GMC: Graph-based Multi-view Clustering

Hao Wang, Yan Yang, *Member, IEEE,* and Bing Liu, *Fellow, IEEE*

Abstract—Multi-view graph-based clustering aims to provide clustering solutions to multi-view data. However, most existing methods do not give sufficient consideration to weights of different views and require an additional clustering step to produce the final clusters. They also usually optimize their objectives based on fixed graph similarity matrices of all views. In this paper, we propose a general *G*raph-based *M*ulti-view *C*lustering (GMC) to tackle these problems. GMC takes the data graph matrices of all views and fuses them to generate a unified graph matrix. The unified graph matrix in turn improves the data graph matrix of each view, and also gives the final clusters directly. The key novelty of GMC is its learning method, which can help the learning of each view graph matrix and the learning of the unified graph matrix in a mutual reinforcement manner. A novel multi-view fusion technique can automatically weight each data graph matrix to derive the unified graph matrix. A rank constraint without introducing a tuning parameter is also imposed on the graph Laplacian matrix of the unified matrix, which helps partition the data points naturally into the required number of clusters. An alternating iterative optimization algorithm is presented to optimize the objective function. Experimental results using both toy data and real-world data demonstrate that the proposed method outperforms state-of-the-art baselines markedly.

Index Terms—Multi-view clustering, graph-based clustering, data fusion, Laplacian matrix, rank constraint.

1 Introduction

The current dominant paradigm for machine learning is to run an algorithm on the data represented in a single view. We call this paradigm *single-view learning*, because it does not consider any other related information from other views. This is in contrast to our human learning. We humans often look at problems from different views. That is why we can approach a problem holistically and comprehensively. In many real-life problems multi-view data arise naturally. For instance, the same news may be reported by different news organizations, an image may be encoded by different types of features, and a picture shared on websites may have different textual descriptions. All these are referred to as multi-view data, where each individual view constitutes a learning task but each view also has its biases.

The natural and frequent occurrence of multi-view data bred a new learning paradigm, called *multi-view learning*. Existing studies on this new paradigm has been surveyed in [1], [2]. In this paper, we focus on multi-view unsupervised learning and particularly, *multi-view clustering*. Multi-view clustering explores and exploits the complementary information from multiple views to produce a more accurate and robust partitioning of the data than single-view clustering [3], [4]. We will discuss related work in the next section. Among these multi-view clustering methods, one representative category of methods is the graph-based methods [5], [6], [7], [8], [9], [10], [11], [12]. Graph is an important data structure for representing the relationships among various types of objects. Each node in a graph corresponds to an object and each edge represents a relationship between two objects. Broadly speaking, each object in the real world has a variety of relationship graphs as each object can be sampled in different views and the sampled data of each view can form a graph. For example, an author in different bibliographic databases (e.g., DBLP and IEEE) may have different relationship graphs according to his/her papers. A user in Facebook or Twitter can form multiple social networks/graphs according to his/her profile database and social connections. A web page has its outbound link graph, inbound link graph and citation link graph. Clustering is a fundamental topic of data mining, especially when there are no labels for data objects. Clustering results are often used in subsequent applications, such as community detection, recommendation, and information retrieval.

Multi-view graph-based clustering methods typically find a fusion graph across the input graphs of all views first, and then employ an additional clustering algorithm on this fusion graph in order to produce the final clusters. In these methods, the input graph of each view is usually generated from a data similarity matrix with each matrix entry representing the similarity of two data points. Although such methods have achieved state-of-the-art performances, they still have several limitations. **First**, differences in the importance of different views are not considered in some methods, e.g., [5]. Our method handles the differences through automatically generated weights. **Second**, many existing methods require an additional clustering step to produce the final clusters after fusion, e.g., [5], [6], [7]. Our model produces clusters in fusion directly with no additional clustering step. **Third**, most current methods construct the graph of each view in isolation and keep the constructed graph fixed during fusion, e.g., [5], [6], [7], [9], [10], [12]. Our method jointly constructs each view graph and the fusion graph. Thus, the two construction processes help each other naturally. To the best of our knowledge, no existing work can address all these three limitations simultaneously. In this work, we address these limitations simultaneously and formulate our solutions using a joint

- *H. Wang and Y. Yang are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China. E-mail: cshaowang@gmail.com; yyang@swjtu.edu.cn.*
- *B. Liu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA. E-mail: liub@uic.edu.*
- *This work was done when the first author was a visiting student researcher at the University of Illinois at Chicago.*

Manuscript received April 19, 2005; revised August 26, 2015.

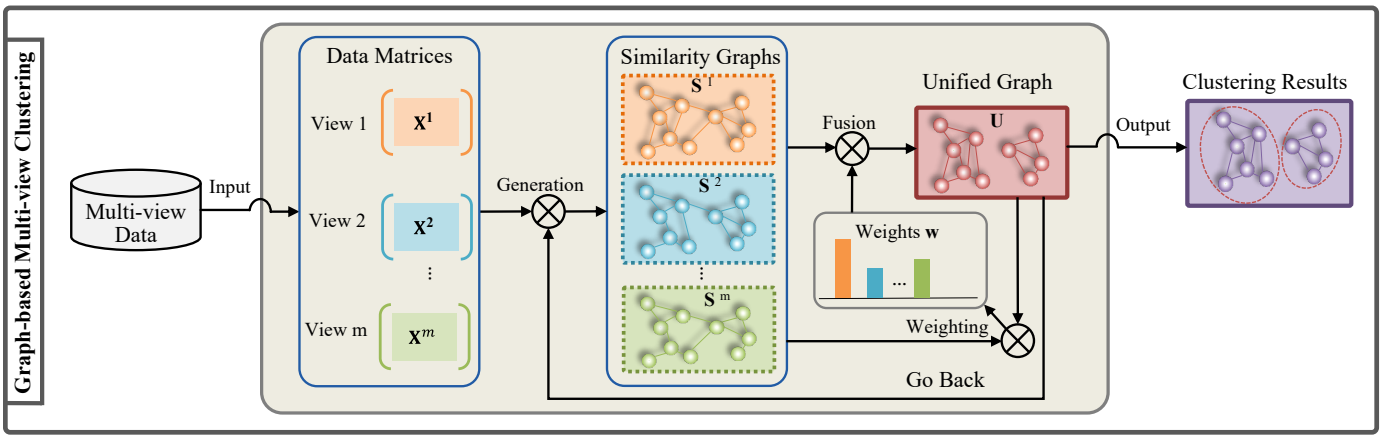


Fig. 1. The flow chart of the proposed Graph-based Multi-view Clustering (GMC).

framework for the first time.

Why do we need to address these three limitations? The reasons are as follows. First, sample selection bias [13] leads to view diversity. Second, additional clustering steps bring about additional PAC (Probably Approximately Correct) bounds [14]. Third, different similarity metrics have impact on multi-view clustering quality [15].

In this paper, we propose a novel multi-view clustering model, denoted by *Graph-based Multi-view Clustering* (GMC). GMC not only can weight each view automatically and produce the final clusters directly after fusion without any additional clustering steps to perform, but also can construct the graph of each view and the fusion graph jointly so that they can help each other in a mutual reinforcement manner. The overall flow of our GMC is shown in Fig. 1. Specifically, the data matrix of each view is first converted to a graph matrix generated from a similarity graph matrix. We call this graph matrix the *similarity-induced graph* (SIG) matrix. The proposed fusion method is then applied to the SIG matrices of all views in order to learn a unified matrix (i.e., fusion graph matrix) $\mathbf{U}$ from the SIG matrices. The learning of $\mathbf{U}$ automatically considers different weights (w_v) of different views (v). Meanwhile, the learned unified matrix $\mathbf{U}$ goes back to improve the SIG matrix of each view. A rank constraint on the Laplacian matrix $\mathbf{L}_U$ of the unified matrix is also imposed to constrain that the number of connected components in the unified matrix is equal to the required number of clusters c. Thus, our model GMC weights and improves the SIG matrix of each view, and generates the unified matrix and the final clusters simultaneously.

In summary, this paper has the following contributions:

1) *Motivation*: It studies an advanced multiple view clustering paradigm and provides a new clustering solution for multi-view data.
2) *Model*: It proposes a general Graph-based Multi-view Clustering (GMC) approach to addressing the aforementioned limitations of the current methods. GMC weights each view automatically, learns the graph of each view and the fusion graph jointly, and produces the final clusters directly after fusion. Remarkably, the learning of each view graph and the learning of the fusion graph can help each other.
3) *Algorithm*: It proposes an alternating iterative optimization algorithm to solve the GMC problem, wherein each sub-problem has an optimal solution.
4) *Results*: Experimental results show that the proposed GMC approach makes considerable improvement over the state-of-the-art baseline methods.

The rest of this paper is organized as follows. Section 2 introduces the related works. Section 3 presents the proposed graph-based multi-view clustering model. The optimization algorithm including its computational complexity analysis and convergence proof are discussed in Section 4. Extensive experiments are conducted in Section 5. Finally, Section 6 concludes this paper.

2 Related Work

Most related works to ours are those in [5], [6], [7], [8], [9], [10], [11], [12], which are existing multi-view graph-based clustering methods. However, existing graph-based approaches cannot handle the above-mentioned limitations simultaneously. For example, a 3-stage graph-based approach to multi-view clustering was proposed in [5] that utilizes a graph representation of subspaces and a hierarchical agglomerative clustering method. It does not consider the weights of different views. Toward this end, weighted multi-view graph-based clustering was studied in [6], [7]. These two approaches first generate a graph for each view and then weight each graph to build a unified representation for K-means to produce the final clusters. More advanced weighted methods were presented in [8], [9], [10], [11], [12]. Although these methods generate the final clusters with no additional clustering algorithms, they construct the graph of each view in isolation and keep the constructed graph fixed during fusion except [8] and [11], which only learn a global graph for all views (without building a graph for each view). Our proposed method can address these limitations. We will also compare with these methods experimentally. Beyond pair-wise similarity matrix fusion as used in the above methods, the high-order similarity matrix (namely the data-cluster similarity matrix) fusion via cross-view graph random walk was presented in [16], [17], in which the data-cluster similarity is the similarity between the data point and the center of cluster. Although such methods can avoid high computational complexity in pair-wise similarity matrix, they need to run an additional clustering algorithm.

Our work is also related to multi-view spectral clustering [18], [19], [20], [21], [22], [23], [24], [25]. Spectral clustering operates on a graph constructed from the data with data points as nodes and edges between them as similarities [21]. That is, the input to spectral clustering is also a similarity graph. The difference from graph-based clustering is that spectral clustering typically finds a low-dimensional embedding representation of the data first, and then performs *K*-means on this embedding representation to produce the final clusters. In this way, multi-view spectral clustering also needs an additional clustering step on the embedding representation. Graph-based clustering produces clusters on the constructed graph of the data, not a new embedding representation, although most of them still require additional clustering steps. Our method obtains the clustering indicators directly from the learned graph of the data.

In addition to multi-view graph-based clustering and multi-view spectral clustering, there are also some other categories of multi-view clustering methods. The other related multi-view clustering methods can be roughly classified into three categories: co-training style clustering [19], [26], [27], [28], multi-kernel clustering [29], [30], [31], [32], and multi-view subspace clustering [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43]. Co-training style clustering works on multi-view data using the co-training strategy [44]. It bootstraps the partitions of different views by using the prior or learned knowledge from one another. By iteratively carrying out this strategy, the partitions of all views arrive at the broadest consensus. Multiple kernel clustering predefines a group of base kernels, and then combines these kernels either linearly or non-linearly to improve the clustering performance. Multi-view subspace clustering aims to learn a unified representation from the feature subspaces of all views by assuming that all views share this unified representation. Then, this unified representation is fed into a clustering model to produce the final result. In general, co-training based methods rely on conditional independence, multi-kernel clustering methods have high computational complexity, and multi-view subspace clustering methods are sensitive to initialization. In the experiment section (i.e., Section 5), we will compare our model with representative approaches of these categories.

Apart from clustering, multi-view learning methods and applications have been investigated in [45], [46], [47], [48], [49], [50], to list a few.

3 Graph-based Multi-view Clustering

This section presents the proposed GMC method. Before that, let us describe some notational conventions. Throughout the paper, matrices are written in boldface capital letters (e.g., $\mathbf{X}$). Vectors are written in boldface lowercase letters (e.g., $\mathbf{x}$). Scalars are written in lowercase letters (e.g., x). For a matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, the jth column vector and the ijth 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 $\|\mathbf{x}\|_p$. Further, $\mathbf{I}$ denotes the identity matrix, and $\mathbf{1}$ denotes a column vector with all entries of one.

Now we formally present our GMC. GMC consists of *SIG matrix construction*, *multiple data graph fusion*, and *multi-view clustering with constrained Laplacian rank*. The main technologies and theories of each component are presented in the subsections. As a whole, GMC models the learning of the SIG matrix of each view, the learning of the fusion graph matrix of all views, and the clustering task into a single framework, which weights each view automatically, generates the SIG matrix of each view and the fusion graph matrix of all views jointly, and produces the clusters directly on the fusion graph matrix.

3.1 SIG Matrix Construction

For a multi-view data set with m views, let $\mathbf{X}^1, ..., \mathbf{X}^m$ be the data matrices of the m views and $\mathbf{X}^v = \{\mathbf{x}_1^v, ..., \mathbf{x}_n^v\} \in \mathbb{R}^{d_v\times n}$ be the v-th view data, where d_v is the dimensionality of the vth view, and n is the number of data points. The most popular method for transforming the data matrix $\mathbf{X}^v$ with similarities $\mathbf{S}^v \in \mathbb{R}^{n\times n}$ to a graph is using a *k*-nearest neighbor graph. If x_j belongs to the *k*-nearest neighbors of x_i, then x_i and x_j are connected. The weight of the edge between them is typically defined by the Gaussian kernel $S(\mathbf{x}_i^v, \mathbf{x}_j^v) = \exp(\frac{-||\mathbf{x}_i^v - \mathbf{x}_j^v||^2}{2\sigma^2})$, where σ controls the width of the neighborhoods. One major disadvantage of this method is that the hyper-parameter σ is hard to set in practice due to the noise and outliers in the data.

The study in [51] found that sparse representation is robust to noise and outliers. Also, it is desirable to construct a SIG matrix of one view such that a smaller distance between two data points corresponds to a large similarity value, and a larger distance between two data points corresponds to a small (or zero) similarity value. Toward this end, we use a sparse representation method to construct the SIG matrices. Mathematically, we model the problem as follows:

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

where $\{\mathbf{S}^v\}$ denotes $\mathbf{S}^1, ..., \mathbf{S}^m$.

We normalize $\mathbf{1}^T\mathbf{s}_i^v = 1$, which makes the second term constant. That is, the normalization $\mathbf{1}^T\mathbf{s}_i^v = 1$ is equivalent to the sparse constraint on $\mathbf{S}$. Then, problem (1) becomes

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

However, problem (2) has a trivial solution, i.e., only one data point with the smallest distance to $\mathbf{x}_i^v$ having the value 1, while all the other data points have the value 0. Following [52], we add a prior on problem (2), formulated as

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

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 problem (3). Here, we construct each SIG matrix for each view independently as each SIG has no relationship

with the others. In the next subsection, we couple each SIG matrix with a unified graph matrix, which is also our key graph matrix.

3.2 Multiple Data Graph Fusion

As mentioned in Section 1, we propose a model in which each view is weighted automatically, and the SIG matrices and the unified graph matrix are learned jointly so that they can help each other in a mutual reinforcement manner. Specially, we compute a unified matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ from the SIG matrices $\mathbf{S}^1, ..., \mathbf{S}^m$ by solving the following optimization problem:

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

where $\mathbf{u}_i \in \mathbb{R}^{n\times 1}$ is a column vector, u_{ij} is the j-th element of $\mathbf{u}_i$, and w_v is the weight of the v-th SIG matrix $\mathbf{S}^v$. According to Theorem 1, the weights $\mathbf{w} = \{w_1, ...w_m\}$ are determined automatically.

Theorem 1. *If the weights* $\mathbf{w}$ *are fixed, solving problem (4) is equivalent to solving the following problem:*

$$\min_{\mathbf{U}} \sum_{v=1}^{m} \sqrt{\left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2}, \quad s.t.\ \forall i, u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1. \tag{5}$$

Proof. The Lagrange function of Eq. (5) is shown below. Note that we use the terms problem $(\cdot)$ and Eq. $(\cdot)$ interchangeably in the paper as each problem is modeled as an equation.

$$\sum_{v=1}^{m} \sqrt{\left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2} + \Theta(\Lambda, \mathbf{U}) \tag{6}$$

where Λ is the Lagrange multiplier, and $\Theta(\Lambda, \mathbf{U})$ is the formalized term derived from constraints.

Taking the derivative of Eq. (6) with respect to $\mathbf{U}$ and setting the derivative to zero, we have

$$\sum_{v=1}^{m} w_v \frac{\partial \|\mathbf{U}-\mathbf{S}^v\|_F^2}{\partial \mathbf{U}} + \frac{\partial \Theta(\Lambda, \mathbf{U})}{\partial \mathbf{U}} = 0 \tag{7}$$

where

$$w_v = \frac{1}{2\sqrt{\left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2}}. \tag{8}$$

If w_v is fixed, the derivative of the Lagrange function of Eq. (4) is equal to Eq. (7). Thus, Eq. (4) is equivalent to Eq. (5). The weights $\mathbf{w}$ are also determined by Eq. (8). □

Then, combining problem (3) and problem (4), we learn for $\mathbf{S}^1, ..., \mathbf{S}^v$ and $\mathbf{U}$ by solving the following problem:

$$\begin{aligned} \min_{\{\mathbf{S}^v\},\mathbf{U}} &\sum_{v=1}^{m}\sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta \sum_{v=1}^{m}\sum_{i}^{n} \|\mathbf{s}_i^v\|_2^2 \\ &+ \sum_{v=1}^{m} w_v \left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2 \\ &s.t.\ \forall v, s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T\mathbf{s}_i^v = 1, \\ &u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1. \end{aligned} \tag{9}$$

As we can see, the learning of each SIG matrix $\mathbf{S}^1, ..., \mathbf{S}^m$ and the unified graph matrix $\mathbf{U}$ is coupled into a joint problem. In such a way, the learning of both can help each other naturally.

3.3 Multi-view Clustering with Constrained Laplacian Rank

This subsection aims to solve the final problem, i.e., producing the clustering result directly on the unified graph matrix $\mathbf{U}$ without an additional clustering algorithm or step. So far, the unified graph matrix $\mathbf{U}$ obtained through Eq. (9) above cannot tackle this problem.

Now we give an efficient and yet simple solution to achieve this goal by imposing a rank constraint on the graph Laplacian matrix of the unified matrix $\mathbf{U}$. In graph theory, $\mathbf{L}_U = \mathbf{D}_U - (\mathbf{U}^T + \mathbf{U})/2$ is called the graph Laplacian matrix of a graph matrix ($\mathbf{U}$ in this case), where the degree matrix $\mathbf{D}_U$ is defined as a diagonal matrix whose i-th diagonal element is $\sum_j (u_{ij} + u_{ji})/2$. If the unified matrix $\mathbf{U}$ is non-negative, then the Laplacian matrix has the following theorem [53], [54].

Theorem 2. *The multiplicity* r *of the eigenvalue 0 of the Laplacian matrix* $\mathbf{L}_U$ *is equal to the number of connected components in the graph of the unified matrix* $\mathbf{U}$*.*

The proof of Theorem 2 can be found in [53], [54]. As a conclusion, Theorem 2 says that if $rank(\mathbf{L}_U) = n - c$ as $c = r$, the corresponding $\mathbf{U}$ is an ideal case based on which the data points are partitioned into c clusters directly. Thus, there is no need to run an additional clustering algorithm on the unified matrix $\mathbf{U}$ to produce the final clusters. Motivated by Theorem 2, we add a rank constraint to problem (4). Then our multi-view clustering model is turned into

$$\begin{aligned} \min_{\{\mathbf{S}^v\},\mathbf{U}} &\sum_{v=1}^{m}\sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta \sum_{v=1}^{m}\sum_{i}^{n} \|\mathbf{s}_i^v\|_2^2 \\ &+ \sum_{v=1}^{m} w_v \left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2 \\ &s.t.\ \forall v, s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T\mathbf{s}_i^v = 1, \\ &u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1, rank(\mathbf{L}_U) = n - c. \end{aligned} \tag{10}$$

It is difficult to solve Eq. (10) because $\mathbf{L}_U$ depends on the target variable $\mathbf{U}$, and the constraint $rank(\mathbf{L}_U) = n - c$ is also nonlinear.

Let $\vartheta_i(\mathbf{L}_U)$ be the i-th smallest eigenvalue of $\mathbf{L}_U$. It is noted that $\vartheta_i(\mathbf{L}_U) \geq 0$ as $\mathbf{L}_U$ is positive semi-definite [53]. 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 [55], we know

$$\sum_{i=1}^{c} \vartheta_i(\mathbf{L}_U) = \min_{\mathbf{F}\in\mathbb{R}^{n\times c}, \mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}) \tag{11}$$

where $\mathbf{F} = \{\mathbf{f}_1, ...\mathbf{f}_c\}$ is the embedding matrix. Denote that the embedding matrix $\mathbf{F}$ is exactly what spectral clustering tends to learn as we introduced in Section 2.

Plugging the item Eq. (11) into Eq. (10), formally, our objective function is

$$\begin{aligned} \min_{\{\mathbf{S}^v\},\mathbf{U}} &\sum_{v=1}^{m}\sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta \sum_{v=1}^{m}\sum_{i}^{n} \|\mathbf{s}_i^v\|_2^2 \\ &+ \sum_{v=1}^{m} w_v \left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2 + 2\lambda Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}) \\ &s.t.\ \forall v, s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T\mathbf{s}_i^v = 1, \\ &u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1, \mathbf{F}^T\mathbf{F} = \mathbf{I}. \end{aligned} \tag{12}$$

When λ is large enough, the optimal solution to problem (12) will make $\sum_{i=1}^{k}\vartheta_i(\mathbf{L}_U)=0$ hold. It is worth stressing that the parameter λ does not need to be tuned. In practice, we increase or decrease the value of λ when the number of connected components is smaller or greater than c. Hereto, the resulting unified graph matrix $\mathbf{U}$ contains c connected components exactly, which partitions the data points into c clusters. In the next section, we propose a novel algorithm to solve problem (12), and optimize its objective function with alternating rules.

4 Optimization Algorithms

4.1 Optimization Algorithm for Problem (12)

Solving problem (12) to give every variable an optimized solution at once is still challenging because all the variables in the objective function are coupled together. Also, the constraints are not smooth. Under the assumption that $\mathbf{w}$, $\mathbf{S}^1,...,\mathbf{S}^m$ and $\mathbf{F}$ have been obtained, we can calculate $\mathbf{U}$ via the Augmented Lagrange Multiplier (ALM) scheme. ALM has shown its effectiveness in many matrix learning problems [56]. Similarly, $\mathbf{w}$, $\mathbf{S}^1,...,\mathbf{S}^m$ and $\mathbf{F}$ are updated when the other variables are fixed, which inspires us to develop an alternating iterative algorithm to solve problem (12). The specific updated rules are shown below:

Fix w, U and F, update $\mathbf{S}^1,...,\mathbf{S}^m$: When $\mathbf{w}$, $\mathbf{U}$ and F are fixed, the last term of problem (12) is a constant. Updating $\mathbf{S}^1,...,\mathbf{S}^m$ is to solve the following problem:

$$\begin{aligned}\min_{\{\mathbf{S}^v\}}&\sum_{v=1}^{m}\sum_{i,j=1}^{n}\left\|\mathbf{x}_i^v-\mathbf{x}_j^v\right\|_2^2 s_{ij}^v+\beta\sum_{v=1}^{m}\sum_{i}^{n}\|\mathbf{s}_i^v\|_2^2\\&+\sum_{v=1}^{m}w_v\left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2\\&s.t.\ \forall v, s_{ii}^v=0, s_{ij}^v\geq 0, \mathbf{1}^T\mathbf{s}_i^v=1.\end{aligned}\tag{13}$$

As we can see, updating $\mathbf{S}^v$ for each view is independent. Thus, we can update $\mathbf{S}^v$ one by one, formulated as

$$\begin{aligned}\min_{\mathbf{S}^v}&\sum_{i,j=1}^{n}\left\|\mathbf{x}_i^v-\mathbf{x}_j^v\right\|_2^2 s_{ij}^v+\beta\sum_{i}^{n}\|\mathbf{s}_i^v\|_2^2+w_v\left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2\\&s.t.\ s_{ii}^v=0, s_{ij}^v\geq 0, \mathbf{1}^T\mathbf{s}_i^v=1.\end{aligned}\tag{14}$$

In practice, we prefer a data point having similarities with its neighbours. That is, we learn $\mathbf{s}_i^v$ in $\mathbf{S}^v$ with k nonzero values, where k is the number of neighbours. For simplicity, we omit the detailed solution steps and give the final solution below (See the detailed solution steps in Appendix A):

$$s_{ij}^v=\begin{cases}\frac{e_{i,k+1}-e_{ij}+2w_vu_{ij}-2w_vu_{i,k+1}}{ke_{i,k+1}-\sum_{h=1}^{k}e_{ih}-2kw_vu_{i,k+1}+2\sum_{h=1}^{k}w_vu_{ih}} & j\leq k,\\ 0 & j>k.\end{cases}\tag{15}$$

where $e_{ij}=\|\mathbf{x}_i^v-\mathbf{x}_j^v\|_2^2$.

Fix $\mathbf{S}^1,...,\mathbf{S}^m$, U and F, update w_v: When $\mathbf{S}^1,...,\mathbf{S}^m$, $\mathbf{U}$ and $\mathbf{F}$ are fixed, optimizing problem (12) for updating w_v is equivalent to optimizing problem (4). Thus, the value of w_v is updated by Eq. (8) formulated in Subsection 3.2.

Fix $\mathbf{S}^1,...,\mathbf{S}^m$, F and w, update U: Since $Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F})=\frac{1}{2}\sum_{i,j}\|\mathbf{f}_i-\mathbf{f}_j\|_2^2u_{ij}$, $\mathbf{S}^1,...,\mathbf{S}^m$ and $\mathbf{w}$ are fixed, the optimization problem (12) is transformed into

$$\begin{aligned}\min_{\mathbf{U}}&\sum_{v=1}^{m}\sum_{i,j=1}^{n}w_v(u_{ij}-s_{ij}^v)^2+\lambda\sum_{i,j=1}^{n}\|\mathbf{f}_i-\mathbf{f}_j\|_2^2u_{ij}\\&s.t.\ \forall i, u_{ij}\geq 0, \mathbf{1}^T\mathbf{u}_i=1.\end{aligned}\tag{16}$$

Note that problem (16) is independent for different i, so we can solve the following problem separately for each i:

$$\begin{aligned}\min_{\mathbf{u}^i}&\sum_{v=1}^{m}\sum_{j=1}^{n}w_v(u_{ij}-s_{ij}^v)^2+\lambda\sum_{j=1}^{n}\|\mathbf{f}_i-\mathbf{f}_j\|_2^2u_{ij}\\&s.t.\ \forall i, u_{ij}\geq 0, \mathbf{1}^T\mathbf{u}_i=1.\end{aligned}\tag{17}$$

Denote $d_{ij}=\|\mathbf{f}_i-\mathbf{f}_j\|_2^2$, then Eq. (17) is rewritten as

$$\begin{aligned}\min_{\mathbf{u}^i}&\sum_{v=1}^{m}\sum_{j=1}^{n}w_v(u_{ij}-s_{ij}^v)^2+\lambda\sum_{j=1}^{n}d_{ij}u_{ij}\\&s.t.\ \forall i, u_{ij}\geq 0, \mathbf{1}^T\mathbf{u}_i=1.\end{aligned}\tag{18}$$

Further denote $\mathbf{d}_i$ as a vector with the j-th element as d_{ij}, and similarly for $\mathbf{u}_i$ and $\mathbf{s}_i$. Theorem 3 reveals that solving problem (18) is equivalent to solving problem (19).

Theorem 3. *Solving problem (18) is equivalent to solving the following problem:*

$$\begin{aligned}\min_{\mathbf{u}_i}&\sum_{v=1}^{m}\left\|\mathbf{u}_i-\mathbf{s}_i^v+\frac{\lambda}{2mw_v}\mathbf{d}_i\right\|_2^2\\&s.t.\ \forall i, u_{ij}\geq 0, \mathbf{1}^T\mathbf{u}_i=1.\end{aligned}\tag{19}$$

Proof. Rewriting Eq. (19) in an element form, we have

$$\begin{aligned}&\sum_{v=1}^{m}\sum_{j=1}^{n}\left[(u_{ij}-s_{ij}^v)+\frac{\lambda}{2mw_v}d_{ij}\right]^2\\&=\sum_{v=1}^{m}\sum_{j=1}^{n}\begin{bmatrix}u_{ij}^2-2u_{ij}s_{ij}^v+(s_{ij}^v)^2+\frac{\lambda}{mw_v}d_{ij}u_{ij}\\-\frac{\lambda}{mw_v}d_{ij}s_{ij}^v+(\frac{\lambda}{2mw_v}d_{ij})^2\end{bmatrix}.\end{aligned}$$

Both $\frac{\lambda}{mw_v}d_{ij}s_{ij}^v$ and $\frac{\lambda}{2mw_v}d_{ij}$ are constants when $\mathbf{w}$ and $\mathbf{F}$ are fixed. Omitting the latter two items, we get

$$\begin{aligned}&\sum_{v=1}^{m}\sum_{j=1}^{n}\left[u_{ij}^2-2u_{ij}s_{ij}^v+(s_{ij}^v)^2+\frac{\lambda}{mw_v}d_{ij}u_{ij}\right]\\&=\sum_{v=1}^{m}\sum_{j=1}^{n}\left[(u_{ij}-s_{ij}^v)^2+\frac{\lambda}{mw_v}\sum\nolimits_j d_{ij}u_{ij}\right]\\&\simeq\sum_{v=1}^{m}\sum_{j=1}^{n}w_v(u_{ij}-s_{ij}^v)^2+\lambda\sum\nolimits_j d_{ij}u_{ij}.\end{aligned}$$

The symbol $\simeq$ indicates that the latter equation, i.e., problem (18), can be achieved by multiplying the former equation with the weight w_v. □

The solution to problem (19) is presented in the next subsection, where we develop a simple and effective algorithm to solve it.

Fix w, $\mathbf{S}^1,...,\mathbf{S}^m$, and U, update F: With $\mathbf{w}$, $\mathbf{S}^1,...,\mathbf{S}^m$, and $\mathbf{U}$ fixed, optimizing $\mathbf{F}$ amounts to solving the following problem:

$$\min_{\mathbf{F}}\ Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F}),\ s.t.\ \mathbf{F}^T\mathbf{F}=\mathbf{I}.\tag{20}$$

The optimal solution $\mathbf{F}$ is formed by the c eigenvectors of $\mathbf{L}_U$ corresponding to the c smallest eigenvalues.

Hereto, all the variables have been updated. As can be seen from the above rules, updating one variable (e.g., $\mathbf{S}^1, ..., \mathbf{S}^m$) needs to use the other variables (e.g., $\mathbf{w}$, $\mathbf{U}$ and $\mathbf{F}$). In practice, we initialize SIG matrices $\mathbf{S}^1, ..., \mathbf{S}^m$ first by solving problem (3). Note that initializing SIG matrix for each view is independent. Here we take $\mathbf{S}^v$ as an example. According to [52], the initial solution to s_{ij}^v in $\mathbf{S}^v$ is shown below (See the details in Appendix B):

$$s_{ij}^v = \begin{cases} \frac{b_{i,k+1} - b_{ij}}{k b_{i,k+1} - \sum_{h=1}^{k} b_{ih}} & j \leq k, \\ 0 & j > k. \end{cases} \tag{21}$$

where $b_{ij} = \|\mathbf{x}_i^v - \mathbf{x}_j^v\|_2^2$, and k is the number of neighbours.

In summary, the procedure for solving the proposed objective function Eq. (12) is listed in Algorithm 1.

Algorithm 1 Optimization algorithm

Input: Data for m views $\mathbf{X}^1, ..., \mathbf{X}^m$ with $\mathbf{X}^v \in \mathbb{R}^{d_v \times n}$, the number of clusters c, the number of neighbours k, initial parameter λ (tuned automatically in the algorithm).
Output: The learned unified matrix $\mathbf{U} \in \mathbb{R}^{n \times n}$.

```
Initialize SIG matrices {S^v} by using Eq. (21),
Initialize the weight for each view, w_v = 1/m,
Initialize U by connecting {S^v} with w, and then F is obtained by solving Eq. (20),
repeat
    Fix w, U and F, update {S^v} by using Eq. (15);
    Fix {S^v}, F and U, update w by using Eq. (8);
    Fix w, {S^v} and F, update U by solving problem (19);
    Fix w, {S^v} and U, update F which is formed by the c eigenvectors of L_U corresponding to the c smallest eigenvalues;
until Theorem 2 or the maximum iteration reached.
return The learned unified matrix U with exact c connected components, which are the final clusters.
```

4.2 Solution to Problem (19)

Since the second and third terms in Eq. (19) are constants, we define $\mathbf{q}^v = \mathbf{s}_i^v - \frac{\lambda}{2mw_v}\mathbf{d}_i$, and then problem (19) can be simplified to

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

Let ϕ and φ be the Lagrange multipliers for the constraints. The Lagrangian function of problem (22) is

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

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

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

$$\begin{cases} \forall j,\ \sum_{v=1}^{m} u_{ij}^* - \sum_{v=1}^{m} q_j^v - \phi^* - \varphi_j^* = 0 & (24) \\ \forall j,\ u_{ij}^* \geq 0 & (25) \\ \forall j,\ \varphi_j^* \geq 0 & (26) \\ \forall j,\ u_{ij}^*\varphi_j^* = 0 & (27) \end{cases}$$

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

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

$$\mathbf{u}_i^* = \frac{\sum_{v=1}^{m}\mathbf{q}^v}{m} + \frac{\mathbf{1}}{n} - \frac{\sum_{v=1}^{m}\mathbf{1}^T q^v\mathbf{1}}{mn} - \frac{\mathbf{1}^T\varphi^*\mathbf{1}}{mn} + \frac{\varphi^*}{m}. \tag{28}$$

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. (28) becomes $\mathbf{u}_i^* = \mathbf{p} - \hat{\varphi}^*\mathbf{1} + \frac{\varphi^*}{m}$. Furthermore, for $\forall j$, we have

$$u_{ij}^* = p_j - \hat{\varphi}^* + \frac{\varphi_j^*}{m}. \tag{29}$$

According to Eqs. (25)-(27) and Eq. (29), we know that $p_j - \hat{\varphi}^* + \frac{\varphi_j^*}{m} = (p_j - \hat{\varphi}^*)_+$, where $(a)_+ = max(a, 0)$. Namely, the optimal solution of u_{ij}^* can be obtained if $\hat{\varphi}^*$ is known, formulated as

$$u_{ij}^* = (p_j - \hat{\varphi}^*)_+. \tag{30}$$

Due to Eq. (29), we derive $\varphi_j^* = m(u_{ij}^* + \hat{\varphi}^* - p_j)$. Similarly, we get $\varphi_j^* = m(\hat{\varphi}^* - p_j)_+$ according to Eqs. (25)-(27). Owning to the denotation $\hat{\varphi}^* = \frac{\mathbf{1}^T\varphi^*}{mn}$, the optimal solution $\hat{\varphi}^*$ is represented as $\hat{\varphi}^* = \frac{1}{n}\sum_{j=1}^{n}(\hat{\varphi}^* - p_j)_+$. Then, we define a function of $\hat{\varphi}$ as

$$f(\hat{\varphi}) = \frac{1}{n}\sum_{j=1}^{n}(\hat{\varphi} - p_j)_+ - \hat{\varphi}. \tag{31}$$

Thus, $\hat{\varphi}^*$ is obtained by solving the root finding problem as $f(\hat{\varphi}^*) = 0$. Since $\hat{\varphi} \geq 0$, $f'(\hat{\varphi}_t) \leq 0$ and $f'(\hat{\varphi}_t) \leq 0$ is a piece-wise linear and convex function, the root of $f(\hat{\varphi}) = 0$ can be solved via the Newton method efficiently, as follows

$$\hat{\varphi}_{t+1} = \hat{\varphi}_t - \frac{f(\hat{\varphi}_t)}{f'(\hat{\varphi}_t)}. \tag{32}$$

4.3 Computational Complexity Analysis

The computational complexity in solving problem (12) consists of four parts. Specifically, the update of $\mathbf{S}^1, ..., \mathbf{S}^m$ has the computational complexity of $\mathcal{O}(mnk)$, where k is the number of neighbours. The update of weights w by Eq. (8) takes $\mathcal{O}(mn^2)$. The learning of the unified matrix $\mathbf{U}$ takes $\mathcal{O}(cn)$, where c is the number of clusters. Updating $\mathbf{F}$ needs to calculate the eigenvectors of the Laplacian matrix. It costs $\mathcal{O}(cn^2)$, which is the complexity of most Laplacian matrix eigenvector problems. Besides, we initialize SIG matrices $\mathbf{S}^1, ..., \mathbf{S}^m$ by solving problem (3), taking $\mathcal{O}(mnkd)$, where $d = max(d_1, ..., d_m)$. Suppose the updates stop after t iterations, the overall computational complexity is

$$\mathcal{O}(((mk + mn + c + cn)n)t + mnkd).$$

Note that $m \ll n$, $k \ll n$, $c \ll n$, and $t \ll n$. The main complexity is the eigen-decomposition procedure

which is also a basic step in graph-based clustering methods and spectral clustering methods. For large-scale datasets, the study on large-scale graph-based clustering, such as [17], [57], [58], can be applied to speed up our method. Another possibility is to use sparse matrix to speed up as the initialized SIG matrices are sparse since we make a data point having high similarities with its neighbors.

4.4 Convergence Analysis

Since problem (12) is not a joint convex problem of all variables, obtaining a globally optimal solution is still an open problem. We solve problem (12) using an alternating algorithm (i.e., Algorithm 1). As each sub-problem is convex and we find the optimal solution of each sub-problem, Algorithm 1 converges obviously. The convergence of each sub-problem is shown as follows.

Update $\mathbf{S}^1, ..., \mathbf{S}^m$. It is easy to check that the objective function of problem (13) is a convex function because the second order derivative of this function with respect to $\mathbf{s}_i^v$ is equal to 1 (positive value). Thus, it decreases monotonically with the ALM scheme.

Update $\mathbf{w}$. Problem (4) is a linear convex function, and we give a closed-form solution to weights $\mathbf{w}$.

Update $\mathbf{U}$. Following [10], we prove the convergence of problem (16) based on the lemma below.

Lemma 1. *[59] For any non-zero matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times n}$, the following inequality holds:*

$$\|\mathbf{A}\|_F - \frac{\|\mathbf{A}\|_F^2}{2\|\mathbf{B}\|_F} \leq \|\mathbf{B}\|_F - \frac{\|\mathbf{B}\|_F^2}{2\|\mathbf{B}\|_F}. \tag{33}$$

Denote that $\tilde{\mathbf{U}}$ is the updated result in each iteration and $\mathcal{G}(\mathbf{U}) = 2\lambda Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F})$, we can derive

$$\sum_{v=1}^{m} \frac{\|\tilde{\mathbf{U}} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F} + \mathcal{G}(\tilde{\mathbf{U}}) \leq \sum_{v=1}^{m} \frac{\|\mathbf{U} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F} + \mathcal{G}(\mathbf{U}). \tag{34}$$

According to Lemma 1, we get

$$\begin{aligned} &\sum_{v=1}^{m} \|\tilde{\mathbf{U}} - \mathbf{S}^v\|_F - \sum_{v=1}^{m} \frac{\|\mathbf{U} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F} \\ &\leq \sum_{v=1}^{m} \|\mathbf{U} - \mathbf{S}^v\|_F - \sum_{v=1}^{m} \frac{\|\mathbf{U} - \mathbf{S}^v\|_F^2}{2\|\mathbf{U} - \mathbf{S}^v\|_F}. \end{aligned} \tag{35}$$

Summing Eq. (34) and Eq. (35) over both sides, we have

$$\sum_{v=1}^{m} \|\tilde{\mathbf{U}} - \mathbf{S}^v\|_F + \mathcal{G}(\tilde{\mathbf{U}}) \leq \sum_{v=1}^{m} \|\mathbf{U} - \mathbf{S}^v\|_F + \mathcal{G}(\tilde{\mathbf{U}}). \tag{36}$$

The inequality (36) indicates that the objective function of problem (16) decreases monotonically in each iteration until it converges.

Update $\mathbf{F}$. The Hessian matrix of Eq. (20) is

$$\frac{\partial^2 Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F})}{\partial\mathbf{F}\partial\mathbf{F}^T} = \mathbf{L}_U + \mathbf{L}_U^T \tag{37}$$

where the Laplacian matrix $\mathbf{L}_U$ is positive semi-definite. Then, the Hessian matrix of Eq. (20) is also positive semi-definite. So, Eq. (20) is a convex function with respect to $\mathbf{F}$ updated by $arg\ min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{L}_U\mathbf{F})$ in each iteration.

5 Experiments

In this section, we report the experiments that have been conducted to evaluate the performance of the proposed GMC model using ten data sets (two toy data sets and eight real-world data sets). We performed experiments on a Windows Server 2008 R2 with Intel Xeon processor, 24GB RAM and MATLAB development environment. We used MATLAB development environment because all baselines used MATLAB. That is, all experiments are performed in the same environment.

5.1 Experiments on Toy Data

Following [12], we generated two toy data sets, and conducted experiments on them to give a visual illustration of the capability of GMC.

Toy data sets. The first toy data set consists of two-views, named *Two-Moon data set*, as shown in Fig. 2a and Fig. 2e. 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). Each cluster has 100 sample points. The second toy data set also consists of two-views but generated with different patterns, named *Three-Ring data set*, as shown in Fig. 3a and Fig. 3e. Each view is generated randomly also with random Gaussian noise. Each view has three clusters with 30 sample points, 90 sample points and 180 sample points, respectively.

Results on toy data sets. Fig. 2b and Fig. 2f show the constructed graphs with the initialized SIG matrices produced by Eq. (21). If we look at each view in isolation, we can see that the two clusters are connected and not easy to separate. Fig. 2c and Fig. 2g show the constructed graphs with the learned SIG matrices in Algorithm 1. As can be seen, some edges have been weakened (or deleted) and some edges have been strengthened, which indicates that the learned unified matrix can improve the SIG matrices. The clusters are still not separated. Fig. 2d and Fig. 2h show the constructed graphs with the learned unified matrix. The unified matrix separates the two moons very well because it can exploit the complementary information from the two views. That is, the hard-to-separate data points in each view can be easily separated by using the complementary information from the other view. The same conclusions can be drawn for Fig. 3. In summary, the learning of the SIG matrices and the unified matrix can help improve each other, and the unified matrix partitions data well.

5.2 Experiments on Real-World Data

To demonstrate how the clustering performance can be improved using our algorithm GMC, we compare GMC with nine baseline algorithms on eight benchmark data sets.

Data sets. We conduct experiments on the following data sets, which are commonly used in the literature.

- *One-hundred plant species leaves data set* [1] *(100leaves)*: It consists of 1600 samples from each of one hundred plant species. For each sample, shape descriptor, fine scale margin and texture histogram are given.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2903810, IEEE Transactions on Knowledge and Data Engineering

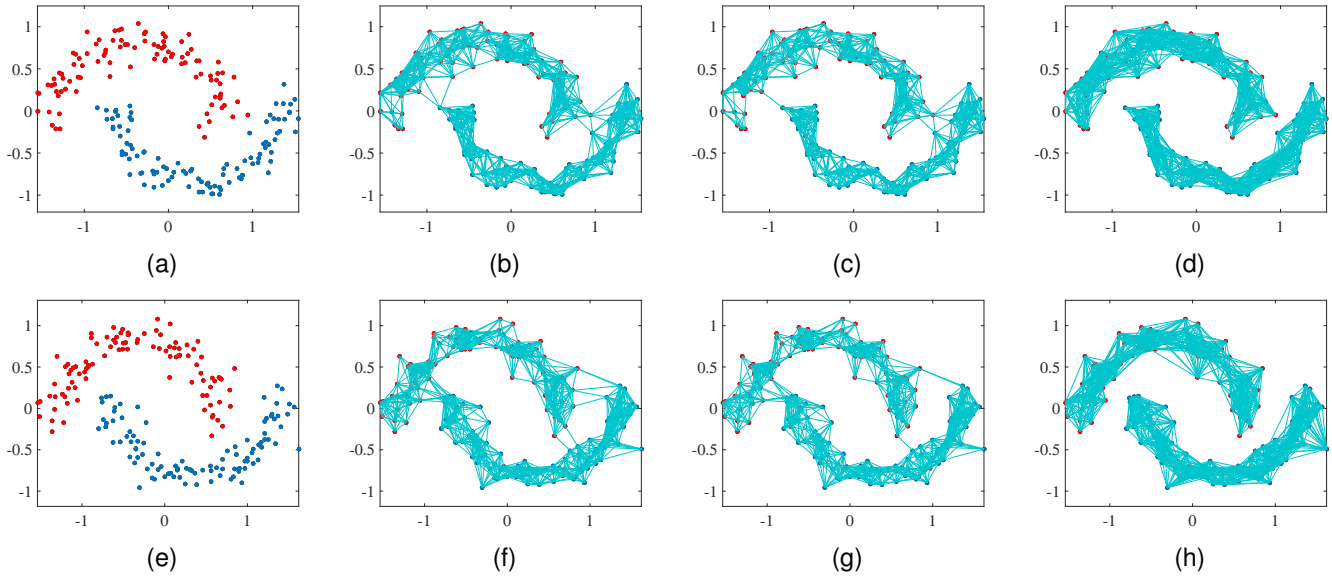


Fig. 2. Results on the Two-Moon data set. The upper row is the first view. The lower row is the second view. 2a and 2e are the generated sample data points of the first view and the second view, respectively. 2b and 2f are the constructed graphs with the initialized SIG matrices for the first view and the second view, respectively. 2c and 2g are the constructed graphs with the learned SIG matrices for the first view and the second view, respectively. 2d and 2h are the constructed graphs with the unified matrix for the first view and the second view, respectively.

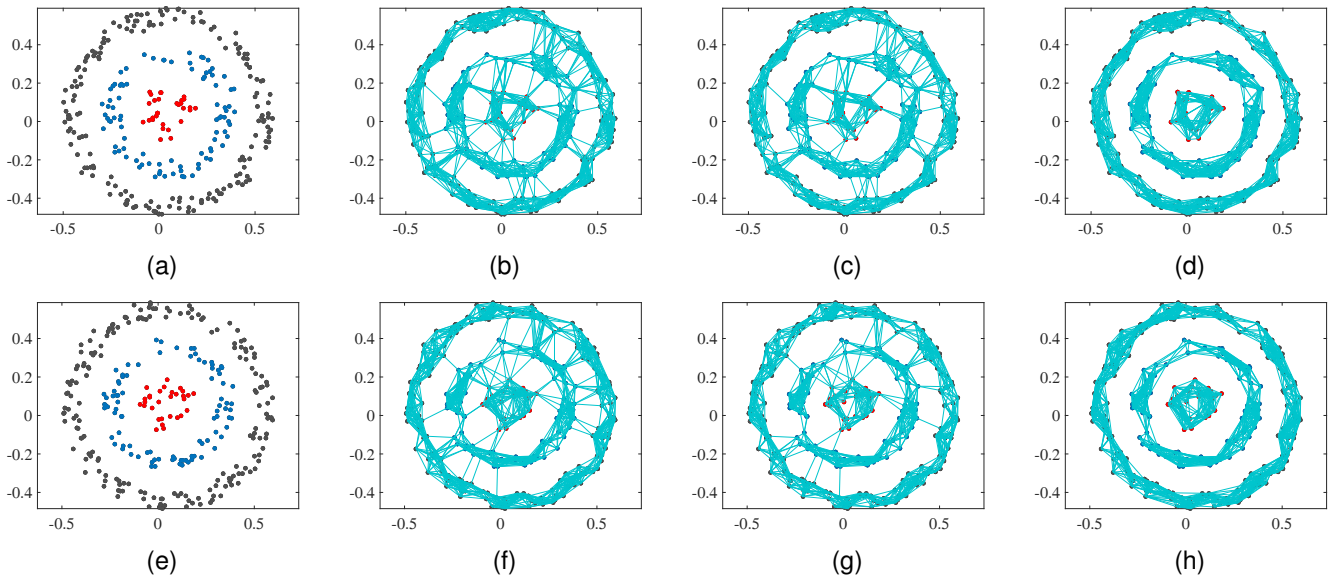


Fig. 3. Results on the Three-Ring data set. The notes are the same to Fig. 2.

- *3 source data set*[2] *(3source)*: It consists of 169 news, which were reported by three news organizations, i.e., BBC, Reuters, and The Guardian. Each news was manually annotated with one of six topical labels.
- *Amsterdam Library of Object Images*[3] *(ALOI)*. It is from the work in [43]. The data set consists of 11025 images of 100 small objects. Each image is represented with four types of features, i.e., RGB, HSV, Color similiarity and Haralick features.
- *BBC data set*[4] *(BBC)*. It is collected from the BBC news website. BBC data set consists of 685 documents. Each document was split into four segments and was manually annotated with one of five topical labels.
- *Handwritten digit 2 source data set*[5] *(Hdigit)*. This handwritten digits (0-9) data set is from two sources, i.e., MNIST Handwritten Digits and USPS Handwritten Digits. The data set consists of 10000 samples.
- *Mfeat handwritten digit data set*[6] *(Mfeat)*. This handwritten digits (0-9) data set is from the UCI repository. The data set consists of 2000 samples. Each sample is represented by six types of features.

2. http://mlg.ucd.ie/datasets/3sources.html
3. http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView
4. http://mlg.ucd.ie/datasets/segment.html
5. https://cs.nyu.edu/~roweis/data.html
6. http://archive.ics.uci.edu/ml/datasets/Multiple+Features

- *Newsgroups data set*[7] *(NGs)*. It is a subset of the 20 Newsgroup datasets. NGs consists of 500 newsgroup documents. Each raw document was pre-processed with three different methods (giving three views), and was annotated with one of five topical labels.
- *WebKB data set*[8] *(WebKB)*. It consists of web-pages collected from computer science departments of university. There are 203 web-pages with 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.

All the data sets are summarized in Table 1, where n, m, and c denote the number of instances, views, and clusters, respectively. d_v denotes the dimension of features in view v. More information of each data set can be found via the link in the footnote.

TABLE 1
Summary of the benchmark data sets

Data set	n	m	c	d_1	d_2	d_3	d_4	d_5	d_6
100leaves	1600	3	100	64	64	64	–	–	–
3sources	169	3	6	3560	3631	3068	–	–	–
ALOI	11025	4	100	77	13	64	64	–	–
BBC	685	4	5	4659	4633	4665	4684	–	–
Hdigit	10000	2	10	784	256	–	–	–	–
Mfeat	2000	6	10	216	76	64	6	240	47
NGs	500	3	5	2000	2000	2000	–	–	–
WebKB	203	3	4	1703	230	230	–	–	–

Baselines. We compare our method with the following baseline methods: Single view *K*-means (**S*K*-means**), Single view Normalized cut (**SNcut**) [60], Multi-view *K*-means Clustering (**M*K*C**) [41], Multi-view clustering via Non-negative Matrix Factorization (**MultiNMF**) [37], Co-regularized Spectral Clustering (**CoregSC**) [19], Multi-view Spectral Clustering (**MSC**) [20], Adaptive Structure-based Multi-view clustering (**ASMV**) [11], Multiple Graph Learning (**MGL**) [6], Multi-view Clustering with Graph Learning (**MCGL**) [12]. Note that S*K*-means and SNcut only works for single-view data. For multi-view data, we apply S*K*-means and SNcut to each view and report the results of the view that gives the best performance.

For the baselines, we obtained the original systems from their authors and ran the systems using their default parameter settings. Note that all baselines use MATLAB and we also use it. For GMC, we empirically set $k = 15$. The parameter λ is set to 1 as its initial value, which is tuned automatically in the clustering process for each data set. Specifically, in each iteration, we increase it (λ=λ*2) or decrease it (λ=$\lambda/2$) if the connected components of $\mathbf{U}$ is smaller or greater than the number of clusters c, respectively.

The clustering results are evaluated by comparing the obtained label of each instance with the provided label by the data set. Three metrics, the accuracy (ACC), the normalized mutual information (NMI), the adjusted rand index (ARI), and the F1 measure (F-measure) are used to measure the clustering performance. In order to randomize the experiments, we run each algorithm 10 times and report the means and standard deviations of the performance measures.

Results on real-world data sets. The comparison results are shown in Tables 2, 3, 4, and 5 where we also report the average score over eight data sets 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 GMC method is markedly better than all baselines. GMC gives the best performance on all the data sets, except the HW data set in terms of NMI and the 100leaves data set in terms of ARI. The results clearly show that our GMC method is a promising multi-view clustering method.
- All the graph-based methods, i.e, SNcut, ASMV, MGL, MCGL and GMC, perform robustly except MGL. For MGL, it feeds the learned unified representation to *K*-means in order to produce the final clusters. Since *K*-means is sensitive to the initialized cluster centers, it results in a high standard deviation for MGL.
- Compared with the recent graph-based methods ASMV and MCGL, GMC performs better or comparably. This shows that by jointly learning the individual graph and the unified graph, GMC can learn a better unified graph across all views.
- In general, MKC and MultiNMF are worse than CoregSC and MSC, both of which are worse than our method. One reason is that they all rely on additional clustering algorithms.
- MultiNMF has no value for the HW data set. The reason is that MultiNMF is not suitable for handling data containing negative numbers (HW data set in this case) as it is based on non-negative matrix factorization.
- In some cases, single view baseline methods, i.e., S*K*-means and SNcut, are even slightly better than some multi-view baseline methods. This indicates that exploring multi-view data still needs good techniques.

5.3 Mode Evaluation

To further demonstrate how the clustering performance can be improved by our method, three variants of GMC are created as follows:

a. The learned unified matrix does not go back to improve the initialized SIG matrix of each view, i.e., removing Step 5 in Algorithm 1, denoted by GMC-S.
b. The weights of different views are set to the same in Algorithm 1, denoted by GMC-W.
c. The learned embedding matrix is fed into an additional clustering algorithm (e.g., *K*-means) to produce the final clustering results, denoted by GMC-K.

We then compare GMC with these three variants. The results are shown in Fig. 4. From the Fig. 4, we can see that GMC outperforms its three variants. This indicates that our overall model GMC, i.e., weighting the graph of each view to learn the unified graph while helping each other without an additional clustering step, is promising.

7. http://lig-membres.imag.fr/grimal/data.html
8. https://linqs.soe.ucsc.edu/data

TABLE 2
Clustering performance comparison in terms of ACC on eight real-world data sets

ACC (%)	100leaves	3sources	ALOI	BBC	Hdigit	Mfeat	NGs	WebKB	Average
S*K*-means	57.96 (1.41)	44.02 (4.27)	49.22 (4.97)	38.20 (6.21)	50.23 (4.91)	70.20 (8.31)	22.72 (1.52)	72.32 (5.12)	50.61 (4.59)
SNcut	43.04 (1.29)	40.24 (→0)	19.95 (1.29)	33.14 (→0)	43.49 (0.30)	69.30 (→0)	23.60 (→0)	66.50 (0.00)	42.41 (0.36)
M*K*C	1.00 (→0)	46.63 (10.68)	1.01 (→0)	60.34 (11.08)	77.37 (6.11)	49.24 (27.78)	41.76 (4.28)	59.85 (6.45)	42.15 (8.30)
MultiNMF	67.78 (1.57)	48.58 (2.56)	1.02 (→0)	48.95 (2.68)	71.99 (4.26)	–.– (–.–)	20.20 (0.00)	53.69 (0.00)	39.03 (1.38)
CoregSC	77.06 (2.58)	54.79 (2.99)	52.17 (2.13)	47.01 (0.00)	81.59 (2.87)	75.56 (5.96)	27.68 (1.53)	59.70 (1.43)	59.45 (2.44)
MSC	73.79 (2.21)	47.51 (2.97)	47.38 (7.65)	67.32 (4.94)	72.39 (6.58)	79.18 (8.21)	31.12 (0.67)	47.34 (3.92)	58.26 (4.64)
ASMV	79.06 (→0)	33.73 (→0)	45.55 (→0)	33.72 (0.00)	75.19 (0.00)	57.45 (→0)	22.80 (→0)	72.41 (→0)	52.49 (→0)
MGL	69.04 (2.42)	67.51 (6.67)	48.07 (1.51)	53.96 (11.05)	85.94 (12.66)	74.40 (8.19)	82.18 (14.70)	73.84 (3.93)	69.37 (7.64)
MCGL	81.06 (→0)	30.77 (→0)	46.25 (→0)	35.33 (→0)	99.46 (0.00)	85.30 (0.00)	24.60 (0.00)	54.19 (0.00)	57.12 (→0)
GMC	**82.38 (→0)**	**69.23 (→0)**	**57.05 (→0)**	**69.34 (→0)**	**99.81 (0.00)**	**88.20 (→0)**	**98.20 (0.00)**	**76.35 (→0)**	**80.07 (→0)**

TABLE 3
Clustering performance comparison in terms of NMI on eight real-world data sets

NMI (%)	100leaves	3sources	ALOI	BBC	Hdigit	Mfeat	NGs	WebKB	Average
S*K*-means	80.29 (0.66)	24.28 (4.39)	68.71 (6.32)	8.93 (9.96)	47.70 (2.91)	71.50 (3.66)	5.87 (3.63)	36.77 (6.96)	43.03 (4.81)
SNcut	74.86 (0.51)	10.45 (→0)	47.43 (1.51)	1.85 (→0)	43.84 (0.62)	83.02 (0.00)	7.39 (→0)	20.58 (→0)	36.18 (0.33)
M*K*C	0.00 (0.00)	36.65 (10.05)	0.00 (0.00)	47.86 (8.51)	77.11 (4.30)	53.25 (36.83)	32.80 (5.95)	27.53 (2.52)	34.40 (8.52)
MultiNMF	86.36 (0.51)	46.67 (1.46)	2.06 (→0)	33.52 (2.36)	64.54 (3.30)	–.– (–.–)	1.56 (0.00)	3.83 (→0)	29.82 (0.95)
CoregSC	91.65 (0.59)	52.38 (1.98)	69.93 (1.32)	28.63 (0.00)	68.90 (1.73)	74.21 (3.27)	8.80 (0.77)	31.39 (2.36)	53.24 (1.50)
MSC	90.14 (0.76)	38.50 (2.27)	63.58 (5.44)	55.31 (1.44)	64.75 (1.30)	75.60 (3.24)	9.72 (1.26)	22.37 (1.65)	52.50 (2.17)
ASMV	90.09 (→0)	8.96 (→0)	67.67 (→0)	3.48 (0.00)	89.32 (0.00)	67.09 (→0)	6.30 (→0)	28.80 (→0)	45.21(→0)
MGL	87.53 (0.76)	57.68 (8.61)	70.52 (0.70)	36.97 (18.97)	94.09 (7.63)	82.64 (4.73)	83.04 (8.96)	**43.62 (1.43)**	69.51 (6.47)
MCGL	91.30 (0.00)	10.34 (→0)	66.57 (→0)	7.41 (→0)	98.32 (0.00)	**90.55 (0.00)**	10.72 (→0)	8.60 (→0)	47.98 (→0)
GMC	**92.92 (0.00)**	**62.16 (0.00)**	**73.50 (0.00)**	**56.28 (0.00)**	**99.39 (0.00)**	90.50 (→0)	**93.92 (→0)**	41.64 (0.00)	**76.29 (→0)**

TABLE 4
Clustering performance comparison in terms of ARI on eight real-world data sets

ARI (%)	100leaves	3sources	ALOI	BBC	Hdigit	Mfeat	NGs	WebKB	Average
S*K*-means	46.93 (1.65)	5.64 (4.55)	36.01 (7.84)	3.44 (5.55)	32.38 (4.12)	60.41 (6.07)	0.36 (0.72)	33.60 (12.12)	27.35 (5.33)
SNcut	30.22 (1.06)	4.41 (→0)	2.25 (0.32)	0.22 (→0)	27.70 (0.08)	70.29 (0.00)	0.42 (→0)	25.19 (→0)	20.09 (0.18)
M*K*C	0.00 (0.00)	24.61 (14.08)	0.00 (0.00)	34.50 (12.18)	68.91 (6.34)	42.80 (29.98)	14.31 (2.87)	29.16 (4.03)	26.79 (8.69)
MultiNMF	58.55 (1.60)	26.54 (3.71)	0.02 (0.00)	12.62 (3.50)	55.43 (2.99)	–.– (–.–)	→ 0 (→0)	1.43 (0.00)	19.32 (1.48)
CoregSC	**72.29 (1.92)**	33.39 (2.85)	40.97 (4.52)	27.27 (0.00)	65.97 (3.07)	68.85 (5.73)	1.29 (0.24)	31.80 (3.51)	42.35 (2.73)
MSC	67.88 (2.26)	26.18 (3.81)	33.05 (4.81)	46.58 (2.20)	57.32 (4.29)	68.03 (6.28)	4.90 (0.24)	20.91 (3.45)	40.61 (3.42)
ASMV	61.04 (→0)	-2.11 (→0)	5.33 (0.00)	0.18 (→0)	76.19 (0.00)	40.47 (→0)	0.21 (→0)	38.10 (→0)	27.42 (→0)
MGL	38.58 (5.65)	44.31 (11.74)	19.87 (4.37)	31.53 (16.69)	86.23 (11.82)	68.88 (10.76)	75.78 (17.81)	38.74 (2.55)	50.49 (10.17)
MCGL	51.55 (→0)	-3.38 (→0)	4.41 (0.00)	0.53 (→0)	98.80 (0.00)	83.13 (→0)	0.53 (→0)	4.01 (→0)	29.95 (→0)
GMC	49.74 (→0)	**44.31 (0.00)**	**43.05 (→0)**	**47.89 (→0)**	**99.58 (→0)**	**85.02 (→0)**	**95.54 (0.00)**	**42.80 (0.00)**	**63.57 (→0)**

TABLE 5
Clustering performance comparison in terms of F-measure on eight real-world data sets

F-measure (%)	100leaves	3sources	ALOI	BBC	Hdigit	Mfeat	NGs	WebKB	Average
S*K*-means	47.49 (1.63)	36.27 (2.35)	36.72 (3.76)	38.92 (2.21)	39.50 (2.73)	64.55 (5.33)	32.97 (2.76)	65.26 (4.52)	45.21 (2.85)
SNcut	31.17 (1.04)	34.61 (→0)	4.13 (0.20)	38.09 (→0)	35.88 (0.25)	73.60 (→0)	32.94 (0.00)	56.77 (0.00)	38.40 (1.85)
M*K*C	1.86 (0.00)	41.14 (10.88)	1.96 (→0)	50.18 (9.03)	72.02 (5.84)	51.30 (2.33)	38.17 (2.11)	55.05 (4.12)	38.96 (6.91)
MultiNMF	58.99 (1.58)	43.53 (2.68)	1.98 (→0)	39.22 (2.06)	60.00 (2.61)	–.– (–.–)	33.03 (0.00)	56.60 (→0)	36.67 (1.12)
CoregSC	**72.57 (1.90)**	47.75 (1.91)	40.51 (2.38)	48.79 (0.00)	69.38 (2.48)	69.34 (5.11)	31.97 (0.26)	54.64 (2.41)	54.37 (2.06)
MSC	68.21 (2.23)	40.87 (3.05)	33.66 (3.68)	58.77 (1.83)	61.59 (3.64)	71.29 (5.58)	26.49 (5.09)	45.99 (3.13)	50.86 (2.96)
ASMV	61.48 (→0)	35.28 (→0)	7.12 (→0)	37.81 (0.00)	78.91 (0.00)	48.52 (→0)	32.79 (0.00)	65.27 (→0)	45.90 (→0)
MGL	39.44(5.53)	59.66 (7.12)	21.12 (4.22)	54.02 (8.53)	87.70 (13.14)	72.38 (9.37)	81.56 (13.98)	67.87 (0.99)	60.47 (7.74)
MCGL	52.17 (0.00)	34.17 (0.00)	6.21 (→0)	37.62 (0.00)	98.92 (0.00)	84.93 (→0)	32.75 (→0)	55.28 (0.00)	50.26 (→0)
GMC	50.42 (→0)	**60.47 (0.00)**	**43.66 (→0)**	**63.33 (→0)**	**99.62 (→0)**	**86.58 (→0)**	**96.43 (→0)**	**69.33 (0.00)**	**71.15 (→0)**

5.4 Computational Efficiency

We now evaluate the computational efficiency of the proposed method. Running time and scalability test are used to measure the computational effiiency.

Running Time. We experimented with each method in the same computing environment and recorded the running

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2903810, IEEE Transactions on Knowledge and Data Engineering

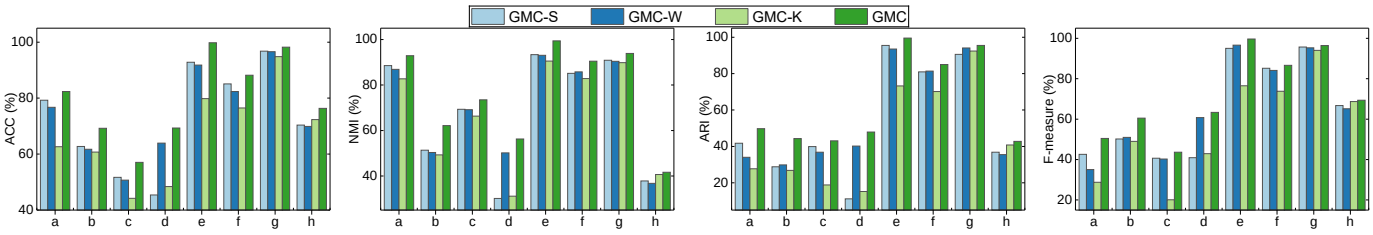


Fig. 4. Performance comparison of GMC and its variants on eight real-world data sets. The alphabetic lettere a-h in each figure denote the data sets 100leaves, 3sources, ALOI, BBC, Hdigit, Mfeat, NGs, and WebKB, respectively.

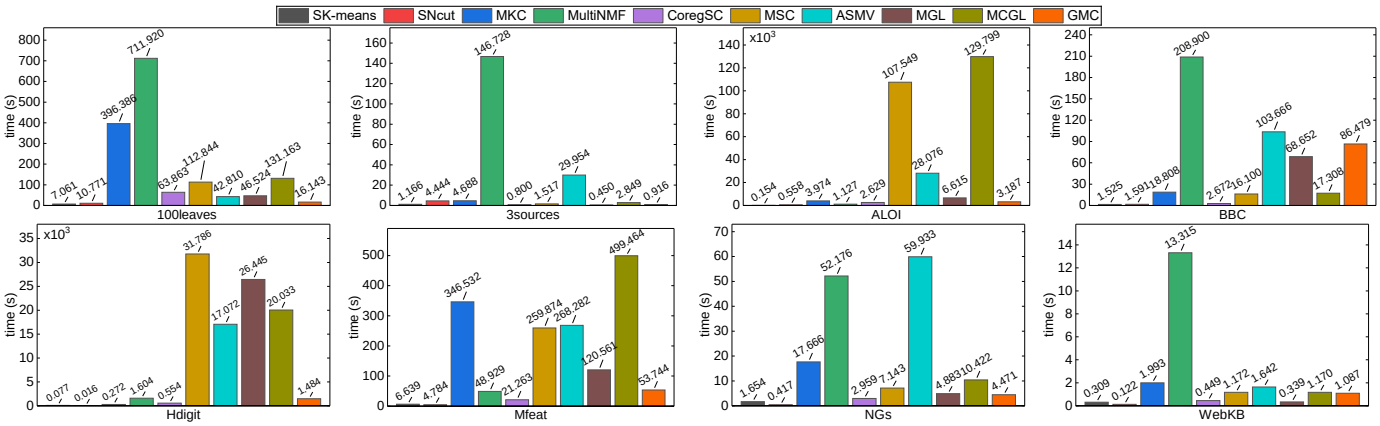


Fig. 5. Performance comparison of each method in terms of running time on eight real-world data sets.

time of each method on each data set. The results are shown in Fig. 5. From the figure, we can see that single-view methods often perform more efficiently than multi-view methods. The reason is clear because multi-view clustering methods need to handle multiple views simultaneously. For all multi-view clustering methods, our method GMC achieves the best performance in most cases. More precisely, GMC is superior to ASMV on all eight data sets, superior to MSC and MCGL on seven data sets, superior to MKC and MultiNMF on six data sets, superior to MGL on five data sets, and inferior to CoregSC on seven data sets of the eight data sets. The results show that the computational efficiency of the proposed GMC method is medium in comparison with the baselines. The reason is that although our method calculates matrix eigenvalues and eigenvectors, the whole algorithm converges very quickly, which will be clear in the next subsection.

Scalability Evaluation. To further evaluate the computational efficiency of our GMC, we perform a scalability test on two large-scale data sets (i.e., ALOI and Hdigit) respectively. We first randomly divide each data set into ten balanced subsets and then evaluate the performance of our GMC by feeding different number of subsets. We record the running time on each subset and perform a quadratic polynomial fit. The curve and results table for each data set are given in Fig. 6. From the figure, we see that the test results on each data set satisfy a quadratic polynomial distribution. Thus, we can conclude that the time complexity of our algorithm is around $\mathcal{O}(n^2)$ as discussed in Subsection 4.3.

5.5 Convergence Study

The proposed optimization algorithm for the objective function of GMC is iterative. We have proved its convergence

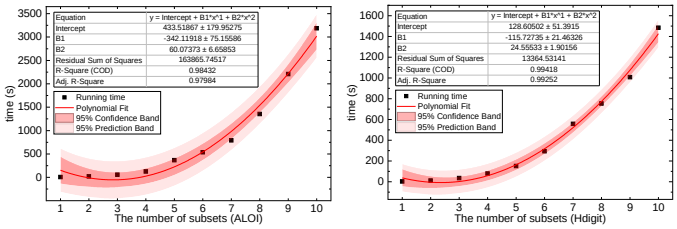


Fig. 6. Scalability evaluation on data sets ALOI and Hdigit.

property. Here we investigate how fast the proposed algorithm can converge.

Fig. 7 shows the convergence curves of GMC on all the eight data sets. For each figure, the x-axis and y-axis denote the iteration number and the value of the objective function, respectively. As can be seen, the proposed algorithm converges very quickly, usually within 10 iterations. The reason is that we provided an optimized solution for each subproblem.

6 Conclusions

This paper presented a novel method for multi-view clustering, called Graph-based Multi-view Clustering (GMC). GMC couples the learning of the similarity-induced graph (SIG) of each view, the learning of the unified graph of all views, and the clustering task into a joint framework. In particular, GMC automatically learns a unified fusion graph from the learned SIGs of all views. The learned unified graph can also help the learning of the SIG of each view. With the rank constraint on the graph Laplacian matrix, the number of connected components in the unified graph is equal to the required number of clusters. As a result, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2903810, IEEE Transactions on Knowledge and Data Engineering

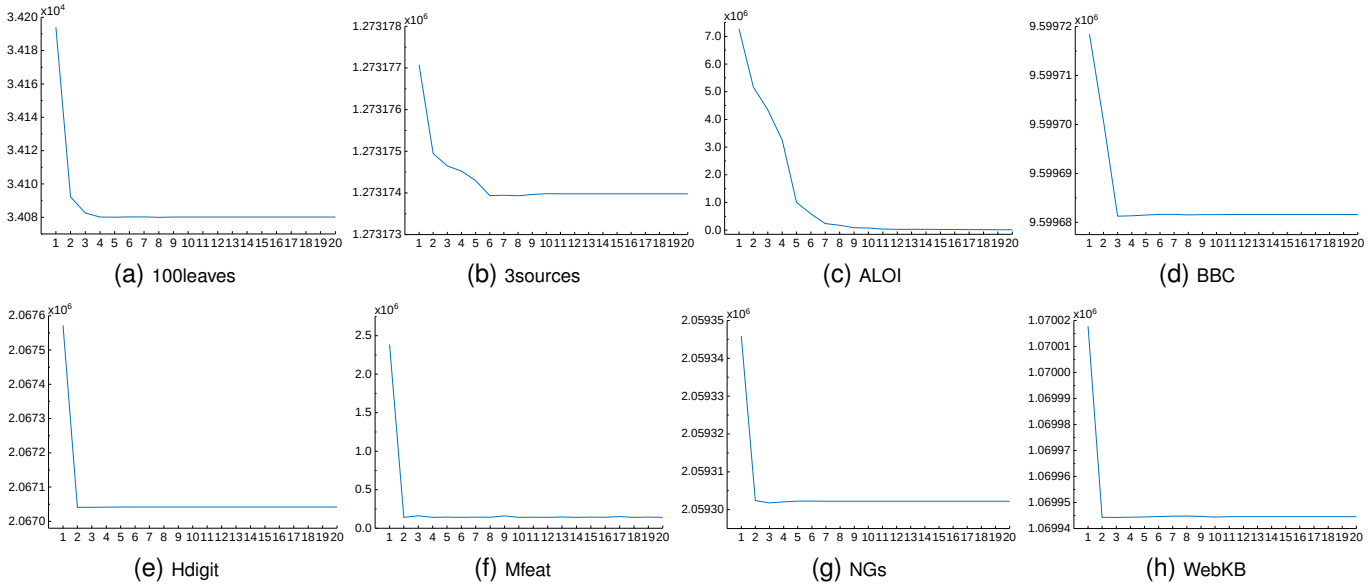


Fig. 7. Convergence curve of GMC over each data set.

clustering structure is uncovered at the same time as the unified graph is produced. Experiment results on two toy data sets and eight real-world data sets demonstrated the superior performance of the proposed GMC method, by comparing it with nine baselines. Our future work includes designing a more general framework that works in both unsupervised setting and semi-supervised setting. We are also interested in exploring techniques to speed up our method for large-scale data.

APPENDIX A
SOLUTION TO PROBLEM (14)

This appendix includes the detailed solution steps to problem (14).

Recall that the objection function of problem (14) is defined as follows:

$$\begin{aligned}\min_{\mathbf{S}^v} &\sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v-\mathbf{x}_j^v\right\|_2^2 s_{ij}^v+\beta\sum_{i}^{n}\|\mathbf{s}_i^v\|_2^2+w_v\left\|\mathbf{U}-\mathbf{S}^v\right\|_F^2 \\ &s.t.\ s_{ii}^v=0, s_{ij}^v\geq 0, \mathbf{1}^T\mathbf{s}_i^v=1.\end{aligned} \tag{38}$$

It is easy to find that problem (38) is independent for different i. Thus, we can solve the following problem separately for each $\mathbf{s}_i^v$:

$$\begin{aligned}\min_{\mathbf{s}_i^v} &\sum_{j=1}^{n} \left\|\mathbf{x}_i^v-\mathbf{x}_j^v\right\|_2^2 s_{ij}^v+\beta\|\mathbf{s}_i^v\|_2^2+w_v\left\|\mathbf{u}_i-\mathbf{s}_i^v\right\|_2^2 \\ &s.t.\ s_{ii}^v=0, s_{ij}^v\geq 0, \mathbf{1}^T\mathbf{s}_i^v=1.\end{aligned} \tag{39}$$

Now we define $e_{ij}=\|\mathbf{x}_i^v-\mathbf{x}_j^v\|$ and denote $\mathbf{e}_i$ as a vector with the j-th entry as e_{ij}. Then, Eq. (39) is simply written as

$$\begin{aligned}\min_{\mathbf{s}_i^v} &\frac{1}{2}\left\|\mathbf{s}_i^v+\frac{\mathbf{e}_i}{2\beta}\right\|_2^2+\frac{1}{2\beta}w_v\|\mathbf{u}_i-\mathbf{s}_i^v\|_2^2 \\ &s.t.\ s_{ii}^v=0, s_{ij}^v\geq 0, \mathbf{1}^T\mathbf{s}_i^v=1.\end{aligned} \tag{40}$$

The Lagrangian function of problem (40) with the constraints $s_{ij}^v\geq 0$ and $\mathbf{1}^T\mathbf{s}_i^v=1$ is defined as

$$\begin{aligned}\ell(\mathbf{s}_i^v,\eta,\xi)=&\frac{1}{2}\left\|\mathbf{s}_i^v+\frac{\mathbf{e}_i}{2\beta}\right\|_2^2+\frac{1}{2\beta}w_v\|\mathbf{u}_i-\mathbf{s}_i^v\|_2^2 \\ &-\eta(\mathbf{1}^T\mathbf{s}_i^v-1)-\xi^T\mathbf{s}_i^v\end{aligned} \tag{41}$$

where η is the Lagrangian coefficient scalar and ξ is the Lagrangian coefficient vector.

Taking the derivative with respect to $\mathbf{s}_i^v$ and setting it to zero, we can derive

$$\mathbf{s}_i^v+\frac{\mathbf{e}_i}{2\beta}-\frac{1}{\beta}w_v(\mathbf{u}_i-\mathbf{s}_i^v)-\eta\mathbf{1}-\xi=\mathbf{0}. \tag{42}$$

The j-th entry of $\mathbf{s}_i^v$ in Eq. (42) is shown below:

$$s_{ij}^v+\frac{e_{ij}}{2\beta}-\frac{1}{\beta}w_v(u_{ij}-s_{ij}^v)-\eta-\xi_j=0. \tag{43}$$

Note that $s_{ij}\xi_j=0$ according to KKT conditions. Then, we obtain the following solution (denoted as $\hat{s}_{ij}$) for s_{ij}:

$$\hat{s}_{ij}=\left(\frac{\frac{-e_{ij}}{2}+w_vu_{ij}+\beta\eta}{\beta+w_v}\right)_+. \tag{44}$$

Suppose $e_{i1},...,e_{in}$ are ordered from small to large. As we constrain $\mathbf{s}_i$ having k nonzero entries, we prefer $\hat{s}_{ik}>0$ and $\hat{s}_{i,k+1}=0$. Then, we arrive at

$$\frac{-e_{ik}}{2}+w_vu_{ik}+\beta\eta>0, \text{ and } \frac{-e_{i,k+1}}{2}+w_vu_{i,k+1}+\beta\eta\leq 0. \tag{45}$$

Besides, Eq. (44) and the constraint $\mathbf{1}^T\mathbf{s}_i^v=1$ give us

$$\eta=\frac{1}{k}\left(1+\frac{w_v}{\beta}+\sum_{h=1}^{k}\frac{e_{ih}}{2\beta}\right). \tag{46}$$

According Eq. (45) and Eq. (46), we have

$$\begin{cases}\beta>\frac{ke_{ik}-\sum_{h=1}^{k}e_{ih}-2kw_vu_{ik}-2w_v}{2}, \\ \beta\leq\frac{ke_{i,k+1}-\sum_{h=1}^{k}e_{ih}-2kw_vu_{i,k+1}-2w_v}{2}.\end{cases} \tag{47}$$

In order to constrain the optimal solution $\hat{\mathrm{s}}_i$ to have k nonzero entries, the parameter β is set to

$$\beta = \frac{ke_{i,k+1} - \sum_{h=1}^{k} e_{ih} - 2kw_v u_{i,k+1} - 2w_v}{2}. \tag{48}$$

According to Eq. (45), Eq. (46) and Eq. (48), the final solution for s_{ij}^v in $\mathbf{s}_i^v$ is shown below:

$$\hat{s}_{ij}^v = \begin{cases} \frac{e_{i,k+1} - e_{ij} + 2w_v u_{ij} - 2w_v u_{i,k+1}}{ke_{i,k+1} - \sum_{h=1}^{k} e_{ih} - 2kw_v u_{i,k+1} + 2\sum_{h=1}^{k} w_v u_{ih}} & j \leq k, \\ 0 & j > k. \end{cases} \tag{49}$$

APPENDIX B
INITIALIZE SIG MATRIX FOR EACH VIEW

This appendix includes the detailed solution steps to initialize SIG matrix for each view.

Recall that the objection function of problem (3) is defined as below:

$$\begin{aligned} \min_{\{\mathbf{S}^v\}} \sum_{v=1}^{m} \sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta \sum_{v=1}^{m} \sum_{i}^{n} \|\mathbf{s}_i^v\|_2^2 \\ s.t.\ \forall v, s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T \mathbf{s}_i^v = 1. \end{aligned} \tag{50}$$

As can be seen, the solution for each view is independent. Here we take $\mathbf{S}^v$ as an example, formulated as

$$\begin{aligned} \min_{\mathbf{S}^v} \sum_{i,j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta \sum_{i}^{n} \|\mathbf{s}_i^v\|_2^2 \\ s.t.\ s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T \mathbf{s}_i^v = 1. \end{aligned} \tag{51}$$

Similar to problem (38), problem (51) is independent for different i, so we solve each $\mathbf{s}_i^v$ separately:

$$\begin{aligned} \min_{\mathbf{s}_i^v} \sum_{j=1}^{n} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}^v + \beta \|\mathbf{s}_i^v\|_2^2 \\ s.t.\ s_{ii}^v = 0, \mathbf{s}_i^v \geq 0, \mathbf{1}^T \mathbf{s}_i^v = 1. \end{aligned} \tag{52}$$

Now we denote $\mathbf{b}_i$ as a vector with the j-th entry as b_{ij} defined by $b_{ij} = \|\mathbf{x}_i^v - \mathbf{x}_j^v\|_2^2$, then initializing $\mathbf{s}_i^v$ in $\mathbf{S}^v$ is formulated as follows:

$$\begin{aligned} \min_{\mathbf{s}_i^v} \frac{1}{2} \left\|\mathbf{s}_i^v + \frac{\mathbf{b}_i}{2\beta}\right\|_2^2 \\ s.t.\ s_{ii}^v = 0, \mathbf{s}_i^v \geq 0, \mathbf{1}^T \mathbf{s}_i^v = 1. \end{aligned} \tag{53}$$

We can follow the same process in solving problem (40) to solve problem (53). Now we give the final solution for s_{ij}^v in $\mathbf{s}_i^v$ directly:

$$\hat{s}_{ij}^v = \begin{cases} \frac{b_{i,k+1} - b_{ij}}{kb_{i,k+1} - \sum_{h=1}^{k} b_{ih}} & j \leq k, \\ 0 & j > k. \end{cases} \tag{54}$$

ACKNOWLEDGMENTS

We would like to thank the authors of the baseline systems for their codes. We are especially grateful to the anonymous reviewers and editor(s) for their comments and suggestions. Hao Wang and Yan Yang's work was supported in part by grants from the National Natural Science Foundation of China (No. 61572407), and the China Scholarship Council (No. 201707000064). Bing Liu's work was supported in part by grants from National Science Foundation (NSF) under grant nos. IIS-1407927 and IIS-1838770, and a research gift from Huawei Technologies Co. Ltd.

REFERENCES

[1] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *CoRR*, vol. abs/1304.5634, 2013.
[2] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, 2017.
[3] G. Chao, S. Sun, and J. Bi, "A survey on multi-view clustering," *CoRR*, vol. abs/1712.06246, 2017.
[4] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Mining and Anal.*, vol. 1, no. 2, pp. 83–107, 2018.
[5] M. Saha, "A graph based approach to multiview clustering," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, 2013, pp. 128–133.
[6] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.
[7] C. Hou, F. Nie, H. Tao, and D. Yi, "Multi-view unsupervised feature selection with adaptive similarity and view weight," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1998–2011, 2017.
[8] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1501–1511, 2018.
[9] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2564–2570.
[10] W. Zhuge, F. Nie, C. Hou, and D. Yi, "Unsupervised single and multiple views feature extraction with structured graph," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2347–2359, 2017.
[11] K. Zhan, X. Chang, J. Guan, L. Chen, Z. Ma, and Y. Yang, "Adaptive structure discovery for multimedia analysis using multiple features," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–9, 2018. [Online]. Available: https://doi.org/10.1109/TCYB.2018.2815012
[12] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–9, 2018. [Online]. Available: https://doi.org/10.1109/TCYB.2017.2751646
[13] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh, "Sample selection bias correction theory," in *Proc. Int. Conf. Algo. Learn. Theory*, 2008, pp. 38–53.
[14] S. Sun, J. Shawe-Taylor, and L. Mao, "PAC-Bayes analysis of multi-view learning," *Inf. Fusion*, vol. 35, pp. 117–131, 2017.
[15] A. Serra, D. Greco, and R. Tagliaferri, "Impact of different metrics on multi-view clustering," in *Proc. Int. Joint Conf. on Neural Netw.*, 2015, pp. 1–8.
[16] Y. Wang, X. Lin, and Q. Zhang, "Towards metric fusion on multi-view data: A cross-view based graph random walk approach," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 805–810.
[17] Y. Wang, W. Zhang, L. Wu, X. Lin, and X. Zhao, "Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 57–70, 2017.
[18] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Trans. Syst., Man, Cybern. B*, vol. 40, no. 6, pp. 1438–1446, 2010.
[19] A. Kumar, P. Rai, and H. D. III, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.
[20] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.
[21] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2833–2843, 2016.
[22] L. Feng, L. Cai, Y. Liu, and S. Liu, "Multi-view spectral clustering via robust local subspace learning," *Soft Comput.*, vol. 21, no. 8, pp. 1937–1948, 2017.
[23] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, "Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2153–2159.
[24] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–11, 2018. [Online]. Available: https://doi.org/10.1109/TNNLS.2017.2777489

[25] Y. Wang and L. Wu, "Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering," *Neural Netw.*, vol. 103, pp. 1–8, 2018.

[26] A. Appice and D. Malerba, "A co-training strategy for multiple view clustering in process mining," *IEEE Trans. Services Comput.*, vol. 9, no. 6, pp. 832–845, 2016.

[27] J. Sun, J. Lu, T. Xu, and J. Bi, "Multi-view sparse co-clustering via proximal alternating linearized minimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 757–766.

[28] S. F. Hussain and S. Bashir, "Co-clustering of multi-view datasets," *Knowl. Inf. Syst.*, no. 3, pp. 1–26, 2016.

[29] V. R. de Sa, P. W. Gallagher, J. M. Lewis, and V. L. Malave, "Multi-view kernel construction," *Mach. Learn.*, vol. 79, no. 1, pp. 47–71, 2010.

[30] Y. Lu, L. Wang, J. Lu, J. Yang, and C. Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognit.*, vol. 47, no. 11, pp. 3656–3664, 2014.

[31] G. Tzortzis and A. Likas, "Kernel-based weighted multi-view clustering," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 675–684.

[32] Y. Wang, X. Liu, Y. Dou, and R. Li, "Multiple kernel clustering framework with improved kernels," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2999–3005.

[33] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4238–4246.

[34] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus." *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3939–3949, 2015.

[35] Q. Yin, S. Wu, and L. Wang, "Unified subspace learning for incomplete and unlabeled multi-view data," *Pattern Recognit.*, vol. 67, pp. 313–327, 2017.

[36] C. Xu, D. Tao, and C. Xu, "Multi-view learning with incomplete views," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5812–5825, 2015.

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

[38] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2921–2927.

[39] Z. Guan, L. Zhang, J. Peng, and J. Fan, "Multi-view concept learning for data representation," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3016–3028, 2015.

[40] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4333–4341.

[41] X. Cai, F. Nie, and H. Huang, "Multi-view K-means clustering on big data," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2598–2604.

[42] J. Wang, F. Tian, H. Yu, C. H. Liu, K. Zhan, and X. Wang, "Diverse non-negative matrix factorization for multiview data representation," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2620–2632, 2018.

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

[44] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. Annual Conf. Comput. Learn. Theory*, 1998, pp. 92–100.

[45] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 709–722, 2013.

[46] C. Xu, D. Tao, and C. Xu, "Large-margin multi-view information bottleneck," *IEEE Trans. on Pattern Anal. and Mach. Inte.*, vol. 36, no. 8, pp. 1559–1572, 2014.

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

[48] F. Wu, X.-Y. Jing, X. You, D. Yue, R. Hu, and J.-Y. Yang, "Multi-view low-rank dictionary learning for image classification," *Pattern Recognit.*, vol. 50, pp. 143–154, 2016.

[49] J. Li, C. Xu, W. Yang, C. Sun, and D. Tao, "Discriminative multi-view interactive image re-ranking," *IEEE Trans. on Image Process.*, vol. 26, no. 7, pp. 3113–3127, 2017.

[50] Y. Zhao, X. You, S. Yu, C. Xu, W. Yuan, X.-Y. Jing, T. Zhang, and D. Tao, "Multi-view manifold learning with locality alignment," *Pattern Recognit.*, vol. 78, pp. 154–166, 2018.

[51] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[52] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.

[53] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The Laplacian spectrum of graphs," *Graph Theory, Comb., and Appl.*, vol. 2, no. 12, pp. 871–898, 1991.

[54] F. R. Chung, *Spectral graph theory*. American Math. Soc., 1997, vol. 92.

[55] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations I," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 35, no. 11, pp. 652–655, 1949.

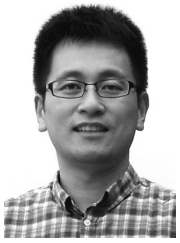
[56] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic Press, 2014.

[57] X. Zhang and Q. You, "Clusterability analysis and incremental sampling for nyström extension based spectral clustering," in *Proc. Int. Conf. Data Mining*, 2011, pp. 942–951.

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

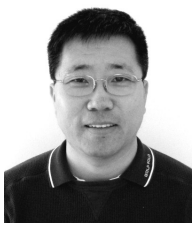
[59] F. Nie, H. Huang, X. Cai, and C. H. Q. Ding, "Efficient and robust feature selection via joint $l_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[60] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.



Hao Wang received the B.Eng. degree from Nanyang Institute of Technology, Nanyang, China, in 2014. He is currently working toward the Ph.D. degree at the Southwest Jiaotong University, Chengdu, China, and is visiting the University of Illinois at Chicago from 2017 to 2019. His current research interests include data mining, multi-view learning, natural language processing, and lifelong machine learning.

Yan Yang received the B.S., and M.S., degrees from Huazhong University of Science and Technology, Wuhan, China, in 1984 and 1987, respectively. She received her Ph.D. degree from Southwest Jiaotong University, Chengdu, China, in 2007. She was a visiting scholar at the University of Waterloo, Waterloo, Canada, from 2002 to 2003 and 2004 to 2005. She is currently a Professor and Vice Dean of the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. Her current research interests include big data analysis and mining, multi-view learning, ensemble learning, and semi-supervised learning. She is a member of the IEEE.



Bing Liu (F'14) received his Ph.D. degree in Artificial Intelligence from the University of Edinburgh. He is currently a Distinguished Professor of Computer Science at the University of Illinois at Chicago. Before joining UIC, he was an associate professor at the School of Computing, National University of Singapore. He has authored four books, and published extensively in top conferences and journals. Two of his papers received 10-year Test-of-Time awards from KDD. He has served as the Program Committee Chair of KDD, ICDM, SDM, CIKM, WSDM, and PAKDD conferences, as associate editors of several leading data mining journals, e.g., TKDE, TWEB, TKDD, DMKD, and as area/track chairs or senior program committee members of numerous NLP, AI, data mining, and Web technology conferences. He also served as the Chair of ACM SIGKDD from 7/1/2013 to 6/30/2017. He is a fellow of the ACM, AAAI, and IEEE.