Graph Learning for Multiview Clustering

Kun Zhan, Changqing Zhang, Junpeng Guan, and Junsheng Wang

***Abstract*—Most existing graph-based clustering methods need a predefined graph and their clustering performance highly depends on the quality of the graph. Aiming to improve the multiview clustering performance, a graph learning-based method is proposed to improve the quality of the graph. Initial graphs are learned from data points of different views, and the initial graphs are further optimized with a rank constraint on the Laplacian matrix. Then, these optimized graphs are integrated into a global graph with a well-designed optimization procedure. The global graph is learned by the optimization procedure with the same rank constraint on its Laplacian matrix. Because of the rank constraint, the cluster indicators are obtained directly by the global graph without performing any graph cut technique and the *k*-means clustering. Experiments are conducted on several benchmark datasets to verify the effectiveness and superiority of the proposed graph learning-based multiview clustering algorithm comparing to the state-of-the-art methods.**

***Index Terms*—Clustering, feature learning, multiview clustering, unsupervised learning.**

I. Introduction

IT IS reasonable and appropriate that an object is represented with various features from multiple views, and usually these different features are complementary of each other. Multiview feature learning can integrate all these features and exploit correlations between views to obtain more refined and higher-level information.Therefore, effectively integrating heterogeneous features from different views to improve the clustering performance is an important topic.

The data structures are usually characterized in the form of the graph. Most existing graph-based clustering methods separate the data clustering with graph construction. Besides with Gaussian function, there are several graph construction methods, such as local linear similarity graph [1], *k*-nearest neighbor graph [2], [3], local discriminant graph [4], [5], pairwise similarity graph [6], and graph learned with subspace clustering [7]. In these methods, the graph construction is independent of the clustering and their performance highly relies on the predefined graph. Recently, some adaptive graph learning methods are proposed by using a rank constraint on the Laplacian matrix for obtaining the cluster indicators directly [8]–[12]. However, there are few graph learning methods for multiview clustering.

In this paper, the cluster indicators are obtained by a learned global graph without performing the graph-cut techniques and the *k*-means clustering algorithms as shown in Fig. 1. A rank constraint on the Laplacian matrix renders the learned graph to achieve an ideal neighbors assignment so that the number of components of the graph is exact the number of clusters and each component corresponds to one cluster. The initial graphs are learned from multiview data, and these graphs are further optimized with a novel and well-designed optimization problem. Then, the optimized graphs are integrated into a global graph. The proposed method jointly optimizes the graph matrices to make use of the data correlation between views.

Since the proposed graph learning-based method better captures the graph structure of the data space, much better quantitative result fully demonstrates the superiority on several real-world datasets for the multiview clustering problem.

The remainder of this paper is organized as follows. We review some related work in Section II. In Section III, we propose two objective functions, the first one is proposed for optimizing each single view graph, and the second one is applied to integrating them into a global graph. The graphs are constrained by the rank of Laplacian matrix. In Section IV, we propose a novel algorithm to optimize the well-designed objective functions in Section III. In Section V, numerical experiments are conducted. We use four real datasets and compare with seven state-of-the-art methods. Section VI concludes with some discussion.

II. Related Work

Numerous multiview clustering methods are proposed recently. At the beginning, co-training learning model is applied to fuse the multiview features [13]–[15], and the reasons for the success of co-training methods have been investigated by Balcan *et al.* [16] and Wang and Zhou [17]. A drawback of the co-training is that it is not robust against outlier features that result in error diffusion. These earlier co-training methods are semisupervised learning algorithms. Kumar *et al.* [18], [19] proposed unsupervised multiview clustering methods based on co-training learning, such as co-training multiview spectral clustering (SC) and co-regularized multiview SC (CRSC). The two methods use predefined Laplacian matrices for each view, and they require the graph construction beforehand and separate the data clustering with

Manuscript received June 20, 2017; accepted September 11, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61201422, in part by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20120211120013, and in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2017-190. This paper was recommended by Associate Editor R. Tagliaferri. *(Corresponding author: Kun Zhan.)*

K. Zhan, J. Guan, and J. Wang are with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China (e-mail: kzhan@lzu.edu.cn).

C. Zhang is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2751646

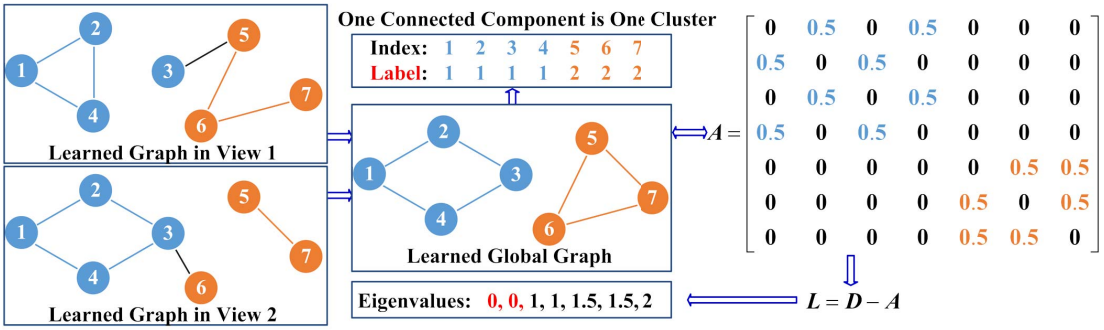


Fig. 1. Schematic of the graph learning.

graph construction, which renders clustering accuracy highly depend on the quality of the input graphs.

As same as the two methods, other SC-based multiview clustering methods have the same problems, such as methods including multimodal SC [20], robust multiview SC (RMSC) [21], multiview SC (MVSC) [22], SC with brainstorming [23], etc. SC is one of the most popular clustering approaches due to taking advantage of the well-defined mathematical framework [24]–[27]. Because the data graph and manifold information are utilized in these graph-based clustering methods, they show better performance than multiview k-means clustering [28]. However, the results of most graph-based MVSC methods are sensitive to the input graph. What is more, the graph does not explicitly represent the clustering structures so that graph-cut-based methods need a post-process k-means to obtain the cluster indicators. Multiview subspace clustering methods also need post-process k-means clustering algorithms because they finally use SC methods to obtain the cluster indicators [29]–[32].

Most multiview clustering methods are graph-based and they need to perform graph-cut techniques and k-means in the learned data space to obtain the clustering indicators.

III. Graph Learning

If all the elements in the similarity matrix $S \in \mathbb{R}^{n\times n}$ are nonnegative, its Laplacian matrix L has the property [33], [34].

Theorem 1: The multiplicity c of 0 as an eigenvalue of L is equal to the number of components of S.

Theorem 1 indicates that if a constraint condition $\text{rank}(L) = n - c$ is satisfied, then S is an ideal neighbors assignment and the data points are already partitioned into c cluster [33]–[35].

The constraint condition $\text{rank}(L) = n - c$ can be satisfied when the sum of the top c smallest eigenvalues of L is zero, i.e., $\sum_{i=1}^{c}\lambda_i = 0$ and λ_i denotes the ith smallest eigenvalue of L. Then, according to Fan's [36] theorem, we have

$$\sum_{i=1}^{c}\lambda_i = \min_{Q} \quad \text{Tr}\left(Q^T L Q\right) \quad \text{s.t.} \quad Q \in \mathbb{R}^{n\times c}, Q^T Q = I \tag{1}$$

where $Q^T = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n]$, $\text{Tr}(\cdot)$ denotes the trace operator, $L = D - [(S^T + S)/2]$ is the Laplacian matrix, $I \in \mathbb{R}^{c\times c}$ is an identity matrix, and D is a diagonal matrix and its elements are column sums of $[(S^T + S)/2]$.

The proof of Fan's theorem can be seen in [37] and [38]. The left term of (1) can be solved with respect to Q by the eigenvectors of L, but it has trivial solution with respect to S, i.e., all elements are assigned to zeros. So we add ℓ_2-norm regularization to smooth the elements of S and add a constraint that the sum of each column of S is one. Then, using the following objective function (2) optimizes S and Q simultaneously:

$$\begin{aligned} \min_{S,Q} \quad & \text{Tr}(Q^T L Q) + \beta\|S\|_{\text{F}}^2 \\ \text{s.t.} \quad & \forall j,\ \mathbf{1}^T \boldsymbol{s}_j = 1,\ \boldsymbol{s}_j \geq 0 \\ & Q \in \mathbb{R}^{n\times c},\ Q^T Q = I \end{aligned} \tag{2}$$

where β is the regularization parameter, and $\boldsymbol{s}_j$ is the jth column of S.

We propose a novel algorithm to optimize the objective function (2) in Section IV-A. According to Theorem 1, the convergence condition of (2) is determined by the eigenvalues of L and can be reached when the sum of the top c smallest eigenvalues of L is zero. The objective function (2) is different from the one proposed in [8]. The objective function (2) has only one regularization parameter and the raw data points are not involved. In [8], the raw data points are involved in their objective function, which may suffer from the curse of dimensionality.

We obtain each view graph $S^{(v)}$ by (2), and the different graphs are integrated into a global graph A by

$$\begin{aligned} \min_{\boldsymbol{a}_j, w_j^{(v)}} \quad & \sum_{i,j=1}^{n}\left\|\boldsymbol{a}_j - \sum_{v=1}^{n_v} w_j^{(v)}\boldsymbol{s}_j^{(v)}\right\|_2^2 \\ \text{s.t.} \quad & \forall j,\ \mathbf{1}^T\boldsymbol{a}_j = 1,\ \boldsymbol{a}_j \geq 0 \\ & \sum_{v=1}^{n_v} w_j^{(v)} = 1,\ \text{rank}(L_a) = n - c \end{aligned} \tag{3}$$

where $\boldsymbol{a}_j$ is the jth column of A, a_{ij} is the (i,j)th element of A, $L_a = D_a - [(A^T + A)/2]$ is the global Laplacian matrix, and D_a is a diagonal matrix and its elements are column sums of the matrix $[(A^T + A)/2]$.

In (3), we use the constraint condition $\text{rank}(L_a) = n - c$ so that the global graph A is optimized well in the light of Theorem 1. The novelty lies in the graph learning for multiview clustering to exploit construction of each single view graph and to integrate these graphs into a global graph. Fig. 1

shows the schematic diagram of graph learning, and it can be seen from Fig. 1 that the number of components of the optimal global graph is exact the cluster number. However, k-means hardly achieves the ideal structure. Equation (3) uses the rank constraint on the Laplacian matrix in a well-designed optimization problem. In Section IV-B, we propose a novel algorithm to optimize the objective function (3) and solve this hard optimization problem with alternating optimization.

IV. Optimization

A. Single View Graph Learning

The first stage is optimizing the each single view graph $S^{(v)}$ independently by (2). We divide the problem (2) into two subproblems and alteratively solve them.

The first subproblem is to fix Q, updating s_{ij}. Then, (2) becomes

$$\begin{aligned} \min_{S} \quad & \sum_{i,j=1}^{n} \left\|\boldsymbol{q}_i - \boldsymbol{q}_j\right\|_2^2 s_{ij} + \beta \|S\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & \forall j,\ \mathbf{1}^T \boldsymbol{s}_j = 1, \boldsymbol{s}_j \geq 0 \end{aligned} \tag{4}$$

where s_{ij} is the (i,j)th element of S.

Each column of S is independent, so solving (4) is equal to optimizing the following problem:

$$\begin{aligned} \min_{s_{ij}} \quad & \sum_{i=1}^{n} \left\|\boldsymbol{q}_i - \boldsymbol{q}_j\right\|_2^2 s_{ij} + \beta \sum_{i=1}^{n} s_{ij}^2 \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{s}_j = 1\,,\ \boldsymbol{s}_j \geq 0. \end{aligned} \tag{5}$$

$\|\boldsymbol{q}_i - \boldsymbol{q}_j\|_2^2$ is denoted by g_{ij}, then (5) becomes

$$\begin{aligned} \min_{s_{ij}} \quad & \sum_{i=1}^{n} g_{ij} s_{ij} + \beta \sum_{i=1}^{n} s_{ij}^2 \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{s}_j = 1\,,\ \boldsymbol{s}_j \geq 0. \end{aligned} \tag{6}$$

Solving (6) is equal to optimizing the following problem:

$$\begin{aligned} \min_{\boldsymbol{s}_j} \quad & \left\|\boldsymbol{s}_j + \frac{1}{2\beta}\boldsymbol{g}_j\right\|_2^2 \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{s}_j = 1\,, \boldsymbol{s}_j \geq 0 \end{aligned} \tag{7}$$

where $\boldsymbol{g}_j$ denotes $[g_{1j}, g_{2j}, \ldots, g_{nj}]^T$.

Then, the Lagrangian function of (7) is

$$\mathcal{L}(\boldsymbol{s}_j, \eta, \boldsymbol{\rho}) = \left\|\boldsymbol{s}_j + \frac{1}{2\beta}\boldsymbol{g}_j\right\|_2^2 - \eta\left(\boldsymbol{s}_j^T \mathbf{1} - 1\right) - \boldsymbol{\rho}^T \boldsymbol{s}_j \tag{8}$$

where η and $\boldsymbol{\rho}$ are the Lagrangian multipliers.

According to the Karush–Kuhn–Tucker condition [39], it can be verified that the optimal solution $\boldsymbol{s}_j$ is

$$\boldsymbol{s}_j = \left(-\frac{\boldsymbol{g}_j}{2\beta} + \eta\right)_+. \tag{9}$$

The second subproblem is to fix S, updating Q. Then, (2) becomes

$$\begin{aligned} \min_{Q} \quad & \mathrm{Tr}\left(Q^T L Q\right) \\ \text{s.t.} \quad & Q \in \mathbb{R}^{n\times c},\ Q^T Q = I. \end{aligned} \tag{10}$$

Algorithm 1 Learning Each Single View Graph $S^{(v)}$, $\forall v$

1: **Input:** Dataset $\mathcal{X} = \left\{X^{(1)}, X^{(2)}, \ldots, X^{(n_v)}\right\}$ from n_v views and the cluster number c.
2: **Output:** Different view graph $S^{(v)}$, $v \in [1, n_v]$.
3: **Initialize:** Each view graph $S^{(v)}$ is initialized with (4) by substituting $\boldsymbol{x}$ into $\boldsymbol{q}$, and using each initial graph $S^{(v)}$ obtains initial $Q^{(v)}$ by (10).
4: **for** $v \in [1, n_v]$ **do**
5: **repeat**
6: Update $\boldsymbol{s}_j^{(v)}$ by using (9).
7: Update $Q^{(v)}$ by using (10).
8: **until** S has c connected components.
9: **end for**

Equation (10) can be solved by calculating the eigenvectors of L.

We alteratively optimize (9) and (10) until the sum of the top c smallest eigenvalues of L is zero. The algorithm for solving (2) is summarized by Algorithm 1. Let $X^{(v)} \in \mathbb{R}^{d^{(v)}\times n}$ denote the feature matrix in vth view. A feature matrix $X^{(v)}$ has n data points and $d^{(v)}$-dimensional features. $\boldsymbol{x}_i^{(v)} \in \mathbb{R}^{d^{(v)}\times 1}$ denotes a data point in vth view. There are n_v view number in a dataset.

B. Global Graph Learning

Using these different views graphs $S^{(v)}$ ($v \in [1, n_v]$) learned by (2), we integrate them into a global graph A by (3). According to Fan's theorem, (3) can be rewritten by

$$\begin{aligned} \min_{\boldsymbol{a}_j, w_j^{(v)}, P} \quad & \sum_{i,j=1}^{n} \left\|\boldsymbol{a}_j - \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)}\right\|_2^2 + \gamma\, \mathrm{Tr}\left(P^T L_a P\right) \\ \text{s.t.} \quad & \forall j\,,\ \mathbf{1}^T \boldsymbol{a}_j = 1\,,\ \boldsymbol{a}_j \geq 0 \\ & \sum_{v=1}^{n_v} w_j^{(v)} = 1\,,\ P \in \mathbb{R}^{n\times c}\,,\ P^T P = I \end{aligned} \tag{11}$$

where γ the tradeoff parameter.

There are three variables in the problem (11), so we divide (11) into three subproblems.

The first subproblem is to fix P and $w_j^{(v)}$, updating $\boldsymbol{a}_j$. Then (11) becomes

$$\begin{aligned} \min_{\boldsymbol{a}_j} \quad & \sum_{i,j=1}^{n} \left\|\boldsymbol{a}_j - \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)}\right\|_2^2 + \gamma \sum_{i,j=1}^{n} \left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|_2^2 a_{ij} \\ \text{s.t.} \quad & \forall j\,,\ \mathbf{1}^T \boldsymbol{a}_j = 1\,, \boldsymbol{a}_j \geq 0. \end{aligned} \tag{12}$$

Different columns of A are independent, so we can solve each column separately

$$\begin{aligned} \min_{\boldsymbol{a}_j} \quad & \sum_{j=1}^{n} \left\|\boldsymbol{a}_j - \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)}\right\|_2^2 + \gamma \sum_{j=1}^{n} \left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|_2^2 a_{ij} \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{a}_j = 1\,,\ \boldsymbol{a}_j \geq 0. \end{aligned} \tag{13}$$

Let h_{ij} denote $\|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2^2$, then solving (13) is equal to optimizing the following problem:

$$\begin{aligned} \min_{\boldsymbol{a}_j} \quad & \left\| \boldsymbol{a}_j + \left(\frac{\gamma}{2}\boldsymbol{h}_j - \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)} \right) \right\|_2^2 \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{a}_j = 1\,,\ \boldsymbol{a}_j \geq 0. \end{aligned} \tag{14}$$

The solution of (14) is the same to solving (7).

The second subproblem is to fix $\boldsymbol{a}_j$ and $w_j^{(v)}$, updating P. Then (11) becomes

$$\begin{aligned} \min_{P} \quad & \mathrm{Tr}\left(P^T L_a P\right) \\ \text{s.t.} \quad & P \in \mathbb{R}^{n\times c}\,,\ P^T P = I. \end{aligned} \tag{15}$$

The solution of (15) is as same as solving (10).

The third subproblem is to fix $\boldsymbol{a}_j$ and P, updating $w_j^{(v)}$. Then, (11) becomes

$$\begin{aligned} \min_{w_j^{(v)}} \quad & \sum_{i,j=1}^{n} \left\| \boldsymbol{a}_j - \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)} \right\|_2^2 \\ \text{s.t.} \quad & \sum_{v=1}^{n_v} w_j^{(v)} = 1. \end{aligned} \tag{16}$$

We optimize each i independently for (16), so solving (16) is equal to optimizing the following problem:

$$\begin{aligned} \min_{w_j^{(v)}} \quad & \sum_{j=1}^{n} \left\| \boldsymbol{a}_j - \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)} \right\|_2^2 \\ \text{s.t.} \quad & \sum_{v=1}^{n_v} w_j^{(v)} = 1. \end{aligned} \tag{17}$$

Denoting $\boldsymbol{z}_j^{(v)} = \boldsymbol{a}_j - \boldsymbol{s}_j^{(v)}$ and $Z_j = [\boldsymbol{z}_j^{(1)}, \boldsymbol{z}_j^{(2)}, \ldots, \boldsymbol{z}_j^{(n_v)}]$, then

$$\begin{aligned} \sum_{j=1}^{n} \left\| \boldsymbol{a}_j - \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)} \right\|_2^2 &= \sum_{j=1}^{n} \left\| \sum_{v=1}^{n_v} w_j^{(v)} \left(\boldsymbol{a}_j - \boldsymbol{s}_j^{(v)} \right) \right\|_2^2 \\ &= \sum_{j=1}^{n} \left\| \sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{z}_j^{(v)} \right\|_2^2 \\ &= \sum_{j=1}^{n} \left\| Z_j \boldsymbol{w}_j \right\|_2^2 \\ &= \sum_{j=1}^{n} \boldsymbol{w}_j^T Z_j^T Z_j \boldsymbol{w}_j \end{aligned} \tag{18}$$

where $\boldsymbol{w}_j = [w_j^{(1)}, w_j^{(2)}, \ldots, w_j^{(n_v)}]^T$.

Then, the Lagrangian function of (17) is given by

$$\mathcal{L}\left(\boldsymbol{w}_j, \phi\right) = \sum_{j=1}^{n} \boldsymbol{w}_j^T Z_j^T Z_j \boldsymbol{w}_j + \phi\left(1 - \boldsymbol{w}_j^T \mathbf{1}\right) \tag{19}$$

where ϕ is the Lagrangian multiplier.

By setting the derivative of (19) with respect to $\boldsymbol{w}_j$ to zero, we have

$$\frac{\partial \mathcal{L}\left(\boldsymbol{w}_j, \phi\right)}{\partial \boldsymbol{w}_j} = Z_j^T Z_j \boldsymbol{w}_j - \phi \mathbf{1} = 0. \tag{20}$$

Algorithm 2 Learning the Global Graph A

1: **Input:** Graph set $\mathcal{S} = \left\{S^{(1)}, S^{(2)}, \ldots, S^{(n_v)}\right\}$ from n_v view and the cluster number c.
2: **Output:** A global graph A.
3: **Initialize:** Each element of $\boldsymbol{w}_j, \forall j$, is set to $\frac{1}{n_v}$, and weighted-sum rule $\sum_{v=1}^{n_v} w_j^{(v)} \boldsymbol{s}_j^{(v)}$ is used to obtain a graph A_0. A Laplacian matrix L_0 is calculated by A_0 and P is initialized with (15) by substituting L_0 into L_a.
4: **while** not converge **do**
5: **repeat**
6: Update $\boldsymbol{a}_j$ by solving (14).
7: Update P by solving (15).
8: **until** A has c connected components.
9: Update $\boldsymbol{w}_j$ by (21).
10: **end while**

Because of the constraint $\boldsymbol{w}_j^T \mathbf{1} = 1$, it is straightforward to check that the solution of (17) with respect to $\boldsymbol{w}_j$ is given by

$$\boldsymbol{w}_j = \frac{\left(Z_j^T Z_j\right)^{-1} \mathbf{1}}{\mathbf{1}^T \left(Z_j^T Z_j\right)^{-1} \mathbf{1}}. \tag{21}$$

We solve (14) as same as solving (7), and solve (15) as same as solving (10). Because of Fan's theorem, $\boldsymbol{a}_j$ and P are optimized together, while $\boldsymbol{w}_j$ is independent with them, we alteratively optimize (14) and (15) until the sum of the top c smallest eigenvalues of L_a is zero. Then, we calculate $\boldsymbol{w}_j$ by (21), and then optimize (14) and (15) again. We repeat the optimization procedure until the objective function (3) converges.

Based on the above analysis of the optimization problem (3), the overall algorithm for solving (3) is summarized in Algorithm 2.

C. Convergence Analysis

The convergence of Algorithm 1 is the same to Algorithm 2 because the top two steps of them are similar. The convergence of Algorithm 2 is given by Theorem 2.

Theorem 2: The alternate updating rules in Algorithm 2 monotonically decrease the objective function value of (3) in each iteration until convergence.

Proof: To fix others and updating $\boldsymbol{a}_j$, it is straightforward to check that (14) is a convex function [39] because the second order derivative of (14) with respect a_{ij} is equal to 1. Then, the overall objective function (3) $\mathcal{O}(\boldsymbol{a}_j, P, \boldsymbol{w}_j)$ decreases monotonically

$$\mathcal{O}\left(\left(\boldsymbol{a}_j\right)^{t+1}, P, \boldsymbol{w}_j\right) \leq \mathcal{O}\left(\left(\boldsymbol{a}_j\right)^{t}, P, \boldsymbol{w}_j\right) \tag{22}$$

where t denotes the iterative time.

The Hessian matrix of Lagrangian function of (15) is positive semidefinite according to [33], so objective function (15) is a convex problem. Then, to fix others and updating P, we can obtain the following inequality:

$$\mathcal{O}\left(\boldsymbol{a}_j, P^{t+1}, \boldsymbol{w}_j\right) \leq \mathcal{O}\left(\boldsymbol{a}_j, P^{t}, \boldsymbol{w}_j\right). \tag{23}$$

Because P is updated by $P^{t+1} = \arg\min_{P^TP=I}\mathrm{Tr}(P^TL_AP)$ with fixed A and $\boldsymbol{w}_j$.

The Hessian matrix of (19) is

$$\frac{\partial^2\mathcal{L}(\boldsymbol{w}_j,\phi)}{\partial(\boldsymbol{w}_j)^2} = 2Z_j^TZ_j. \tag{24}$$

It can be seen from (18) that $\boldsymbol{w}_j^TZ_j^TZ_j\boldsymbol{w}_j = \|Z_j\boldsymbol{w}_j\|_2^2 \geq 0$, then the Hessian matrix (24) is positive semidefinite, so (17) is a convex function with respect to $\boldsymbol{w}_j$ [39]. Then, to fix others and updating $\boldsymbol{w}_j$, we have

$$\mathcal{O}\left(\boldsymbol{a}_j, P, (\boldsymbol{w}_j)^{t+1}\right) \leq \mathcal{O}\left(\boldsymbol{a}_j, P, (\boldsymbol{w}_j)^t\right). \tag{25}$$

As a result, the overall objective function value of (3) decreases monotonically in each iteration until Algorithm 2 converges. ■

D. Computational Complexity Analysis

The computational complexity of the two algorithms is same to each other because the top two steps of them are similar.

The first step of the objective function (3) is to solve (12). We need $O(n)$ time to compute $\boldsymbol{h}_j$ where n is the number of data points, and we need $O(t_1n_vn)$ to solve (14) where t_1 is the iteration number and n_v is the view number. We need n times to calculate each $\boldsymbol{a}_j, \forall j$, so the complexity of the first step of (3) in Algorithm 2 is $O((t_1n_vn+n)n)$.

The second step is an eigen-decomposition procedure, and the complexity of the generalized eigenvector problem is $O((n+c)n^2)$ where c is the cluster number. For solving (15), we need to calculate the c eigenvectors of the Laplacian matrix L_a, so its cost is $O(cn^2)$.

In the third step, we need $O(n_v^2n)$ to calculate the term of $Z_j^TZ_j$ and need $O(n_v^3)$ for matrix inversion. Because there are n data points, we need to calculate n times for $\boldsymbol{w}_j, \forall j$. Then, the complexity of the third step of (3) in Algorithm 2 is $O((n_v^2n+n_v^3)n)$.

Thus, the total time complexity of (3) is

$$O\left(\left(\left(t_1n_v+1+c+n_v^2\right)n^2+n_v^3n\right)t_o\right) \tag{26}$$

where t_o is the number of iteration of the three steps.

Since $n \gg t_1, n \gg c, n \gg n_v$, the main complexity is eigen-decomposition procedure which is also a basic calculation in SC-based methods.

V. Experimental Results

A. Datasets

1) UCI Digits: There are ten-class handwritten digits in the dataset [40]. They are "0," "1," . . . , and "9." Each digit has 200 samples, so it has 2000 data points and ten classes. We use six views for each data point. The first view is the 216-D profile-correlation feature, the second is the 76-D Fourier-coefficient feature, the third is 64-D Karhunen–Loeve-coefficient feature, the fourth is 240-D intensity-averaged feature in 2×3 windows, the fifth is 47-D Zernike moment feature, and the sixth is 6-D morphological feature. The ground-truth labels of the dataset are available.

2) Caltech-101: The image dataset has 101 categories of images [41]. We select widely used seven classes and obtain 1474 images [22]. The seven classes are faces, motorbikes, dollar bill, Garfield, stop sign, and windsor chair. Each image is described by six features. The first feature is the 48-D Gabor feature, the second is the 40-D wavelet-moment feature, the third is the 254-D CENTRIST feature, the fourth is the 1984-D HOG feature, and the fifth is the 512-D GIST feature, and the sixth is the 928-D LBP feature. The ground-truth labels are available.

3) Notting-Hill: The dataset is extracted from the movie "Notting-Hill" [42], [43]. There are five classes and 4660 facial images in the dataset. Each facial image is represented by three features. The first view is the 6750-D Gabor feature, the second view is 3304-D LBP feature, and the third view is the 2000-D intensity feature. The ground-truth labels are available.

4) COIL-20: The dataset is from the Columbia object image library [44]. There are 1440 images of 20 object categories [30]. Each class contains 72 images. The first view is 1024-D intensity feature, the second view is 3304-D LBP feature, and the third view is the 6750-D Gabor feature.

B. Experimental Setup

We evaluate the performance of the proposed multiview clustering with graph learning (MVGL) on the two synthetic datasets and four real datasets. MVGL is compared with the SC [25] and state-of-the-art multiview clustering methods, including CRSC [19], robust multiview k-means clustering (RMKMC) [28], RMSC [21], similarity network fusion (SNF) [45], MVSC [22], and multiple kernel k-means clustering (MKKM) to demonstrate its effectiveness. We compare MVGL with methods, including the following.

1) SC [25] is usually applied to each single view to confirm that considering multiview at the same time actually leads to a superior performance than any single view.
2) CRSC [19] is a relatively earlier multiview clustering method. As same as SC, graphs in different view are constructed by the Gaussian functions in CRSC. We use the default setting as authors' suggestion.
3) RMKMC [28] uses $\ell_{2,1}$-norm to obtain relatively robust result. As authors's suggestion, we have searched the logarithm of its parameter $\log_{10}\gamma$ in the range of $[0.1, 2]$ with interval 0.2 to obtain the best parameter.
4) RMSC [21] use the same graphs as SC and CRSC, and the standard Markov chain is utilized for clustering. Its parameter λ is searched from 0.005 to 100 as authors' suggestion.
5) SNF [45] uses the k-nearest neighbor graphs, and iteratively updates similarity matrix. The setting of the parameter μ is searched in the range of [0.3, 0.8] with interval 0.1 as authors's suggestion and other parameters are defaulted.
6) MVSC [22] uses local manifold integration to fuse heterogeneous features, and speeds up the graph construction. Its parameter r is searched in logarithm $\log_{10} r$ from 0.1 to 2 with interval 0.2 as authors' suggestion.

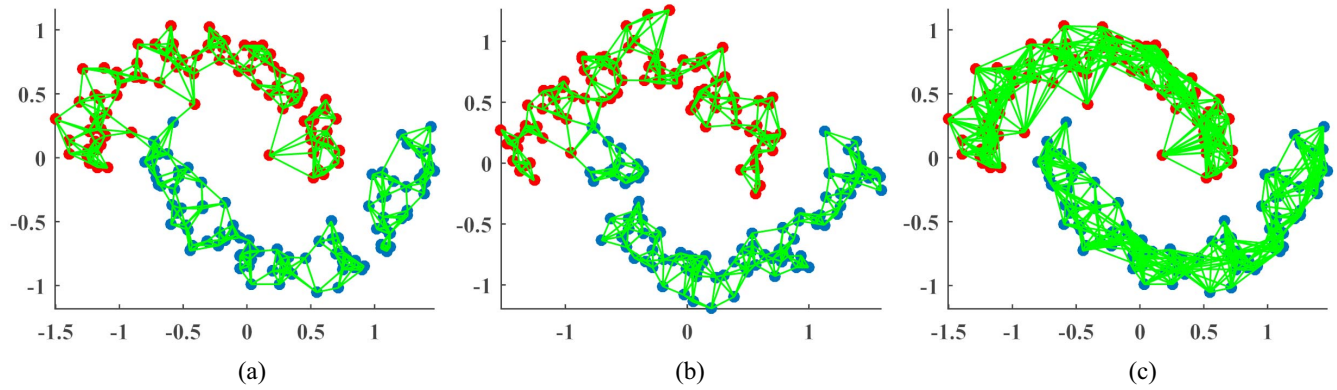


Fig. 2. Learned graph by MVGL on the two-moon synthetic data. Different color points denote the 200 data points. (a) Data points are $X^{(1)}$, and the lines are the learned $s_{ij}^{(1)}$ in view 1. (b) Data points are $X^{(2)}$, and the lines are the learned $s_{ij}^{(2)}$ in view 2. (c) Data points are $X^{(1)}$, and the lines are the learned a_{ij}.

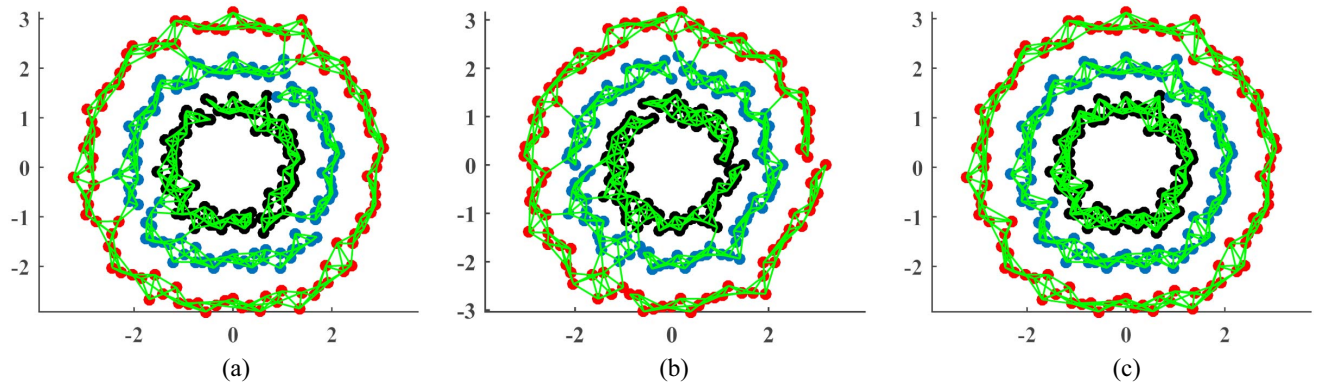


Fig. 3. Learned graph by MVGL on the three-circle synthetic data. Different color points denote the 300 data points. (a) Data points are $X^{(1)}$, and the lines are the learned $s_{ij}^{(1)}$ in view 1. (b) Data points are $X^{(2)}$, and the lines are the learned $s_{ij}^{(2)}$ in view 2. (c) Data points are $X^{(1)}$, and the lines are the learned a_{ij}.

7) MKKM [46] is a multiple kernel learning method for multiview clustering. The kernels of different views are constructed by the Gaussian functions, and these kernels are optimized by MKKM. To obtain the clustering indicators, MKKM also needs a post-process k-means. We have searched the regularization parameter λ from $\{2^{-15}, 2^{-14}, \ldots, 2^{15}\}$ as authors' suggestion.

Without loss of generality, for the eight methods, we run each method ten times and report the mean performance as well as the standard deviation. SC, CRSC, RMSC, SNF, MVSC, and MKKM need to perform k-means after they obtain the new representation of data points, so the metrics are different in each experiment because of the k-means clustering processing. In each experiment, we run k-means clustering processing 30 times to reduce the affect of random initialization, and we report the result with the minimum value for the objective function of k-means among results of these 30 times.

Our proposed MVGL does not require performing k-means and it obtains the clustering indicators using the learned graph directly. Each connected component belongs to one cluster and the clustering indicators are directly obtained by the learned global graph according to Tarjan's [35] strongly connected component algorithm. There are one regularization parameter β in Algorithm 1 and one parameter γ in Algorithm 2. In practice, we determine the two parameters in a heuristic way to accelerate the procedure [8]. Because the convergence criteria is that the sum of the top c smallest eigenvalues of L is zero, we set $\beta = 1$, then increase β if the connected components of $S^{(v)}$ is larger than c and decrease β if it is smaller than c during iteration. The strategy of the regularization parameter γ setting is as similar as β in Algorithm 1. Because the convergence criteria is that the sum of the top c smallest eigenvalues of L_a is zero, we set $\gamma = 1$, then increase γ if the connected components of $S^{(v)}$ is smaller than c and decrease γ if it is larger than c during iteration.

Seven metrics, clustering accuracy (ACC), normalized mutual information (NMI), purity, precision, recall, F-score, and adjusted rand index (ARI) are used to evaluate the clustering performance. These metrics are widely used, and they can be calculated by comparing the obtained label of each sample with the ground-truth label provided by the dataset. For all these metrics, the larger value indicates better clustering performance.

C. Results on Synthetic Datasets

We use two synthetic datasets to evaluate the clustering performance of MVGL. Our synthetic datasets consist of two views and they are generated as follows. As shown in Figs. 2 and 3, the first is two-moon dataset and the second

TABLE I
CLUSTERING PERFORMANCE

Methods	ACC	NMI	Purity	Precision	Recall	F-score	ARI
UCI digits							
SC 1	68.96±0.13	65.59±0.13	71.86±0.08	58.36±0.14	63.27±0.15	60.71±0.15	56.18±0.17
SC 2	69.56±0.03	63.36±0.03	69.56±0.04	56.92±0.04	58.09±0.04	57.50±0.04	52.75±0.04
SC 3	66.35±0.15	63.35±0.15	70.21±0.15	57.75±0.14	59.03±0.14	58.38±0.14	53.72±0.16
SC 4	64.03±0.17	61.22±0.10	66.92±0.16	53.87±0.14	56.18±0.16	55.04±0.15	49.91±0.17
SC 5	57.28±0.19	48.60±0.31	57.29±0.20	43.37±0.22	43.80±0.23	43.57±0.23	37.29±0.25
SC 6	47.25±0.00	49.01±0.00	50.65±0.00	38.22±0.00	38.74±0.00	38.48±0.00	31.62±0.00
CRSC	91.46±0.04	83.99±0.05	91.46±0.04	83.40±0.06	84.23±0.07	83.81±0.07	82.01±0.07
RMKMC	81.64±10.19	86.27±4.94	83.87±8.11	74.03±10.72	80.06±5.07	76.79±8.04	74.06±9.10
RMSC	86.32±0.03	78.03±0.07	86.32±0.03	75.37±0.05	76.48±0.05	75.92±0.05	73.24±0.06
SNF	88.35±0.00	88.90±0.00	88.35±0.00	84.87±0.00	86.74±0.00	85.79±0.00	84.21±0.00
MVSC	81.80±3.77	85.90±1.62	83.55±2.06	76.71±3.94	80.57±1.68	78.55±2.54	76.11±2.87
MKKM	89.45±0.00	81.74±0.00	89.45±0.00	80.64±0.00	81.19±0.00	80.92±0.00	78.80±0.00
MVGL	**94.20±0.00**	**89.05±0.00**	**94.20±0.00**	**87.57±0.00**	**89.15±0.00**	**88.35±0.00**	**94.20±0.00**
Caltech-101							
SC 1	28.28±0.36	15.78±0.14	63.24±0.09	51.81±0.21	20.50±0.18	29.38±0.22	09.63±0.20
SC 2	35.14±0.03	23.76±0.02	73.76±0.03	65.81±0.03	25.45±0.01	36.71±0.02	19.37±0.02
SC 3	38.21±0.07	27.08±0.06	79.42±0.04	67.52±0.08	26.28±0.03	37.83±0.04	20.71±0.05
SC 4	40.23±0.00	40.61±0.00	83.65±0.00	80.85±0.00	31.21±0.00	45.04±0.00	30.01±0.00
SC 5	40.48±0.03	35.22±0.04	81.58±0.04	76.37±0.05	28.89±0.02	41.92±0.03	26.34±0.04
SC 6	48.58±0.00	34.89±0.00	79.78±0.00	77.90±0.00	32.79±0.00	46.16±0.00	30.22±0.00
CRSC	44.69±0.30	37.22±0.24	79.23±0.29	78.08±0.32	31.83±0.20	45.22±0.25	29.48±0.29
RMKMC	52.11±4.97	47.22±3.85	83.65±0.76	84.26±5.25	39.32±4.07	53.58±4.67	38.57±5.59
RMSC	46.06±0.02	39.46±0.02	82.16±0.00	82.60±0.03	34.14±0.01	48.32±0.01	33.27±0.01
SNF	55.62±0.81	49.08±0.36	85.31±0.16	83.86±4.41	41.60±0.89	55.55±0.24	40.24±1.37
MVSC	53.27±6.63	52.91±2.90	84.42±0.96	78.49±3.38	40.61±4.27	53.46±4.35	36.98±4.87
MKKM	42.06±0.00	39.47±0.00	83.38±0.00	81.55±0.00	32.69±0.00	46.67±0.00	31.58±0.00
MVGL	**57.06±0.00**	**53.17±0.00**	**87.04±0.00**	**87.25±0.00**	**46.15±0.00**	**60.37±0.00**	**45.96±0.00**
Notting-Hill							
SC 1	70.15±0.00	67.28±0.00	80.17±0.00	68.24±0.00	68.22±0.00	68.23±0.00	59.29±0.00
SC 2	86.46±0.00	71.97±0.00	86.46±0.00	80.91±0.00	79.04±0.00	79.97±0.00	74.41±0.00
SC 3	73.80±0.00	60.80±0.00	76.55±0.00	68.37±0.00	65.37±0.00	66.84±0.00	57.77±0.00
CRSC	76.65±0.00	67.93±0.00	79.81±0.00	72.78±0.00	69.73±0.00	71.22±0.00	63.35±0.00
RMKMC	71.36±11.16	81.83±8.71	78.49±8.84	69.01±10.82	71.81±7.63	70.30±9.26	61.60±12.34
RMSC	82.77±0.00	77.25±0.00	85.04±0.00	81.61±0.00	82.93±0.00	82.27±0.00	77.23±0.00
SNF	89.16±0.00	88.25±0.00	89.16±0.00	85.77±0.00	98.09±0.00	91.52±0.00	88.93±0.00
MVSC	86.61±10.71	90.93±5.08	91.29±4.72	84.77±7.80	86.67±9.38	85.69±8.50	81.64±10.87
MKKM	78.20±0.00	71.74±0.00	80.15±0.00	75.66±0.00	72.13±0.00	73.85±0.00	66.72±0.00
MVGL	**100±0.00**	**100±0.00**	**100±0.00**	**100±0.00**	**100±0.00**	**100±0.00**	**100±0.00**
COIL-20							
SC 1	63.26±1.20	77.15±0.67	66.24±0.80	57.91±0.81	60.54±1.07	59.19±0.91	57.03±0.96
SC 2	74.26±0.58	83.64±0.50	76.72±0.52	68.64±1.00	72.17±0.82	70.36±0.89	68.78±0.94
SC 3	69.60±0.22	80.01±0.24	70.97±0.25	65.56±0.32	69.30±0.35	67.38±0.33	65.63±0.35
CRSC	75.10±1.15	84.06±0.53	76.27±1.06	70.98±1.05	72.91±1.19	71.93±1.10	70.45±1.15
RMKMC	48.51±4.19	68.65±2.48	52.65±3.16	35.64±4.67	56.30±3.39	43.41±3.57	39.75±4.02
RMSC	75.44±0.29	83.16±0.29	75.81±0.25	71.07±0.31	72.30±0.35	71.68±0.33	70.20±0.35
SNF	84.72±0.00	93.66±0.00	89.10±0.00	74.37±0.00	93.59±0.00	82.88±0.00	81.88±0.00
MVSC	75.47±3.02	87.48±1.83	78.20±2.31	69.09±5.54	75.62±2.17	72.13±3.83	70.61±4.09
MKKM	77.64±0.00	84.37±0.00	77.71±0.00	73.88±0.00	75.19±0.00	74.53±0.00	73.20±0.00
MVGL	**92.50±0.00**	**97.69±0.00**	**95.00±0.00**	**90.58±0.00**	**97.46±0.00**	**93.89±0.00**	**93.57±0.00**

Note: The best results are highlighted in bold.

is three-circle dataset. In the two-moon dataset, $X^{(1)}$ and $X^{(2)}$ both have 200 data points and we add 0.12 and 0.14 percentage of noise to obtain a two-view dataset. There are two clusters in the two-moon dataset and each cluster has 100 samples. In the three-circle dataset, $X^{(1)}$ and $X^{(2)}$ both have 300 data points and we add 0.14 and 0.16 percentage of noise to obtain a two-view dataset. There are three clusters in the three-circle dataset and each cluster has 100 samples. The noise is relatively large so that the data points in different clusters are quite close to each other, and the learned graphs in different views are well integrated into a global graph A in the two datasets as shown in Figs. 2 and 3.

D. Results on Real Datasets

After comparing the proposed method with other baseline algorithms, we show the clustering results in terms of ACC, NMI, purity, precision, recall, F-score, and ARI in Table I, respectively. In Table I, "SC 1" means that the SC is performed in the first view of a dataset, "SC 2" means that SC is performed in the second view of a dataset, and so on.

It can clearly be seen that MVGL achieves the best performance. MVGL improves the clustering performance significantly. The quantitative result fully demonstrates the superiority of MVGL because MVGL better captures the geometrical structure of the data space. The proposed method is different from SC-based and k-means clustering-based, and it can obtain an integrated global graph with a better structure, so the results are better than others. Surprisingly, MVGL obtains the ideal results in the Notting-Hill dataset. The Notting-Hill face dataset is extracted from well constrained videos [43]. Face images belonging to the same track are very similar to each other [43] which implies significant low-rank property. Therefore, the proposed method under the low-rank

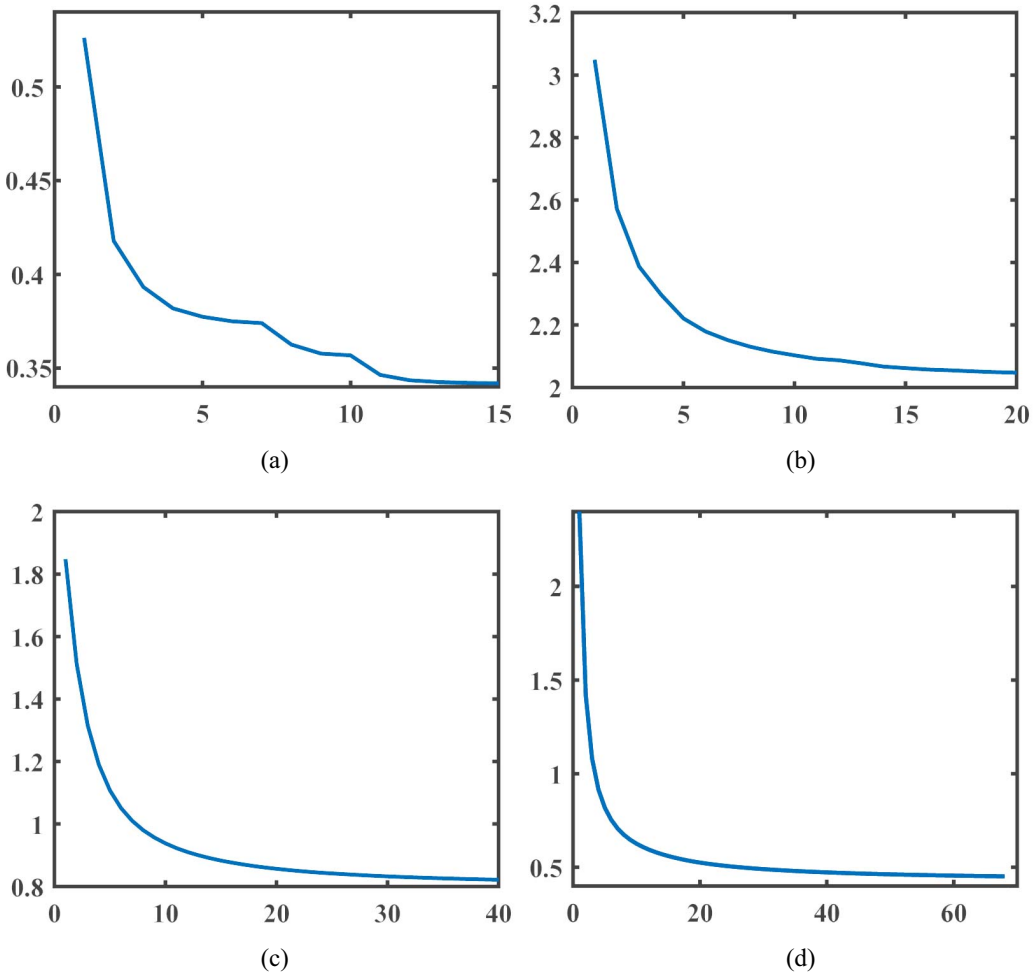


Fig. 4. Convergence curves over different datasets. (a) UCI digits. (b) Caltech-101. (c) Notting-Hill. (d) COIL-20.

assumption and with an integrated ideal global graph achieves much better results.

E. Convergence Study

To solve (3), we develop the Algorithm 2 with an efficient iteration. In Section IV-C, the convergence of Algorithm 2 is proved. To verify the convergence property of MVGL, Fig. 4 illustrates the convergence speed on all four real datasets. In each figure, the x-axis and the y-axis denote the iteration number and the corresponding objective function value, respectively. We can see that the value of the objective function decreases sharply within 20 iterations and then becomes steadily with more iterations. This indicates that MVGL converges sufficiently.

VI. Conclusion

Aiming to improve the multiview clustering performance, MVGL is proposed to enhance the quality of the graph. MVGL learns a global graph from different single view graphs. The integrated global graph has an exact number of the connected components that reflects cluster indicators. What's more, MVGL obtains the clustering indicators without post-process the k-means clustering or any graph cut techniques. Novel algorithms are developed to solve the proposed objective functions. Experimental results on real-world benchmark datasets demonstrate the effectiveness of MVGL.

References

[1] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[2] X. He and P. Niyogi, "Locality preserving projections," in *Proc. NIPS*, vol. 16. 2003, pp. 153–160.
[3] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. NIPS*, vol. 19. Vancouver, BC, Canada, 2006, pp. 507–514.
[4] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2773, Oct. 2010.
[5] Y. Yang *et al.*, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 723–742, Apr. 2012.
[6] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.
[7] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.
[8] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. ACM SIGKDD*, vol. 20. New York, NY, USA, 2014, pp. 977–986.
[9] X. Wang, Y. Liu, F. Nie, and H. Huang, "Discriminative unsupervised dimensionality reduction," in *Proc. IJCAI*, Buenos Aires, Argentina, 2015, pp. 3925–3931.

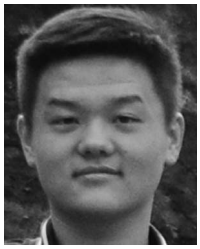
[10] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 1969–1976.
[11] M. Luo *et al.*, "Adaptive unsupervised feature selection with structure regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.
[12] Z. Kang, C. Peng, and Q. Cheng, "Twin learning for similarity and clustering: A unified kernel approach," in *Proc. AAAI*, San Francisco, CA, USA, 2017, pp. 2080–2086.
[13] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. COLT*, vol. 11. Madison, WI, USA, 1998, pp. 92–100.
[14] R. Ghani, "Combining labeled and unlabeled data for multiclass text categorization," in *Proc. ICML*, vol. 2. 2002, pp. 187–194.
[15] U. Brefeld and T. Scheffer, "Co-EM support vector learning," in *Proc. ICML*, vol. 21. Banff, AB, Canada, 2004, pp. 16–23.
[16] M.-F. Balcan, A. Blum, and K. Yang, "Co-training and expansion: Towards bridging theory and practice," in *Proc. NIPS*, vol. 17. Vancouver, BC, Canada, 2004, pp. 89–96.
[17] W. Wang and Z.-H. Zhou, "A new analysis of co-training," in *Proc. ICML*, vol. 27. Haifa, Israel, 2010, pp. 1135–1142.
[18] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proc. ICML*, vol. 28. Bellevue, WA, USA, 2011, pp. 393–400.
[19] A. Kumar, P. Rai, and H. Daumé, "Co-regularized multi-view spectral clustering," in *Proc. NIPS*, vol. 24. Granada, Spain, 2011, pp. 1413–1421.
[20] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *Proc. CVPR*, vol. 24. Colorado Springs, CO, USA, 2011, pp. 1977–1984.
[21] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI*, vol. 28. Québec City, QC, Canada, 2014, pp. 2149–2155.
[22] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. AAAI*, vol. 29. Austin, TX, USA, 2015, pp. 2750–2756.
[23] J. W. Son, J. Jeon, A. Lee, and S.-J. Kim, "Spectral clustering with brainstorming process for multi-view data," in *Proc. AAAI*, vol. 31. San Francisco, CA, USA, 2017, pp. 2548–2554.
[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[25] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, vol. 2. Vancouver, BC, Canada, 2002, pp. 849–856.
[26] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. NIPS*, vol. 17. Vancouver, BC, Canada, 2004, pp. 1601–1608.
[27] U. Von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[28] X. Cai, F. Nie, and H. Huang, "Multi-view K-means clustering on big data," in *Proc. IJCAI*, vol. 23. Beijing, China, 2013, pp. 2598–2604.
[29] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. CVPR*, vol. 28. Boston, MA, USA, 2015, pp. 586–594.
[30] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. ICCV*, Santiago, Chile, 2015, pp. 1582–1590.
[31] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *Proc. ICCV*, Santiago, Chile, 2015, pp. 4238–4246.
[32] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. CVPR*, vol. 30. 2017, pp. 4279–4287.
[33] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The Laplacian spectrum of graphs," *Graph Theory Combinatorics Appl.*, vol. 2, no. 12, pp. 871–898, 1991.
[34] F. R. Chung, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1997.
[35] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, 1972.
[36] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations I," *Proc. Nat. Acad. Sci. USA*, vol. 35, no. 11, pp. 652–655, 1949.
[37] M. L. Overton and R. S. Womersley, "On the sum of the largest eigenvalues of a symmetric matrix," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 1, pp. 41–45, 1992.
[38] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, "Spectral K-way ratio-cut partitioning and clustering," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 9, pp. 1088–1096, Sep. 1994.
[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[40] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Multiple+Features
[41] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, 2007.
[42] B. Wu, Y. Zhang, B.-G. Hu, and Q. Ji, "Constrained clustering and its application to face clustering in videos," in *Proc. CVPR*, Portland, OR, USA, 2013, pp. 3507–3514.
[43] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh, "Constrained multi-view video face clustering," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4381–4393, Nov. 2015.
[44] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-005-96, 1996.
[45] B. Wang *et al.*, "Similarity network fusion for aggregating data types on a genomic scale," *Nat. Methods*, vol. 11, no. 3, pp. 333–337, 2014.
[46] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. AAAI*, vol. 30. Phoenix, AZ, USA, 2016, pp. 1888–1894.

Kun Zhan received the B.S. and Ph.D. degrees from the School of Information Science and Engineering, Lanzhou University, Lanzhou, China, in 2005 and 2010, respectively.

He was a visiting student with the Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS, Canada, from 2009 to 2010. He is currently with Lanzhou University. His current research interests include machine learning and neural networks.

Changqing Zhang received the B.S. and M.S. degrees from the College of Computer Science, Sichuan University, Chengdu, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2016.

He is an Assistant Professor with the School of Computer Science and Technology, Tianjin University. His current research interests include machine learning, data mining, and computer vision.



Junpeng Guan received the B.S. degree in electronic information science and technology from Lanzhou University, Lanzhou, China, in 2015, where he is currently pursuing the master's degree.

His current research interest includes feature learning.

Junsheng Wang received the B.S. degree in automation from Beijing Jiaotong University, Beijing, China, in 2016. He is currently pursuing the master's degree with Lanzhou University, Lanzhou, China.

His current research interest includes multiview learning.