U} - \mathbf{S}^v\|_F^2 \\ & s.t.\ u_{ij} \geq 0, \mathbf{1}^T \mathbf{u}_i = 1 \end{aligned} \tag{4}$$

where w_v is the weight of the vth view. The weight w_v can be determined automatically according to the following theorem.

Theorem 1. *The weight w_v is determined as $\frac{1}{2\sqrt{\|\mathbf{U}-\mathbf{S}^v\|_F^2}}$.*

Proof. We now define an auxiliary function of $\mathbf{U}$ as follows:

$$\begin{aligned} \min_{\mathbf{U}} & \sum_{v=1}^{m} \sqrt{\|\mathbf{U} - \mathbf{S}^v\|_F^2} \\ & s.t.\ u_{ij} \geq 0, \mathbf{1}^T \mathbf{u}_i = 1. \end{aligned} \tag{5}$$

Taking the derivative of the Lagrange function of Eq. (5) with respect to $\mathbf{U}$ and setting the derivative to zero, we have

$$\sum_{v=1}^{m} \hat{w}_v \frac{\partial \|\mathbf{U} - \mathbf{S}^v\|_F^2}{\partial \mathbf{U}} + \frac{\partial \Theta(\Lambda, \mathbf{U})}{\partial \mathbf{U}} = 0 \tag{6}$$

where $\Theta(\Lambda, \mathbf{U})$ is the formalized term derived from the constraints $u_{ij} \geq 0$ and $\mathbf{1}^T\mathbf{u}_i = 1$, Λ is the Lagrange multiplier, and

$$\hat{w}_v = \frac{1}{2\sqrt{\|\mathbf{U} - \mathbf{S}^v\|_F^2}}. \tag{7}$$

If we also perform the same operations on Eq. (4), we can get the same result as shown in Eq. (6). Thus, the solution to weight w_v is $\frac{1}{2\sqrt{\|\mathbf{U}-\mathbf{S}^v\|_F^2}}$. □

Note that Eq. (6) is the derivative of the Lagrange function of Eq. (5) with respect to $\mathbf{U}$. Plug w_v using Eq. (7) into Eq. (6), then Eq. (6) equals to the derivative of Eq. (5) with respect to $\mathbf{U}$. As can be seen, Eqs. (4) and (5) are different and the main character of Theorem 1 is to determine the weights by solving Eq. (5).

3.3. Data clustering

As mentioned in the introduction section, our method can directly produce the clustering result on the unified graph matrix $\mathbf{U}$ without any additional clustering algorithms. So far, the unified graph matrix $\mathbf{U}$ obtained through Eq. (4) above cannot tackle this.

Below, we give an efficient solution to achieve this goal by imposing a rank constraint on the graph Laplacian matrix $\mathbf{L}_U$ of the unified matrix $\mathbf{U}$. According to graph theory [19,45], if graph matrix (our $\mathbf{U}$ in this case) is non-negative, then we have Theorem 2.

Theorem 2. *The graph Laplacian matrix $\mathbf{L} = \mathbf{L}_U$ of the graph matrix $\mathbf{U}$ has the following properties.*

1. *$\mathbf{L}$ is a symmetric positive semi-definite matrix. Thus all eigenvalues of $\mathbf{L}$ are real and non-negative, and $\mathbf{L}$ has a full set of n real and orthogonal eigenvectors.*
2. *$\mathbf{L1} = \mathbf{0}$, where $\mathbf{1} = [1, \ldots, 1]^T$. Thus 0 is an eigenvalue of $\mathbf{L}$ and $\mathbf{1}$ is the corresponding eigenvector.*
3. *If the graph $\mathbf{U}$ has r connected components then $\mathbf{L}$ has r eigenvalues that equal 0.*

The proof of each part in Theorem 2 is presented in [45]. As a conclusion, the part 3 in Theorem 2 says that if $rank(\mathbf{L}_U) = n - c$ as $c = r$, the corresponding $\mathbf{U}$ can be partitioned into c groups directly. Inspired by Theorem 2, we add a rank constraint $rank(\mathbf{L}_U) = n - c$ to problem (4).

We now detail the constraint $rank(\mathbf{L}_U) = n - c$. Let $\vartheta_i(\mathbf{L}_U)$ be the ith smallest eigenvalue of $\mathbf{L}_U$. We know that $\vartheta_i(\mathbf{L}_U) \geq 0$ because $\mathbf{L}_U$ is positive semi-definite (See part 1 in Theorem 2). Then, the constraint $rank(\mathbf{L}_U) = n - c$ can be achieved if $\sum_{i=1}^{c} \vartheta_i(\mathbf{L}_U) = 0$. According to Ky Fan's theorem [46], we have

$$\sum_{i=1}^{c} \vartheta_i(\mathbf{L}_U) = \min_{\mathbf{F}\in\mathbb{R}^{n\times c}, \mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}) \tag{8}$$

where $\mathbf{F} = \{\mathbf{f}_1, \ldots \mathbf{f}_c\}$ is the embedding matrix.

Plugging the item Eq. (8) into problem (4), formally, our objective function is formulated as

$$\begin{aligned} \min_{\mathbf{U}} & \sum_{v=1}^{m} w_v \|\mathbf{U} - \mathbf{S}^v\|_F^2 + 2\lambda Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}) \\ & s.t.\ s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T\mathbf{s}_i^v = 1, \\ & \quad u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1, \mathbf{F}^T\mathbf{F} = \mathbf{I}. \end{aligned} \tag{9}$$

When λ is large enough, the solution to problem (9) will make $\sum_{i=1}^{k} \vartheta_i(\mathbf{L}_U) = 0$ hold. In practice, we simply increase or decrease the value of λ when the number of connected components is smaller or greater than c. Thus, the resulting unified graph matrix $\mathbf{U}$ contains c connected components exactly, which partitions the data points into c clusters.

4. Some insights about problem (9)

Now we give some insights of our method, i.e., problem (9). Theorems 3 and 4 reveal that problem (9) can be seen as a combination of K-means and spectral clustering. The parameter λ adjusts the contributions of both methods.

Theorem 3. *When $\lambda = 0$ and $\mathbf{S}^v = (\mathbf{X}^v)^T\mathbf{X}^v$, problem (9) is equivalent to K-means clustering.*

Proof. Given the data matrix $\mathbf{X}^v = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d_v\times n}$ and the user-specified cluster number c, the objective function of K-means is

$$\min \sum_{j=1}^{c} \sum_{\mathbf{x}_i^v \in \mathbf{z}_j} \|\mathbf{x}_i^v - \mathbf{z}_j\|_2^2 \tag{10}$$

where $\mathbf{z}_j$ is the cluster center of the jth cluster.

Let $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_k] \in \mathbb{R}^{d_v\times c}$ and $\mathbf{Y} \in \mathbb{R}^{n\times c}$ be the clustering center matrix and the indicator matrix, respectively. For the indicator matrix $\mathbf{Y}$, each element is in [0, 1], each column vector contain at least one non-zero element and the sum of each row vector is 1. Then, Eq. (10) can be rewritten as

$$\begin{aligned} & \min_{\mathbf{Z},\mathbf{Y}} \left\|\mathbf{X}^v - \mathbf{Z}\mathbf{Y}^T\right\|_F^2 \\ \Leftrightarrow & \min_{\mathbf{Z},\mathbf{Y}} Tr(\mathbf{Y}\mathbf{Z}^T\mathbf{Z}\mathbf{Y}^T) - 2Tr((\mathbf{X}^v)^T\mathbf{Z}\mathbf{Y}^T). \end{aligned} \tag{11}$$

Since the solution to $\mathbf{Z}$ with respect to $\mathbf{X}^v$ and $\mathbf{Y}$ is $\mathbf{Z} = \mathbf{X}^v\mathbf{Y}^T(\mathbf{Y}\mathbf{Y}^T)^{-1}$, we have the equation $Tr(\mathbf{Y}\mathbf{Z}^T\mathbf{Z}\mathbf{Y}^T) = Tr((\mathbf{X}^v)^T\mathbf{Z}\mathbf{Y}^T)$. So, problem (11) becomes

$$\begin{aligned} & \max_{\mathbf{Z},\mathbf{B}} Tr((\mathbf{X}^v)^T\mathbf{Z}\mathbf{Y}^T) \\ \Leftrightarrow & \max_{\mathbf{Y}} Tr((\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T(\mathbf{X}^v)^T\mathbf{X}^v\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}) \\ \Leftrightarrow & \max_{\mathbf{R}} Tr(\mathbf{R}^T(\mathbf{X}^v)^T\mathbf{X}^v\mathbf{R}) \end{aligned} \tag{12}$$

where $\mathbf{R} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$. Note that $Tr(\mathbf{R}\mathbf{R}^T\mathbf{R}\mathbf{R}^T) = Tr(\mathbf{R}\mathbf{R}^T) = Tr(\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}) = c$. Then problem (12) can be transformed to the following problem

$$\min_{\mathbf{R}} \left\|\mathbf{R}\mathbf{R}^T - (\mathbf{X}^v)^T\mathbf{X}^v\right\|_F^2. \tag{13}$$

Define $\mathbf{U} = \mathbf{R}\mathbf{R}^T$ and $\mathbf{S}^v = (\mathbf{X}^v)^T\mathbf{X}^v$, Eq. (13) is rewritten as

$$\min_{\mathbf{U}} \|\mathbf{U} - \mathbf{S}^v\|_F^2 \tag{14}$$

where $\mathbf{U}$ is a symmetry matrix. So, K-means clustering is to seek a matrix $\mathbf{U}$ by minimizing problem (14), which is the first term in problem (9). □

Theorem 4. *When $\lambda \to \infty$, problem (9) is equivalent to spectral clustering.*

Proof. When $\lambda \to \infty$, problem (9) can be seen as the following problem

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times c}, \mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}). \tag{15}$$

As introduced in Section 2, spectral clustering also aims to solve the problem shown above. From this viewpoint, spectral clustering and our method are equivalent. □

Hereto, we presented our method in detail. In the next section, we propose our optimization algorithms to solve problem (3) and problem (9).

5. Optimization algorithms

5.1. Optimization algorithm for problem (3)

As can be seen, problem (3) is independent for each data point i, so we can solve the following problem separately for each data point i:

$$\min_{\mathbf{s}_i^v} \sum_{j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta\|\mathbf{s}_i^v\|_2^2 \quad s.t.\ s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T\mathbf{s}_i^v = 1. \tag{16}$$

We denote $d_{ij} = \|\mathbf{x}_i^v - \mathbf{x}_i^v\|_2^2$ and further denote $\mathbf{d}_i$ as a vector with the jth element as d_{ij}. Then we can formulate problem (17) in a vector form as follows:

$$\min_{\mathbf{s}_i^v} \left\|\mathbf{s}_i^v + \frac{\mathbf{d}_i}{2\beta}\right\|_2^2 \quad s.t.\ s_{ij} \geq 0, \mathbf{1}^T\mathbf{s}_i^v = 1. \tag{17}$$

In practice, we prefer a data point having similarities with its neighbors. That is, we aim to learn a $\mathbf{s}_i^v$ with k nonzero values, where k is the number of neighbors. This problem can be solved with a closed form solution as in [47]. Here we give a summary of this algorithm, which is shown in Algorithm 1.

Algorithm 1: Algorithm for computing $\mathbf{s}_i^v$ in problem (17)

Input : Data matrix $\mathbf{X}^v$, and the number of neighbors k.
Output: Vector $\mathbf{s}_i^v$ in the SIG matrix $\mathbf{S}^v$.

begin
- Take the Lagrangian function of problem (17): $\ell(\mathbf{s}_i^v, \eta, \xi) = \left\|\mathbf{s}_i^v + \frac{\mathbf{d}_i}{2\beta}\right\|_2^2 - \eta(\mathbf{1}^T\mathbf{s}_i^v - 1) - \xi^T\mathbf{s}_i^v$;
- According to the Karush–Kuhn–Tucker conditions, we have the optimal solution $\hat{s}_{ij}^v = (-\frac{d_{ij}}{2\alpha} + \eta)_+$, where $(a)_+ = max(a, 0)$;
- Order $d_{i1}, \ldots, d_{in}$ from small to large;
- Due to the constraint $\mathbf{1}^T\mathbf{s}_i^v = 1$, we have $\eta = \frac{1}{k} + \frac{1}{2k\beta}\sum_{j=1}^{k} h_{ij}$;
- As there are only k nonzero values in $\mathbf{s}_i^v$, we have the maximal β, denoted as $\beta = \frac{k}{2}d_{i,k+1} - \frac{1}{2}\sum j = 1^k d_{ij}$;
- Get the resulting $\mathbf{s}_i^v$ with jth entry as $s_{ij}^v = \begin{cases} \frac{d_{i,k+1} - d_{ij}}{k d_{i,k+1} - \sum_{h=1}^{k} b_{ih}} & j \leq k, \\ 0 & j > k; \end{cases}$

end

Result: The resulting $\mathbf{s}_i^v$ with k nonzero values.

5.2. Optimization algorithm for problem (9)

Suppose we have obtained $\mathbf{S}^1, \ldots, \mathbf{S}^m$ by using Algorithm 1. Now we compute the unified matrix $\mathbf{U}$ by solving problem (9). Solving problem (9) to give every variable an optimized solution at once is still challenging because all the variables are coupled. Assume $\mathbf{w}$ and $\mathbf{F}$ are fixed, we can compute $\mathbf{U}$ via the augmented Lagrange multiplier scheme, which has been shown effective in many matrix learning problems [48]. Similarly, $\mathbf{w}$ and $\mathbf{F}$ can be updated when the other variables are fixed. These inspire us to develop an alternating iterative algorithm to solve problem (9). The detailed updated rules are shown below:

Keep U and F fixed, update w: When $\mathbf{U}$ and $\mathbf{F}$ are fixed, the second term of problem (9) is a constant. Then we have problem (4). As discussed early, the value of w_v is updated by Eq. (7).

Keep w and F fixed, update U: When $\mathbf{w}$ and $\mathbf{F}$ are fixed, the optimization problem (9) is turned into

$$\min_{\mathbf{U}} \sum_{v=1}^{m}\sum_{i,j=1}^{n} w_v(u_{ij} - s_{ij}^v)^2 + 2\lambda Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}) \quad s.t.\ u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1. \tag{18}$$

Since $Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}) = \frac{1}{2}\sum_{i,j}\|\mathbf{f}_i - \mathbf{f}_j\|_2^2 u_{ij}$, problem (18) is rewritten as

$$\min_{\mathbf{U}} \sum_{v=1}^{m}\sum_{i,j=1}^{n} w_v(u_{ij} - s_{ij}^v)^2 + \lambda\sum_{i,j=1}^{n}\|\mathbf{f}_i - \mathbf{f}_j\|_2^2 u_{ij} \quad s.t.\ u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1. \tag{19}$$

Similar to d_{ij} and $\mathbf{d}_i$, we denote $h_{ij} = \|\mathbf{f}_i - \mathbf{f}_j\|_2^2$ and further denote $\mathbf{h}_i$ as a vector with the jth element as h_{ij}. Then we can formulate problem (19) in a vector form as follows

$$\min_{\mathbf{u}_i} \sum_{v=1}^{m}\left\|\mathbf{u}_i - \mathbf{s}_i^v + \frac{\lambda}{2mw_v}\mathbf{h}_i\right\|_2^2 \quad s.t.\ u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1. \tag{20}$$

For simplicity, the solution to problem (20) is presented in the next subsection, where we develop a simple and yet effective algorithm.

Keep w and U fixed, update F: With $\mathbf{w}$ and $\mathbf{U}$ fixed, optimizing $\mathbf{F}$ is to solve the following problem:

$$\min_{\mathbf{F}} Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}) \quad s.t.\ \mathbf{F}^T\mathbf{F} = \mathbf{I}. \tag{21}$$

The study in [19] indicates that the optimal solution to $\mathbf{F}$ is formed by the c eigenvectors of $\mathbf{L}_U$ corresponding to the c smallest eigenvalues.

Hereto, all the variables have been updated. The procedure of the proposed algorithm to solve problem (9) is summarized in Algorithm 2. We denote that Algorithm 2 can be fed with different types of SIG matrices. If Algorithm 2 is fed with the SIG matrices generated from Algorithm 1, it forms our proposed multi-view clustering method. We briefly name it GBS-KO (GBS is fed with k-nearest graph defined by our SIG matrices.). Compared to single view graph-based clustering algorithm, the proposed algorithm needs to optimize Problem (18). The computational complexity of optimizing Problem (18) is $O(tmn^2)$ in total. In practice, $t \ll n$ and $m \ll n$. Thus, the proposed algorithm does not increase the computational complexity of graph-based clustering, i.e., $O(n^3)$. For large-scale data, the study on large-scale spectral clustering such as [49,50] can be used to speed up our algorithm as our algorithm exploits the key characters of spectral clustering.

5.3. Solution to problem (20)

We now give the solution to problem (20). When $\mathbf{w}$ and $\mathbf{F}$ are fixed, the second and third terms in problem (20) are constants. Denoting $\mathbf{q}^v = \mathbf{s}_i^v - \frac{\lambda}{2mw_v}\mathbf{e}_i$, then problem (20) is simplified to

$$\min_{\mathbf{u}_i} \sum_{v=1}^{m}\|\mathbf{u}_i - \mathbf{q}^v\|_2^2 \quad s.t.\ u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1. \tag{22}$$

Algorithm 2: Algorithm for solving problem (9)

Input : SIG matrices with m views $\mathbf{S}^1, \ldots, \mathbf{S}^m$, the number of clusters c, and initial parameter λ.
Output: The learned unified matrix $\mathbf{U}$.

```
begin
    Initialize the weight for each view with w_v = 1/m;
    Initialize U by connecting S^1, ..., S^m with w, and then F is
    obtained by solving Eq. (21);
    while Property 2 or the maximum iteration reached do
        Keep F and U fixed, update w_v by using Eq. (7);
        Keep w and F fixed, update U by solving problem (20);
        Keep w and U fixed, update F, which is formed by the
        c eigenvectors of L_U corresponding to the c smallest
        eigenvalues;
    end
end
```

Result: The learned unified matrix $\mathbf{U}$ with exact c connected components, which are the final clusters.

Let φ and ϕ be the Lagrange multipliers for the constraints $u_{ij} \geq 0$ and $\mathbf{1}^T\mathbf{u}_i = 1$ respectively. The Lagrangian function of problem (22) is

$$\ell(\mathbf{u}_i, \phi, \varphi) = \frac{1}{2}\sum_{v=1}^{m}\|\mathbf{u}_i - \mathbf{q}_v\|_2^2 - \varphi^T\mathbf{u}_i - \phi(\mathbf{1}^T\mathbf{u}_i - 1) \tag{23}$$

where φ is a Lagrangian coefficient vector and ϕ is a scalar.

Suppose the optimal solution to problem (22) is $\mathbf{u}_i^*$, and the corresponding Lagrange multipliers are ϕ^* and φ^* respectively. According to the Karush–Kuhn–Tucker conditions, we have

$$\begin{cases} \forall j,\ \sum_{v=1}^{m} u_{ij}^* - \sum_{v=1}^{m} q_j^v - \varphi_j^* - \phi^* = 0 & \text{(a)} \\ \forall j,\ u_{ij}^* \geq 0 & \text{(b)} \\ \forall j,\ \varphi_j^* \geq 0 & \text{(c)} \\ \forall j,\ u_{ij}^*\varphi_j^* = 0 & \text{(d)} \end{cases} \tag{24}$$

Writing Eq. (24)(a) in a vector form, we get $\sum_{v=1}^{m}\mathbf{u}_i^* - \sum_{v=1}^{m}\mathbf{q}^v - \varphi^* - \phi^*\mathbf{1} = \mathbf{0}$. Due to the constraint $\mathbf{1}^T\mathbf{u}_i^* = 1$, we have $\phi^* = \frac{m - \sum_{v=1}^{m}\mathbf{1}^T\mathbf{q}^v - \mathbf{1}^T\varphi^*}{n}$.

Thus, the optimal solution $\mathbf{u}_i^*$ is formulated as

$$\mathbf{u}_i^* = \frac{\sum_{v=1}^{m}\mathbf{q}^v}{m} + \frac{\mathbf{1}}{n} - \frac{\sum_{v=1}^{m}\mathbf{1}^T q^v\mathbf{1}}{mn} - \frac{\mathbf{1}^T\varphi^*\mathbf{1}}{mn} + \frac{\varphi^*}{m}. \tag{25}$$

We further denote $\mathbf{p} = \frac{\sum_{v=1}^{m}q^v}{m} + \frac{\mathbf{1}}{n} - \frac{\sum_{v=1}^{m}\mathbf{1}^T q^v\mathbf{1}}{mn}$ and $\hat{\varphi}^* = \frac{\mathbf{1}^T\varphi^*}{mn}$. Then Eq. (25) is simplified to $\mathbf{u}_i^* = \mathbf{p} - \hat{\varphi}^*\mathbf{1} + \frac{\varphi^*}{m}$. As a result, for $\forall j$, we have

$$u_{ij}^* = p_j - \hat{\varphi}^* + \frac{\varphi_j^*}{m}. \tag{26}$$

According to Eqs. (24)(b)–(d) and (26), we know that $p_j - \hat{\varphi}^* + \frac{\varphi_j^*}{m} = (p_j - \hat{\varphi}^*)_+$. That is, the optimal solution u_{ij}^* can be obtained if $\hat{\varphi}^*$ is known, which is formulated as

$$u_{ij}^* = (p_j - \hat{\varphi}^*)_+. \tag{27}$$

In such a way, we get an adaptive neighbor graph because $(p_j - \hat{\varphi})_+ = max(p_j - \hat{\varphi}, 0)$. Furthermore, we can derive $\varphi_j^* = m(u_{ij}^* + \hat{\varphi}^* - p_j)$ from Eq. (26). Similarly, we then have $\varphi_j^* = m(\hat{\varphi}^* - p_j)_+$ according to Eqs. (24)(b)–(d). As denoted above $\hat{\varphi}^* = \frac{\mathbf{1}^T\varphi^*}{mn}$, the optimal solution $\hat{\varphi}^*$ is represented as $\hat{\varphi}^* = \frac{1}{n}\sum_{j=1}^{n}(\hat{\varphi}^* - p_j)_+$. Now we define a function of $\hat{\varphi}$ as

$$\mathcal{F}(\hat{\varphi}) = \frac{1}{n}\sum_{j=1}^{n}(\hat{\varphi} - p_j)_+ - \hat{\varphi}. \tag{28}$$

As can be seen, $\hat{\varphi}^*$ is determined by solving the root finding problem when $\mathcal{F}(\hat{\varphi}^*) = 0$. Since $\hat{\varphi} \geq 0$, $\mathcal{F}'(\hat{\varphi}_t) \leq 0$ and $\mathcal{F}'(\hat{\varphi}_t) \leq 0$ is a piece-wise linear and convex function, the root of $\mathcal{F}(\hat{\varphi}) = 0$ can be computed via the Newton method efficiently, shown below

$$\hat{\varphi}_{t+1} = \hat{\varphi}_t - \frac{\mathcal{F}(\hat{\varphi}_t)}{\mathcal{F}'(\hat{\varphi}_t)}. \tag{29}$$

5.4. Convergence proof

As mentioned above, we solve problem (17) with a closed form solution (i.e., Algorithm 1) and solve problem (9) with an alternating iterative algorithm (i.e., Algorithm 2). We now analyze the convergence of these two problems. First, it is easy to check that Eq./problem (17) is a convex function because the two-order derivative of this function with respect to $\mathbf{s}_i^v$ is a positive value. Thus, Eq. (17) decreases monotonically with the computing scheme of $\mathbf{s}_i^v$. Second, proof of the convergence of Algorithm 2 is equivalent to proving each sub-problem is convex. If we can find the optimal solution of each sub-problem, Algorithm 2 converges obviously. The convergence proof of each sub-problem is as follows.

Update $\mathbf{w}$. We have problem (4), which is a linear convex function. We also give a closed-form solution, i.e., Eq. (7), to each weight w_v.

Update $\mathbf{U}$. We have problem (18). We now prove the convergence of problem (18) based on Lemma 1 from [51].

Lemma 1. *For any non-zero matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times n}$, the following inequality holds:*

$$\|\mathbf{A}\|_F - \frac{\|\mathbf{A}\|_F^2}{2\|\mathbf{B}\|_F} \leq \|\mathbf{B}\|_F - \frac{\|\mathbf{B}\|_F^2}{2\|\mathbf{B}\|_F}. \tag{30}$$

Proof. Let $\Delta(\mathbf{U}) = 2\lambda Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F})$ and $\tilde{\mathbf{U}}$ be the resulting solution in each iteration, we can derive

$$\sum_{v=1}^{m}\frac{\|\tilde{\mathbf{U}} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F} + \Delta(\tilde{\mathbf{U}}) \leq \sum_{v=1}^{m}\frac{\|\mathbf{U} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F} + \Delta(\mathbf{U}). \tag{31}$$

According to Lemma 1, we have

$$\begin{aligned} &\sum_{v=1}^{m}\|\tilde{\mathbf{U}} - \mathbf{S}^v\|_F - \sum_{v=1}^{m}\frac{\|\mathbf{U} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F} \\ &\leq \sum_{v=1}^{m}\|\mathbf{U} - \mathbf{S}^v\|_F - \sum_{v=1}^{m}\frac{\|\mathbf{U} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F}. \end{aligned} \tag{32}$$

We now sum Eqs. (31) and (32) over both sides, which gives us

$$\sum_{v=1}^{m}\|\tilde{\mathbf{U}} - \mathbf{S}^v\|_F + \Delta(\tilde{\mathbf{U}}) \leq \sum_{v=1}^{m}\|\mathbf{U} - \mathbf{S}^v\|_F + \Delta(\tilde{\mathbf{U}}). \tag{33}$$

The inequality (33) says that the objective function of problem (18) decreases monotonically in each iteration until it converges. □

Update $\mathbf{F}$. We have problem (21). The Hessian matrix of Eq. (21) (i.e., problem (21)) is

$$\frac{\partial^2 Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F})}{\partial\mathbf{F}\partial\mathbf{F}^T} = \mathbf{L}_U + \mathbf{L}_U^T. \tag{34}$$

Since the Laplacian matrix $\mathbf{L}_U$ is positive semi-definite, the Hessian matrix of Eq. (21) is also positive semi-definite. Thus, Eq. (21) is a convex function with respect to $\mathbf{F}$, which is updated by $arg\ min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F})$ in each iteration.

6. Experiments

In this section, we evaluate our system GBS for multi-view clustering on both toy data and real-world data. We performed experiments on Windows Server 2008 R2 having Intel Xeon processor, 24 GB RAM and MATLAB development environment. We used MATLAB development environment because all the baselines used it. Before going further, we introduce the settings of our method. As mentioned earlier, if GBS is fed with the SIG matrices generated from Algorithm 1, we have the proposed multi-view clustering method, denoted by GBS-KO. Following [13], we set the number of neighbors $k = 15$. The parameter λ is set to 1 as an initial value, which is tuned automatically. In each iteration, we increase it ($\lambda = \lambda$*2) or decrease it ($\lambda = \lambda/2$) when the connected components of the learned graph is smaller or greater than the number of clusters c, respectively. After this, the experiments are divided into two parts.

6.1. Experiments on toy data

We first give a visual illustration of the capability of the proposed GBS-KO on a toy dataset.

Specially, we generate a two-view dataset, named *Two-Moon dataset*, as shown in Figs. 2a and 2d. Each view is generated with a moon pattern with 0.12 percentage of random Gaussian noise adding. There are two clusters, i.e., the upper moon (red) and the lower moon (blue). Figs. 2b, 2c, 2e and 2f show the experimental results.

Among these figures, Figs. 2b and 2e show the constructed graphs with the SIG matrices. From the figures, we can see that the two clusters are connected and not easy to separate on both views. Figs. 2c and 2f show the constructed graphs with the learned unified graph matrix. As can be seen, the two clusters are partitioned very well because GBS-KO can exploit the complementary information from the two views. That is, the hard-to-separate data points in one view can be easily separated with the help of another view.

6.2. Experiments on real-world data

Now we study the impact of different graph metrics on multi-view clustering and demonstrate the improvements of the proposed GBS-KO method for multi-view clustering. Experiments are conducted on eight real-world benchmark datasets. The clustering results are evaluated by comparing the obtained label of each instance with the provided label by the dataset. Two metrics, the accuracy (ACC) and the normalized mutual information (NMI), are used to measure the clustering performance [52,53]. In order to randomize the experiments, we run each algorithm 10 times and report the means and standard deviations of the performance measures.

Datasets. The statistics of these datasets are summarized in Table 1, where # instance, # view, and # cluster denote the number of instances, views, and clusters, respectively. # d_v denotes the dimensionality of the features in view v. Here we give a brief description of each dataset:

- *3 source dataset*[1] *(3source)* consists of 169 news reported by three online news organizations, i.e., BBC, Reuters, and The Guardian. Each news was manually annotated with one of six topical labels.

Table 1
Summary of the benchmark datasets.

Dataset	# instance	# view	# cluster	# d_1	# d_2	# d_3	# d_4	# d_5	# d_6
3sources	169	3	6	3560	3631	3068	–	–	–
BBC	685	4	5	4659	4633	4665	4684	–	–
BBCSport	544	2	5	3183	3203	–	–	–	–
NGs	500	3	5	2000	2000	2000	–	–	–
WebKB	203	3	4	1703	230	230	–	–	–
100leaves	1600	3	100	64	64	64	–	–	–
HW	2000	6	10	216	76	64	6	240	47
HW2sources	2000	2	10	784	256	–	–	–	–

- *BBC dataset*[2] *(BBC)* [54] consists of 685 documents collected from the BBC news website. Each document was split into four segments and was manually annotated with one of five topical labels.
- *BBCSport dataset see footnote 2 (BBCSport)* [54] consists of 544 documents collected from the BBC Sport website. Each document was split into two segments and was manually annotated with one of the five topical labels.
- *Newsgroups data set*[3] *(NGs)* [55] is a subset of the 20 Newsgroup datasets. It consists of 500 newsgroup documents. Each raw document was pre-processed with three different feature extraction methods (giving three views), and was annotated with one of five topical labels.
- *WebKB data set*[4] *(WebKB)* [56] consists of 203 web-pages of 4 classes. Each web-page is described by the content of the page, the anchor text of the hyper-link, and the text in its title.
- *One-hundred plant species leaves data set*[5] *(100leaves)* [57] is from the UCI repository. The dataset consists of 1600 samples with three views. Each sample is one of the one hundred plant species.
- *Handwritten digit data set*[6] *(HW)* [58] is from the UCI repository. The dataset consists of 2000 samples with six views. Each sample is one of the handwritten digits (0–9).
- *Handwritten digit 2 source dataset*[7] *(HW2sources)* consists of 2000 samples collected from two sources, i.e., MNIST Handwritten Digits (0–9) and USPS Handwritten Digits (0–9).

6.2.1. Comparisons of different graph metrics

To have a comparison of the impact of different graph metrics on multi-view clustering, GBS is fed with other five types of graphs below (Note that binary similarity cannot be used for complete graph because it gives 1 to all edges.):

(a) complete graph with Gaussian kernel similarity, denoted by GBS-CG;
(b) complete graph with cosine similarity, denoted by GBS-CC;
(c) k-nearest graph with binary similarity, denoted by GBS-KB;
(d) k-nearest graph with Gaussian kernel similarity, denoted by GBS-KG;
(e) k-nearest graph with cosine similarity, denoted by GBS-KC.

Fig. 3a shows the ACC scores of GBS with each graph metric on eight benchmark datasets. Fig. 3b shows the NMI scores of GBS with each graph metric on the eight benchmark datasets. From the figures, we can see that the clustering performances strongly depend on the graph metrics. Complete graph (dash line) is inferior to

1 http://mlg.ucd.ie/datasets/3sources.html.

2 http://mlg.ucd.ie/datasets/segment.html.

3 http://lig-membres.imag.fr/grimal/data.html.

4 https://linqs.soe.ucsc.edu/data.

5 https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set.

6 http://archive.ics.uci.edu/ml/datasets/Multiple+Features.

7 https://cs.nyu.edu/roweis/data.html.

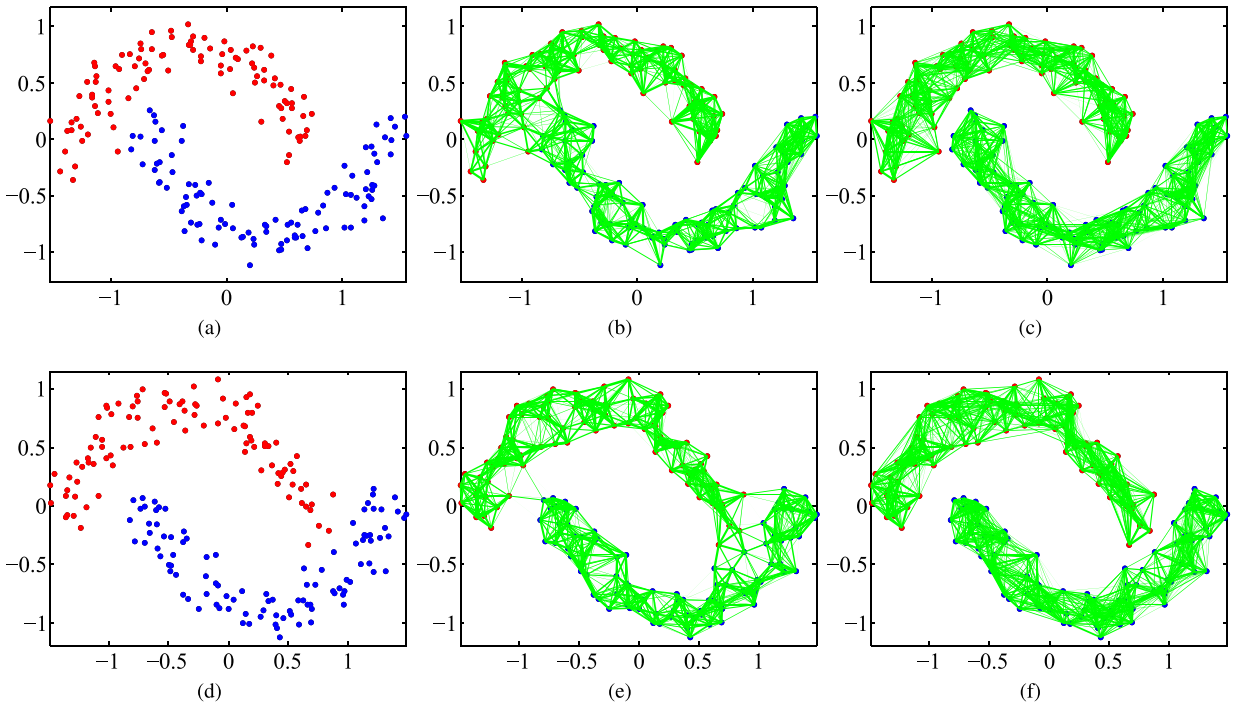

Fig. 2. A small example of GBS-KO on a toy two-view dataset. The upper row is the first view. The lower row is the second view. Notes: 2a and 2d are the generated sample data points of the first view and the second view, respectively; 2b and 2e are the constructed graphs with the SIG matrices for the first view and the second view, respectively; and 2c and 2f are the constructed graphs with the unified graph matrix for the first view and the second view, respectively . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

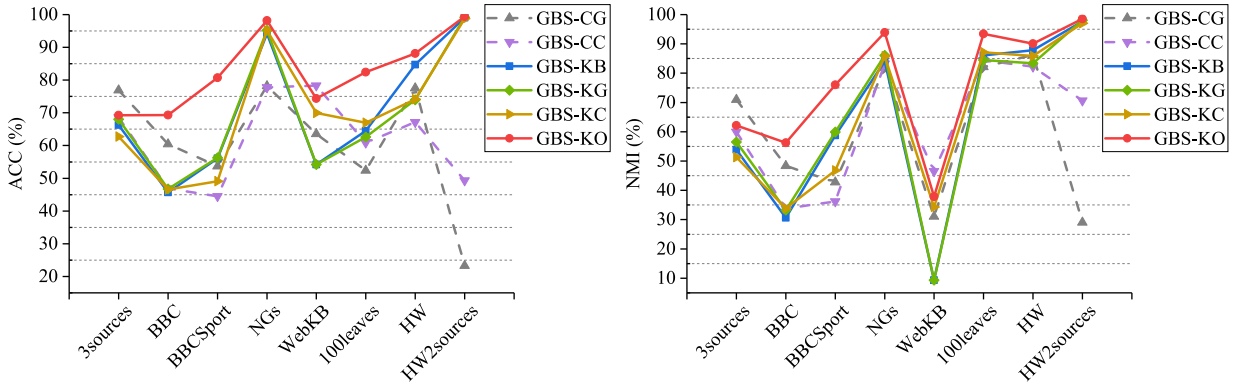

Fig. 3. Performance comparison of GBS with different graph metrics on eight real-world datasets.

k-nearest graph (solid line). For the methods with complete graph, the clustering results of both GBS-CG and GBS-CC are dependent on the datasets. For the methods with k-nearest graph, GBS-KC is slight better than GBS-KB and GBS-KG, but worse than our method GBS-KO. This suggests the importance of the geometrical structure and the sparse representation in learning the SIG matrices. As a conclusion, k-nearest graph with our method is a better choice for multi-view clustering.

6.2.2. Improvements of GBS-KO for multi-view clustering

To demonstrate how the clustering performance can be improved by our method, we compared GBS-KO with the following baseline methods, where the first two are single-view clustering methods and the others are multi-view clustering methods.

- K-means (Kmeans): This baseline conducts the K-means algorithm on each data view and reports the results of the view that gives the best performance. We use the code available in the tool-box of MATLAB. The maximum number of iterations is set to 200.
- Normalized cut (Ncut) [18]: Normalized cut is a typical graph-based method. We employ it for each view and report the results of the view that gives the best performance. The code is obtained from the author J. Shi.[8] The graph of each view is constructed by following the authors' suggestion to compute the similarity matrix as the squared $\mathbf{D}/\sigma$, where $\mathbf{D}$ is the pairwise Euclidean distance matrix and $\sigma = 0.05 * max(\mathbf{D})$.
- Co-regularized Spectral Clustering (CoregSC) [21]: This is a typical multi-view clustering method based on spectral clustering and kernel learning in a co-training style. We obtained the code from the author A. Kumar[9] and used the default parameter settings.
- Multi-view Spectral Clustering (MSC) [22]: This is a robust multi-view spectral clustering. We obtained the code from

8 www.cis.upenn.edu/~jshi.

9 http://legacydirs.umiacs.umd.edu/~abhishek/papers.html.

Table 2
Clustering performance comparison of each method in terms of ACC on eight real-world datasets.

ACC (%)	3sources	BBC	BBCSport	NGs	WebKB	100leaves	HW	HW2sources	Ave.
Kmeans	44.02 (4.27)	38.20 (6.21)	39.39 (3.81)	22.72 (1.52)	72.32 (5.12)	57.96 (1.41)	70.20 (8.31)	54.38 (3.95)	49.90 (4.33)
Ncut	40.24 (→0)	33.14 (→0)	35.85 (0.00)	23.60 (→0)	66.50 (0.00)	43.04 (1.29)	69.30 (→0)	38.80 (0.20)	43.26 (0.20)
CoregSC	54.79 (2.99)	47.01 (0.00)	43.44 (2.11)	27.68 (1.53)	59.70 (1.43)	77.06 (2.58)	75.56 (5.96)	81.50 (3.86)	58.34 (2.56)
MSC	47.51 (2.97)	62.32 (4.94)	35.39 (1.13)	31.12 (0.67)	47.34 (3.92)	73.79 (2.21)	79.18 (8.21)	68.78 (4.55)	55.68 (3.58)
MGL	67.51 (6.67)	53.96 (11.05)	53.90 (12.37)	82.18 (14.70)	73.84 (3.93)	69.04 (2.42)	74.40 (8.19)	68.93 (11.46)	67.97 (8.85)
MCGL	30.77 (→0)	35.33 (→0)	39.15 (0.00)	24.60 (0.00)	54.19 (0.00)	81.06 (→0)	85.30 (0.00)	97.95 (0.00)	56.04 (→0)
ASMV	33.73 (→0)	33.72 (0.00)	36.58 (0.00)	22.80 (→0)	72.41 (→0)	79.06 (→0)	57.45 (→0)	74.20 (0.00)	51.24 (→0)
GBS-KO	**69.23** (→0)	**69.34** (→0)	**80.70** (→0)	**98.20** (0.00)	**74.38** (0.00)	**82.44** (→0)	**88.10** (0.00)	**99.40** (0.00)	**82.72** (→0)

Table 3
Clustering performance comparison of each method in terms of NMI on eight real-world datasets.

NMI (%)	3sources	BBC	BBCSport	NGs	WebKB	100leaves	HW	HW2sources	Ave.
Kmeans	24.28 (4.39)	8.93 (9.96)	11.15 (6.09)	5.87 (3.63)	36.77 (6.96)	80.29 (0.66)	71.50 (3.66)	48.33 (1.90)	36.04 (4.65)
Ncut	10.45 (→0)	1.85 (→0)	1.76 (0.00)	7.39 (→0)	20.58 (→0)	74.86 (0.51)	83.02 (0.00)	36.43 (0.16)	28.98 (0.12)
CoregSC	52.38 (1.98)	28.63 (0.00)	22.80 (0.59)	8.80 (0.77)	31.39 (2.36)	91.65 (0.59)	74.21 (3.27)	70.63 (2.06)	47.56 (1.45)
MSC	38.50 (2.27)	55.31 (1.44)	10.10 (1.22)	9.72 (1.26)	22.37 (1.65)	90.14 (0.76)	75.60 (3.24)	62.77 (2.23)	45.56 (1.76)
MGL	57.68 (8.61)	36.97 (18.97)	41.14 (21.14)	83.04 (8.96)	**43.62** (1.43)	87.53 (0.76)	82.64 (4.73)	84.79 (5.36)	64.71 (8.75)
MCGL	10.34 (→0)	7.41 (→0)	8.72 (→0)	10.72 (→0)	8.60 (→0)	91.30 (0.00)	**90.55** (0.00)	95.55 (0.00)	40.33 (→0)
ASMV	8.96 (→0)	3.48 (0.00)	3.05 (→0)	6.30 (→0)	28.80 (→0)	90.09 (→0)	67.09 (→0)	87.38 (0.00)	36.89 (→0)
GBS-KO	**62.16** (0.00)	**56.27** (0.00)	**76.00** (0.00)	**93.92** (→0)	37.83 (0.00)	**93.43** (→0)	90.11 (→0)	**98.53** (0.00)	**76.03** (→0)

the author Y. Pan.[10] The parameter λ is set as 0.005 according to the authors' setting.

- Multiple Graph Learning (MGL) [6]: This is a parameter-free graph-based framework for multi-view clustering and semi-supervised classification. We use its clustering setting because here we handle clustering problem. The code is obtained from the author F. Nie.[11]
- Multi-view Clustering with Graph Learning (MCGL) [13]: This is a recently proposed multi-view graph-based clustering method. We obtained the code from the author K. Zhan.[12] The number of neighbors is set to the default value 10.
- Adaptive Structure-based Multi-view clustering (ASMV) [14]: This is a recent graph-based multi-view clustering approach. The code is provided by the author K. Zhan.[13] According to the authors' settings, the number of neighbors is set to 15, the scale factor *LA* is set to 10, the projection directions m is set to 50, and the parameters θ and γ are set to the default values.

Results. The performance comparison results are shown in Tables 2 and 3, where we also record the average score (denoted by Ave.) over eight datasets for each method. Note that $\rightarrow$ 0 means that the value is close to zero, and 0.00 denotes zero. The numbers in the parentheses are the standard deviations. From the tables, we make the following observations:

- Our proposed GBS-KO method is markedly better than all baselines. GBS-KO gives the best performance on all the datasets, except the WebKB dataset in terms of NMI and the HW dataset in terms of NMI. The results clearly show that our GBS-KO method is a promising multi-view clustering method.
- All the graph-based methods, i.e, Ncut, MGL, MCGL, ASMV and GBS-KO, perform robustly except MGL. For MGL, it employs K-means on the learned unified representation to produce the final clusters. Since K-means is sensitive to the initial cluster centers, it results in a high standard deviation for MGL.
- Compared with the recently proposed graph-based methods MCGL and ASMV, GBS-KO performs better or comparably. This suggests that by considering the geometrical structure of each view, the sparse representation, and the auto-weighting can achieve a better clustering result.
- In some cases, single view baseline methods, i.e., SK-means and SNcut, are even slightly better than some multi-view baseline methods. This indicates that exploring multi-view data still needs good techniques.

6.2.3. Comparison among BestView, FeatConcat and GBS-KO

We observed that single view baseline methods perform slightly better than some multi-view baseline methods just now. To further evaluate the clustering performance of our method, we conduct GBS-KO on each view data of each multi-view dataset. We record the results on each view data and report the results of the view that gives the best performance, denoted by BestView. We also merge the features of all views into a single view. Then this merged single view data are fed into GBS-KO, denoted by FeatConcat. Fig. 4 shows the comparison results among them on all benchmark datasets. From the figure, we can clearly see that GBS-KO always results in the best performance on all datasets except for the dataset WebKB. The reason why the results on the dataset WebKB are similar may be that the clustering ability of each view of this dataset is similar, which leads to the results of these three methods are also similar. We also observe that BestView is slightly superior to FeatConcat in most cases. This shows that simply concatenating the features of all views cannot achieve a better performance. Weighting technique is needed as we did in this paper.

10 http://ss.sysu.edu.cn/~py/.
11 https://www.escience.cn/people/fpnie.
12 https://github.com/kunzhan/MVCF.
13 https://github.com/kunzhan/MVGL.

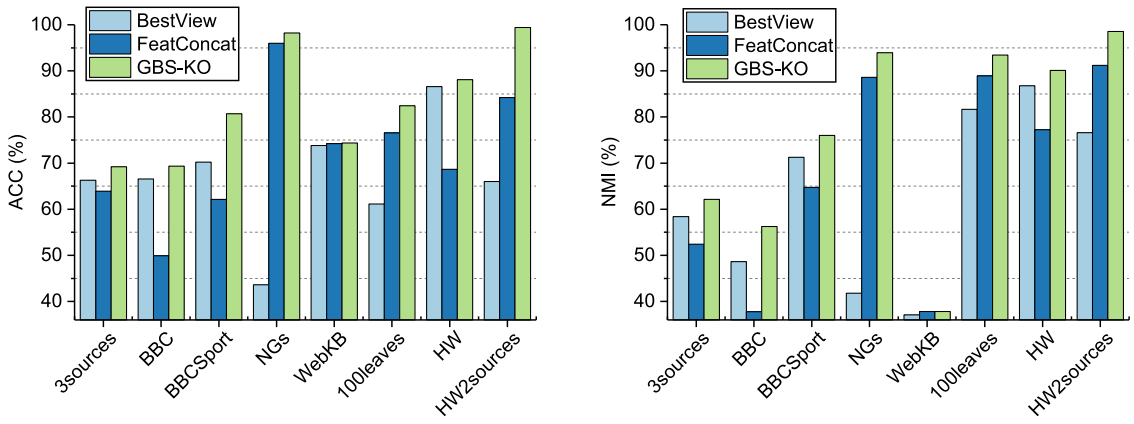

Fig. 4. Performance comparison among BestView, FeatConcat and GBS-KO on eight real-world datasets.

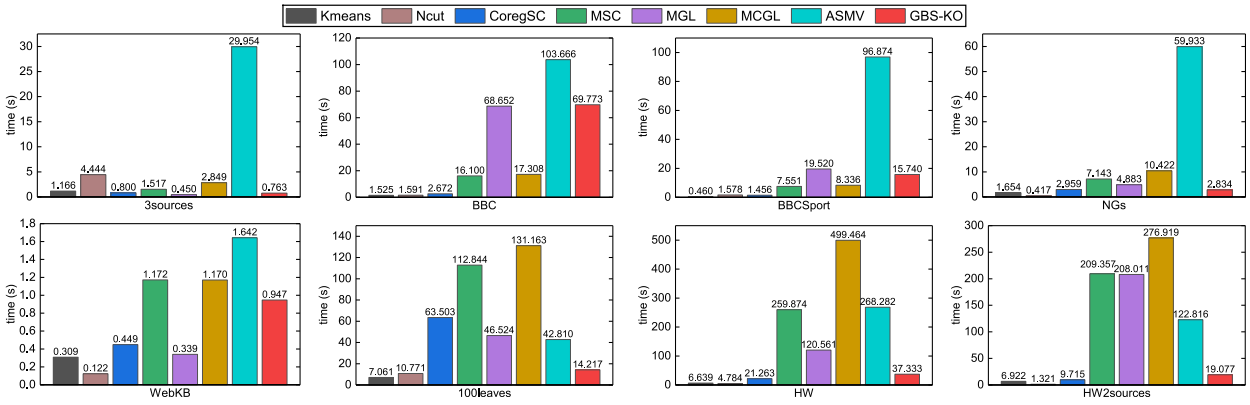

Fig. 5. Performance comparison of each method in terms of running time on eight real-world datasets.

6.3. Running time

Since the computational complexity of the proposed algorithm has been analyzed in Section 5.2, here we further evaluate running time of the proposed GBS-KO in practice. The results are shown in Fig. 5. From the figure, we can see that single view baselines (i.e., Kmeans and NCut) often perform more efficient than multi-view baselines (i.e., CoregSC, MSC, MGL, MCGL and ASMV) including the proposed method GBS-KO. The reason is clear because multi-view clustering methods need to handle multi-view data. For all multi-view clustering methods, our method GBS-KO achieves the best performance in most cases. More precisely, GBS-KO is superior to ASMV on all eight datasets, and superior to MSC, MGL and MCGL on six datasets of all eight datasets. GBS-KO is inferior to CoregSC on five datasets of all datasets. All the above results show that the running time of our method GBS-KO is medium among all the methods.

7. Conclusions

This paper presented a general system for multi-view clustering, called Graph-Based System (GBS). We discussed the impact of different graph metrics on GBS and introduced a new graph construction method based on manifold learning and sparse representation. As a conclusion, k-nearest graph with our method is a better choice for multi-view clustering. A novel multi-view clustering method based on GBS is proposed to tackle the limitations of the existing graph-based multi-view clustering methods. The proposed method can automatically weight the constructed graph of each view to learn a unified graph, which gives the final clusters without any additional clustering steps. Extensive experiments on both toy data and real-world data showed the superior performance of the proposed method.

Several questions remain to be investigated in our future work:

1. The proposed method constructs the graph of each view in isolation and keeps the constructed graph fixed during fusion. Instead of keeping them fixed, improving the constructed graph with the help of the learned unified graph during fusion may be more promising for multi-view clustering. This suggests a way to extend the proposed method.
2. This paper concerns unsupervised multi-view clustering. In real-world problems, data are partially labeled, the unlabeled ones can be labeled according to learning the pairwise constraints (i.e., must-link and cannot-link). So, our another future work is to learn the pairwise constraints for semi-supervised multi-view clustering or classification, such as [12,59].
3. Also, it would be very interesting to study incomplete multi-view clustering based on graph theory. The incomplete multi-view clustering here means that parts of instances are not available in some views [60–62]. It is unclear how to model graph for incomplete multi-view clustering.

Acknowledgments

This work was supported by the National Science Foundation of China (No. 61572407), and the China Scholarship Council (No. 201707000064). We would like to thank the authors of the baseline systems for their codes.

References

[1] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, 2013, CoRR abs/1304.5634.
[2] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: Recent progress and new challenges, Inf. Fusion 38 (2017) 43–54.
[3] G. Chao, S. Sun, J. Bi, A survey on multi-view clustering, 2017, CoRR abs/1712.06246.
[4] Y. Yang, H. Wang, Multi-view clustering: A survey, Big Data Min. Anal. 1 (2) (2018) 83–107.
[5] M. Saha, A graph based approach to multiview clustering, in: Proceedings of the International Conference on Pattern Recognition and Machine Intelligence, 2013, pp. 128–133.
[6] F. Nie, J. Li, X. Li, Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 1881–1887.
[7] C. Hou, F. Nie, H. Tao, D. Yi, Multi-view unsupervised feature selection with adaptive similarity and view weight, IEEE Trans. Knowl. Data Eng. 29 (9) (2017) 1998–2011.
[8] F. Nie, J. Li, X. Li, Self-weighted multiview clustering with multiple graphs, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2017, pp. 2564–2570.
[9] H. Tao, C. Hou, J. Zhu, D. Yi, Multi-view Clustering with adaptively learned graph, in: Proceedings of the Asian Conference on Machine Learning, 2017, pp. 113–128.
[10] W. Zhuge, F. Nie, C. Hou, D. Yi, Unsupervised single and multiple views feature extraction with structured graph, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2347–2359.
[11] F. Nie, G. Cai, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2408–2414.
[12] F. Nie, G. Cai, J. Li, X. Li, Auto-weighted multi-view learning for image clustering and semi-supervised classification, IEEE Trans. Image Process. 27 (3) (2018) 1501–1511.
[13] K. Zhan, C. Zhang, J. Guan, J. Wang, Graph learning for multiview clustering, IEEE Trans. Cybern. 48 (10) (2018) 2887–2895.
[14] K. Zhan, X. Chang, J. Guan, L. Chen, Z. Ma, Y. Yang, Adaptive structure discovery for multimedia analysis using multiple features, IEEE Trans. Cybern. PP (99) (2018) 1–9.
[15] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.
[16] H. Wang, Y. Yang, T. Li, Multi-view clustering via concept factorization with local manifold regularization, in: Proceedings of the IEEE International Conference on Data Mining, 2016, pp. 1245–1250.
[17] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[18] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[19] B. Mohar, Y. Alavi, G. Chartrand, O. Oellermann, The Laplacian spectrum of graphs, Graph Theory Combin. Appl. 2 (12) (1991) 871–898.
[20] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. B 40 (6) (2010) 1438–1446.
[21] A. Kumar, P. Rai, H.D. III, Co-regularized multi-view spectral clustering, in: Proceedings of the Annual Conference on Neural Information Processing Systems, 2011, pp. 1413–1421.
[22] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: Proceedings of the AAAI International Conference on Artificial Intelligence, 2014, pp. 2149–2155.
[23] C. Lu, S. Yan, Z. Lin, Convex sparse spectral clustering: Single-view to multi-view, IEEE Trans. Image Process. 25 (6) (2016) 2833–2843.
[24] L. Feng, L. Cai, Y. Liu, S. Liu, Multi-view spectral clustering via robust local subspace learning, Soft Comput. 21 (8) (2017) 1937–1948.
[25] J.W. Son, J. Jeon, A. Lee, S. Kim, Spectral clustering with brainstorming process for multi-view data, in: Proceedings of the AAAI International Conference on Artificial Intelligence, 2017, pp. 2548–2554.
[26] Y. Wang, L. Wu, X. Lin, J. Gao, Multiview spectral clustering via structured low-rank matrix factorization, IEEE Trans. Neural Netw. Learn. Syst. 29 (10) (2018) 4833–4843,.
[27] L. Zong, X. Zhang, X. Liu, H. Yu, Weighted multi-view spectral clustering based on spectral perturbation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, pp. 4621–4628.
[28] S. Sun, J. Shawe-Taylor, L. Mao, PAC-Bayes analysis of multi-view learning, Inf. Fusion 35 (2017) 117–131.
[29] E. Eaton, M. Desjardins, S. Jacob, Multi-view constrained clustering with an incomplete mapping between views, Knowl. Inf. Syst. 38 (1) (2014) 231–257.
[30] Y. Lu, L. Wang, J. Lu, J. Yang, C. Shen, Multiple kernel clustering based on centered kernel alignment, Pattern Recognit. 47 (11) (2014) 3656–3664.
[31] X. Zhang, L. Zong, X. Liu, H. Yu, Constrained NMF-based multi-View clustering on unmapped data, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 3174–3180.
[32] X. Zhang, X. Zhang, H. Liu, X. Liu, Multi-task multi-view clustering, IEEE Trans. Knowl. Data Eng. 18 (12) (2016) 3324–3338.
[33] L. Zong, X. Zhang, H. Yu, Q. Zhao, F. Ding, Local linear neighbor reconstruction for multi-view data, Pattern Recognit. Lett. 84 (2016) 56–62.
[34] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, E. Zhu, Multiple kernel K-means with incomplete kernels, in: Proceedings of the AAAI International Conference on Artificial Intelligence, 2017, pp. 2259–2265.
[35] L. Zong, X. Zhang, L. Zhao, H. Yu, Q. Zhao, Multi-view clustering via multi-manifold regularized non-negative matrix factorization, Neural Netw. 88 (2017) 74–89.
[36] C. Zhang, Q. Hu, H. Fu, P. Zhu, X. Cao, Latent multi-view subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2017, pp. 4333–4341.
[37] S. Wang, E.K. Wang, X. Li, Y. Ye, R.Y.K. Lau, X. Du, Multi-view learning via multiple graph regularized generative model, Knowl.-Based Syst. 121 (2017) 153–162.
[38] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in: Proceedings of the AAAI International Conference on Artificial Intelligence, 2017, pp. 2921–2927.
[39] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, Knowl.-Based Syst. (2018) 1–8.
[40] S. Huang, Y. Ren, Z. Xu, Robust multi-view data clustering with multi-view capped-norm k-means, Neurocomputing (2018) 197–208.
[41] F. Nie, L. Tian, X. Li, Multiview clustering via adaptively weighted procrustes, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 2022–2030.
[42] B. Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer Science & Business Media, 2007.
[43] D. Tian, A review on image feature extraction and representation techniques, Int. J. Multimedia Ubiquitous Eng. 8 (4) (2013) 385–396.
[44] O. Lartillot, P. Toiviainen, A matlab toolbox for musical feature extraction from audio, in: Proceedings of the International Conference on Digital Audio Effects, 2007, pp. 237–244.
[45] I.S. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 269–274.
[46] K. Fan, On a theorem of Weyl concerning eigenvalues of linear transformations I, Proc. Natl. Acad. Sci. USA 35 (11) (1949) 652–655.
[47] F. Nie, X. Wang, M.I. Jordan, H. Huang, The constrained Laplacian rank algorithm for graph-based clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016, pp. 1969–1976.
[48] D.P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, Academic Press, 2014.
[49] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph, in: Proceedings of the AAAI International Conference on Artificial Intelligence, 2015, pp. 2750–2756.
[50] X. Zhang, L. Zong, Q. You, X. Yong, Sampling for nyström extension-based spectral clustering: Incremental perspective and novel analysis, ACM Trans. Knowl. Discovery Data 11 (1) (2016) 7:1–7:25.
[51] F. Nie, H. Huang, X. Cai, C.H.Q. Ding, Efficient and robust feature selection via joint l2,1-norms minimization, in: Proceedings of the Annual Conference on Neural Information Processing Systems, 2010, pp. 1813–1821.
[52] D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, IEEE Trans. Knowl. Data Eng. 17 (12) (2005) 1624–1637.
[53] J. Hu, T. Li, C. Luo, H. Fujita, Y. Yang, Incremental fuzzy cluster ensemble learning based on rough set theory, Knowl.-Based Syst. 132 (2017) 144–155.
[54] D. Greene, P. Cunningham, Practical solutions to the problem of diagonal dominance in kernel document clustering, in: Proceedings of the International Conference on Machine Learning, 2006, pp. 377–384.
[55] S.F. Hussain, G. Bisson, C. Grimal, An improved co-similarity measure for document clustering, in: Proceedings of the International Conference on Machine Learning and Applications, 2010, pp. 190–197.
[56] Q. Lu, L. Getoor, Link-based classification, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 496–503.
[57] C. Mallah, J. Cope, J. Orwell, Plant leaf classification using probabilistic integration of shape, texture and margin features, in: Proceedings of the IASTED International Conference Signal Processing, Pattern Recognition and Applications, 2013, pp. 279–286.
[58] M. van Breukelen, R.P.W. Duin, D.M.J. Tax, J.E. den Hartog, Handwritten digit recognition by combined classifiers, Kybernetika 34 (4) (1998) 381–386.
[59] Q. Qian, S. Chen, X. Zhou, Multi-view classification with cross-view must-link and cannot-link side information, Knowl.-Based Syst. 54 (2013) 137–146.
[60] C. Xu, D. Tao, C. Xu, Multi-view learning with incomplete views, IEEE Trans. Image Process. 24 (12) (2015) 5812–5825.
[61] Q. Yin, S. Wu, L. Wang, Unified subspace learning for incomplete and unlabeled multi-view data, Pattern Recognit. 67 (2017) 313–327.
[62] J. Liu, Y. Jiang, Z. Li, Z.H. Zhou, Partially shared latent factor learning with multiview data, IEEE Trans. Neural Netw. Learn. Syst. 26 (6) (2014) 1233–1246.