Structured Graph Learning for Scalable Subspace Clustering: From Single View to Multiview

Zhao Kang, Zhiping Lin, Xiaofeng Zhu, *Senior Member, IEEE*, and Wenbo Xu

***Abstract*—Graph-based subspace clustering methods have exhibited promising performance. However, they still suffer some of these drawbacks: they encounter the expensive time overhead, they fail to explore the explicit clusters, and cannot generalize to unseen data points. In this work, we propose a scalable graph learning framework, seeking to address the above three challenges simultaneously. Specifically, it is based on the ideas of anchor points and bipartite graph. Rather than building an $n \times n$ graph, where n is the number of samples, we construct a bipartite graph to depict the relationship between samples and anchor points. Meanwhile, a connectivity constraint is employed to ensure that the connected components indicate clusters directly. We further establish the connection between our method and the K-means clustering. Moreover, a model to process multiview data is also proposed, which is linearly scaled with respect to n. Extensive experiments demonstrate the efficiency and effectiveness of our approach with respect to many state-of-the-art clustering methods.**

***Index Terms*—Anchor graph, bipartite graph, connectivity constraint, large-scale data, multiview learning, out of sample, subspace clustering.**

I. Introduction

AS AN unsupervised technique, clustering has always been an important research topic in machine learning, pattern recognition, and data mining. During the past few decades, a plethora of clustering methods has been developed, such as K-means [1], spectral clustering [2], hierarchical clustering [3], DBSCAN [4], and deep clustering [5], to name a few. In recent years, to tackle the curse of dimensionality, subspace clustering has received increasing attention. It is capable of finding relevant dimensions spanning a subspace for each cluster [6], [7]. Among multiple approaches, graph-based subspace clustering often generates the best performance [8], [9]. As a result, graph-based subspace clustering methods have been in hot pursuit in recent years.

Specifically, some representative methods of this category include sparse subspace clustering (SSC) [10], low-rank representation (LRR) [11], and least squares regression (LSR) [12]. To enjoy the benefit of discriminative features brought by deep neural works, some deep subspace clustering networks have recently been proposed [13], [14]. In general, they are implemented in two individual steps. First, an $n \times n$ graph that represents the pairwise similarity between samples is learned. Second, the learned graph is input to the spectral clustering algorithm, which typically involves eigendecomposition of the Laplacian matrix. This pipeline procedure has one flaw, that is, the graph might potentially not be optimal for the downstream clustering since it might fail to achieve the cluster structure with an exact cluster number [15], [16].

In particular, it is always difficult to capture complex similarity patterns for data with high-level semanticity (human level), e.g., speech, textual data, images, and videos [17], [18]. In addition, the graph requires $\mathcal{O}(n^2)$ memory, and eigendecomposition often consumes $\mathcal{O}(n^3)$ time [19]. Consequently, they are costly and make the processing of large-scale data prohibitive. Another usually ignored aspect is the out-of-sample problem [20], that is, the existing framework does not generalize to unseen data points. Put differently, for testing data points that are never seen during training, the existing graph-based subspace clustering model cannot handle them because the "graph structure" must be learned for all of the data points during training.

Some recent research endeavors are devoted to reducing the algorithm complexity. For example, some off-the-shelf projection and sampling methods are applied for spectral clustering [21]. The bipartite graph is also widely used to speed up spectral clustering [22]–[24]. To get rid of the ad hoc functions used to calculate similarity, several scalable subspace clustering methods are proposed. Combining sparse representation and bipartite graph, Adler *et al.* [25] proposed a linear subspace clustering algorithm. Wang *et al.* [26] applied a data selection method to speed up computation. Li *et al.* [27] adopted sampling and fast regression coding to cluster the codes of data points. You *et al.* [28] employed orthogonal matching pursuit to reduce the computational load, but it often loses clustering accuracy. Pourkamali-Anaraki *et al.* [29] proposed an efficient solver for SSC. Accelerated LRR is also developed [30], [31]. Peng *et al.* [32] solved the large-scale

Manuscript received 12 July 2020; revised 27 October 2020 and 14 January 2021; accepted 20 February 2021. Date of publication 17 March 2021; date of current version 18 August 2022. This work was supported in part by the Natural Science Foundation of China under Grant 61806045 and Grant U19A2059; in part by the National Key Research and Development Program of China under Grant 2018AAA0100204 and Grant 2018YFC0807500; in part by the Sichuan Science and Technology Program under Project 2020YFS0057; in part by the Ministry of Science and Technology of Sichuan Province Program under Grant 2018GZDZX0048 and Grant 20ZDYF0343; and in part by the Fundamental Research Fund for the Central Universities under Project ZYGX2019Z015. This article was recommended by Associate Editor S. Das. *(Corresponding author: Zhao Kang.)*

Zhao Kang, Zhiping Lin, and Xiaofeng Zhu are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: zkang@uestc.edu.cn; 201921080534@std.uestc.edu.cn; seanzhuxf@gmail.com).

Wenbo Xu is with the School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: xuwenbo@uestc.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3061660.

Digital Object Identifier 10.1109/TCYB.2021.3061660

challenge by converting it to an out-of-sample problem. Though these techniques can alleviate computational overhead, they often ignore the graph structure or fail to address the out-of-sample problem.

Moreover, by virtue of the development of data acquisition and processing technologies, an increasing volume of data is represented by multiple views [33]–[37]. For example, a video might consist of text, images, and sounds [38], [39]; an image can be described in different features, e.g., SIFT, GIST, LBP, HoG, and Garbor [40], [41]; and a document can be translated into different languages [42], [43]. These heterogeneous features often supply complementary information that could be helpful for our tasks at hand [44]–[46]. As a result, multiview subspace clustering has also been investigated [47]–[49]. For instance, Cao *et al.* [50] considered both the consistency and the diversity among the multiple views; Gao *et al.* [51] learned multiple graphs and let them correspond to a unique cluster indicator matrix; Zhang *et al.* [47], [52] performed subspace clustering in latent space; and Kang *et al.* [53] proposed to fuse multiview information in partition space. These algorithms often achieve better results than the single view methods.

Unfortunately, most of the existing multiview subspace clustering methods also encounter a scalability problem, which hinders their applications on large-scale data. Some single-view large-scale subspace clustering cannot be directly extended to cope with multiview data. Recently, Kang *et al.* [54] made the first effort to tackle the challenge of large-scale multiview subspace clustering. However, it ignores the graph structure and fails to deal with unseen data.

In this article, we simultaneously deal with the three issues of subspace clustering, that is: 1) high complexity; 2) explicit graph structure; and 3) out of sample. Apart from the single-view model, we also propose a multiview model, seeking a structured graph that compatibly crosses multiple views. First, we build a dictionary matrix by selecting m landmarks from raw data by the K-means algorithm. Second, we learn the relationship between our raw data and the landmarks, which generates a bipartite graph with k-connected components if the data contain k clusters. Thus, a cluster indicator matrix is naturally obtained. Third, for multiview data, extra viewwise weights are introduced to discriminate different views. The advantages of our approach are: the small affinity matrix can preserve the manifold of the data, the constraint of the bipartite graph discovers the underlying cluster structure, our method addresses the out-of-sample problem, and our overall complexity is linear to the size of data. Compared to state-of-the-art techniques, our methods gain a lot in terms of effectiveness and efficiency.

In a nutshell, the key contributions of this article are as follows.

1) We present a novel structured graph learning (SGL) framework for large-scale subspace clustering in linear time. Besides, this method also solves the out-of-sample problem.
2) A scalable multiview subspace clustering method is proposed. The bipartite graph, cluster indicator matrix, and viewwise weights learn from each other interactively and supervise each other adaptively.
3) Theoretical analysis establishes the connection between our method and K-means clustering. Extensive experimental results demonstrate the superiority of our method with respect to many state-of-the-art techniques.

The remainder of this article is as follows. In Section II, we introduce some background on subspace clustering and anchor graph. Section III presents our proposed graph learning model and solution. In Section IV, we perform theoretical analysis and compare time complexity. Section V extends our method to the multiview data. Single-view and multiview experiments are conducted in Sections VI and VII, respectively. This article is rounded up with a conclusion in Section VIII.

II. Background

In this section, we present an overview of the single-view and multiview subspace clustering, and then introduce the anchor graph.

A. Subspace Clustering

Given data $X \in \mathcal{R}^{d\times n}$, which includes n samples each with d features, subspace clustering assumes that each data sample can be expressed as a combination of other data points in the same subspace. This combination coefficient matrix is considered as the similarity graph, which captures the global structure of data. In general, the following model is solved [16]:

$$\min_{S} \quad \|X - XS\|_F^2 + \alpha f(S) \quad \text{s.t.} \quad S \geq 0, S\mathbf{1} = \mathbf{1} \tag{1}$$

where $S \in \mathcal{R}^{n\times n}$ is the non-negative similarity matrix and $\alpha > 0$ is a balance parameter. The first term is the reconstruction error and the second term $f(\cdot)$ is a regularizer function, including low-rank constraint [11], [55]; sparse ℓ_1 norm [10]; and Frobenius norm [56], [57]. $\mathbf{1}$ is a vector of ones and $S\mathbf{1} = \mathbf{1}$ means that each row of S adds up to 1.

It can be seen that the size of graph S, which often results in $\mathcal{O}(n^3)$ computational complexity, would be a burden on both computation and storage for large-scale data. Furthermore, the subsequent spectral clustering step also suffers $\mathcal{O}(n^3)$ complexity. Some recent subspace clustering methods with low complexity have been developed. For instance, SSCOMP [28] is built on orthogonal matching pursuit, ESSC [29] is a proximal gradient framework to solve SSC, and ALRR [30] is a faster solver for LRR. Though they can deal with large-scale data, they fail to cope with the out-of-sample problem and ignore the graph structure. Kang *et al.* [16] considered that the graph should have exactly k components if the data contain k clusters. However, it has $\mathcal{O}(n^3)$ complexity and cannot address the out-of-sample challenge. For new samples, SLSR [32] projects them into the union of subspaces spanned by in-sample data. However, SLSR does not explicitly consider the graph structure. In practice, the data can display structures beyond simply being low rank or sparse [58]. In summary, there is no single method that can address all three challenges faced by subspace clustering: 1) high complexity; 2) graph structure; and 3) out of sample.

For multiview subspace clustering, more attention is paid to how to boost the clustering accuracy by fully exploring

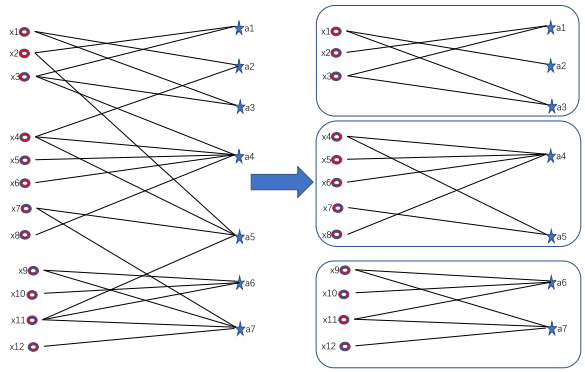

Fig. 1. Optimal bipartite graph with constraint. Initially, the nodes in the left of the bipartite graph are connected randomly with the right anchors. After enforcing the constraint, the structured bipartite graph contains a specified number of connected components.

the complementary information carried by multiview data. For instance, multiview low-rank SSC (MLRSSC) [59] harnessing both low-rank and sparsity constraints shows much better performance than previous methods, such as [60]. Multiview subspace clustering with intactness-aware similarity (MSC_IAS) [61] tries to construct a graph in latent space, which leads to superior accuracy. Kang *et al.* [54] made the first attempt to address the scalability issue of multiview subspace clustering. Though it has a linear complexity, it fails to consider the discriminative nature of different views. Moreover, the graph learning and clustering stage are separated, so the graph structure is ignored.

B. Anchor Graph

Anchors or landmarks were previously used in scalable spectral clustering [23]. Basically, its idea is to select a small set of data samples called anchors or landmarks to represent the neighborhood structure. Typically, these representative points are chosen based on the K-means method (the cluster centers) or random sampling [62]. Specifically, with m anchors $A = \{a_1, \ldots, a_m\} \in \mathcal{R}^{d\times m}$, a small graph $Z \in \mathcal{R}^{n\times m}$ is built to measure the relationship between the anchors and the entire data. Typically, Z is constructed by the Gaussian kernel function [23], [63], which might not be flexible enough to characterize complex data.

For subspace clustering, we can borrow the idea of anchor and treat A as a dictionary [54]. Afterward, we can solve the following model to learn Z automatically, that is:

$$\begin{aligned} \min_{Z} \ & \left\|X - AZ^\top\right\|_F^2 + \alpha\|Z\|_F^2 \\ \text{s.t.} \ & 0 \le Z, \ Z\mathbf{1} = \mathbf{1}. \end{aligned} \tag{2}$$

Though (2) is quite simple, it does not consider any cluster structure. Put differently, Z might be just one connected component, as shown in the left-hand side of Fig. 1. It would be desired if it has exactly k-connected components denoted by $Z \in \Omega$ [22], as shown in the right-hand part of Fig. 1. Then, problem (2) becomes

$$\begin{aligned} \min_{Z} \ & \left\|X - AZ^\top\right\|_F^2 + \alpha\|Z\|_F^2 \\ \text{s.t.} \ & 0 \le Z, \ Z\mathbf{1} = \mathbf{1}, Z \in \Omega. \end{aligned} \tag{3}$$

In the next section, we will show how to address this challenging problem.

III. Structured Graph Learning

To explicitly explore the cluster structure of Z, we can make use of the bipartite graph. To be precise, bipartite graph S associated with Z is defined by $S = \begin{bmatrix} & Z \\ Z^\top & \end{bmatrix} \in \mathcal{R}^{(n+m)\times(n+m)}$. Then, the normalized Laplacian matrix L is expressed as $L = I - D^{-(1/2)}SD^{-(1/2)}$, where D is a diagonal matrix with its ith diagonal element defined as $d_i = \sum_{j=1}^{n+m} s_{ij}$. According to the spectral graph theory [64], the normalized Laplacian has the following property.

Theorem 1: The number of connected components in S is equal to the cardinality k of the 0 eigenvalue of L.

By Theorem 1, if $\text{rank}(L) = (n+m) - k$ meets, the n data samples and m anchors are grouped into k clusters. Therefore, to achieve the ideal subspace clustering with specified k clusters, we can explicitly express the constraint in problem (3). It yields

$$\begin{aligned} \min_{Z} \ & \left\|X - AZ^\top\right\|_F^2 + \alpha\|Z\|_F^2 \\ \text{s.t.} \ & 0 \le Z, \ Z\mathbf{1} = \mathbf{1}, \ \text{rank}(L) = (n+m) - k. \end{aligned} \tag{4}$$

Considering that the rank constraint is hard to tackle, we can relax the constraint by following [65]. Eventually, our proposed SGL framework for subspace clustering can be formulated as

$$\begin{aligned} \min_{Z,F} \ & \left\|X - AZ^\top\right\|_F^2 + \alpha\|Z\|_F^2 + \beta Tr\left(F^\top LF\right) \\ \text{s.t.} \ & 0 \le Z, \ Z\mathbf{1} = \mathbf{1}, F^\top F = I \end{aligned} \tag{5}$$

where $F \in \mathcal{R}^{(n+m)\times k}$. It is worth pointing out that the above SGL can also be applied to semisupervised classification [66]. This problem can be solved by an alternating optimization strategy.

A. Optimization Strategy

We solve Z and F iteratively, that is, fix one of them and then update the other one.

1) Fix F and Solve Z: Note that L is $I - D^{-(1/2)}SD^{-(1/2)}$, so both S and D depend on variable Z. Fortunately, we can employ the following equation:

$$\text{Tr}\left(F^\top LF\right) = \frac{1}{2}\sum_{i=1}^{(n+m)}\sum_{j=1}^{(n+m)} s_{ij}\left\|\frac{F_{i,:}}{\sqrt{d_i}} - \frac{F_{j,:}}{\sqrt{d_j}}\right\|_2^2. \tag{6}$$

Considering the structure of S, we can further convert the above formula into the following formulation:

$$\text{Tr}\left(F^\top LF\right) = \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}\left\|\frac{F_{i,:}}{\sqrt{d_i}} - \frac{F_{n+j,:}}{\sqrt{d_{n+j}}}\right\|_2^2. \tag{7}$$

Defining $\|(F_{i,:}/\sqrt{d_i}) - [(F_{n+j,:})/(\sqrt{d_{n+j}})]\|_2^2$ as W_{ij}, we can solve our problem row by row as

$$\min_{Z_{i,:}} \; Z_{i,:}A^\top AZ_{i,:}^T - 2X_{:,i}^\top AZ_{i,:}^\top + \alpha Z_{i,:}Z_{i,:}^\top + \beta W_{i,:}Z_{i,:}^\top$$
$$\text{s.t. } 0 \le Z_{ij} \le 1, \sum_j Z_{ij} = 1. \tag{8}$$

This problem can be easily solved via convex quadratic programming.

2) Fix Z and Solve F: When Z is fixed, the first and the second terms in (5) become constant. Our problem can be equivalently written as

$$\max_{F\in\mathbb{R}^{(n+m)\times k}, F^\top F=I} \operatorname{Tr}\left(F^\top D^{-\frac{1}{2}} S D^{-\frac{1}{2}} F\right). \tag{9}$$

In general, computing the eigenvectors of S takes $\mathcal{O}(k(n+m)^2)$. To circumvent this complexity, we employ the special structure of S and compute the eigenvectors of Z instead. Concretely, we decompose F and D as

$$F = \begin{bmatrix} U \\ V \end{bmatrix}, \; D = \begin{bmatrix} D_U & \\ & D_V \end{bmatrix} \tag{10}$$

where $U \in \mathcal{R}^{n\times k}, V \in \mathcal{R}^{m\times k}, D_U \in \mathcal{R}^{n\times n}$, and $D_V \in \mathcal{R}^{m\times m}$, (9) can be rewritten as follows:

$$\max_{U^\top U + V^\top V = I} \operatorname{Tr}\left(U^\top D_U^{-\frac{1}{2}} Z D_V^{-\frac{1}{2}} V\right). \tag{11}$$

According to the following theorem [22], it can be easily solved.

Theorem 2: Suppose $Q \in \mathcal{R}^{n\times m}, X \in \mathcal{R}^{n\times k}$, and $Y \in \mathcal{R}^{m\times k}$. The optimal solutions to the problem

$$\max_{X^\top X + Y^\top Y = I} \operatorname{Tr}\left(X^\top Q Y\right)$$

are $X = (\sqrt{2}/2)U_1$ and $Y = (\sqrt{2}/2)V_1$, where U_1 and V_1 are corresponding to the top k left and right singular vectors of Q, respectively.

The complete algorithm for subspace clustering is summarized in Algorithm 1. By the theory of alternating optimization [67], the objective function value of our problem (5) will monotonically decrease in each iteration. Moreover, since the objective function has a lower bound, such as zero, the above iteration converges.

B. Out-of-Sample Problem

The out-of-sample problem is hard and little discussed for subspace clustering. This is because we must compute the graph that consists of all data points, which is inherently impossible to just involve unseen data. In contrast, our SGL method can handle new data. Note that our method also outputs embedding vectors V and cluster labels for the anchor points. Then, we just need to implement the classic k-nearest neighbor (kNN) algorithm, which will propagate the labels to the new data. For each new data point, this process takes $\mathcal{O}(md)$ time, which is far lower than the cost carried out on the training data $\mathcal{O}(nd)$. Therefore, our method can handle the out-of-sample problem efficiently.

Algorithm 1 SGL for Subspace Clustering

Input: Data matrix $X \in \mathcal{R}^{d\times n}$, anchor matrix $A \in \mathcal{R}^{d\times m}$, cluster number k, parameters α and β
Output: k clusters

1: Initialize the matrix F randomly.
2: **while** convergence condition does not meet **do**
3: Update Z in Eq. (8) via convex quadratic programming
4: Update U in Eq. (11) by Theorem 2
5: **end while**
6: Run K-means on matrix U to achieve the final partition

IV. Theoretical Analysis

In this section, we demonstrate that our method is connected to K-means method and provide the computational complexity analysis.

A. Relationship With K-Means Algorithm

Theorem 3: When we let α go to ∞, our problem (4) is equal to the K-means problem.

Proof: In (4), we set a constraint to make Z satisfy this property: each component contains several data points and anchor points; the number of connected components is k, which means all the points are classified into clusters. Thus, the value of $\beta \operatorname{Tr}(F^\top LF)$ would be equal to zero. Denote the ith component of Z by $Z^i \in \mathcal{R}^{m_i\times n_i}$, where n_i represents the data points number corresponding to this component and m_i denotes the anchor number corresponding to this component. Hence, solving (5) is to solve the following problem for each component of Z:

$$\min_{Z^i} \; \left\|X^i - A^i\left(Z^i\right)^\top\right\|_F^2 + \alpha\left\|Z^i\right\|_F^2$$
$$\text{s.t. } \; Z^i\mathbf{1} = \mathbf{1}, \quad 0 \le Z^i \le 1 \tag{12}$$

where X^i and A^i consist of samples and anchors corresponding to the ith component of Z. When $\alpha \to \infty$, the above problem becomes

$$\min_{Z^i} \; \left\|Z^i\right\|_F^2$$
$$\text{s.t. } \; Z^i\mathbf{1} = \mathbf{1}, \quad 0 \le Z^i \le 1. \tag{13}$$

The optimal solution is that all elements of Z^i are equal to $(1/m_i)$. Thus, when $\alpha \to \infty$, the optimal solution Z to problem (4) is

$$z_{ij} = \begin{cases} \frac{1}{m_p}, & x_i \text{ and } a_j \text{ in the same } p\text{th component} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Let us denote the solution set of this partition as $\mathcal{K}$. It can be shown that $\|Z\|_F^2 = k$. Thus, (4) becomes

$$\min_{Z_i\in\mathcal{K}} \left\|X_i - AZ_i^\top\right\|^2. \tag{15}$$

We can find that (15) is exactly the objective function in the K-means method and $AZ_i^\top$ denotes the centroid of cluster i. Therefore, the problem (4) is the problem of K-means. ■

TABLE I
SUMMARY OF COMPUTATIONAL COMPLEXITY OF VARIOUS METHODS. IN THIS TABLE, n IS THE NUMBER OF DATA POINTS AND d IS THE SIZE OF FEATURES. m_1 IS THE NUMBER OF PREDEFINED UPPER BOUND FOR THE RANK OF THE COEFFICIENT MATRIX. t IS THE NUMBER OF ITERATIONS UNTIL THE CONVERGENCE. t_1 IS NUMBER OF ITERATIONS FOR THE K-MEANS ALGORITHM. t_2 IS THE NUMBER OF ITERATIONS OF THE SUBALTERNATING SYSTEM. m IS THE NUMBER OF ANCHORS. k IS THE NUMBER OF THE CLUSTERS

Method	Time complexity
ALRR	$\mathcal{O}(4dm_1n)$
KMM	$\mathcal{O}\left(n\left(\left(md+mc+m^2\right)t_2+md\right)t\right)$
ESSC	$\mathcal{O}(n\log n)$
FNC	$\mathcal{O}(nmd+nmk)$
SSCOMP	$\mathcal{O}(n^2dt)$
SGL	$\mathcal{O}(nm^3t+2mnt+nmt_1d+nk^2t_1)$

B. Complexity Analysis

The proposed method uses the anchor idea to construct a smaller graph $Z \in \mathcal{R}^{n\times m}(m \ll n)$ and performs singular value decomposition (SVD) on a smaller matrix Z, so that the complexity can be reduced significantly. Specifically speaking, denoting t as the iteration number, we implement SVD on Z in each iteration to obtain the indicator matrix U, which takes $\mathcal{O}(m^3t+mnt)$. W computation costs $\mathcal{O}(mnt)$. We apply the built-in MATLAB function *quadprog* to solve Z in each iteration, which leads to $\mathcal{O}(nm^3t)$. As shown in (8), Z can be efficiently solved in parallel. Besides, we need extra $\mathcal{O}(nmt_1d)$ time to build the dictionary A by the K-means algorithm and $\mathcal{O}(nk^2t_1)$ time to achieve the final results by applying K-means on U, where t_1 is another iteration number. In conclusion, the time complexity of our method is linear to the number of points n.

We compare the time complexity of several recent scalable clustering methods in Table I. As we can see, most methods have a linear complexity. As shown in the experimental part, some of them sacrifice accuracy in the pursuit of improving time efficiency.

V. MULTIVIEW STRUCTURED GRAPH LEARNING

Compared to the single-view scenario, few large-scale multiview subspace clustering is studied. In this section, we show that our SGL method can be easily extended to handle multiview data $[X^1, X^2, \ldots, X^c]$, where $X^v \in \mathcal{R}^{d^{(v)}\times n}$ is the vth view data matrix with $d^{(v)}$ features. For multiview clustering, all views are required to share the same cluster pattern. Therefore, it is reasonable to assume that there exists a unique graph Z. However, different views might play various roles, so we introduce a weight factor λ^v to balance the importance of different views. Finally, our proposed multiview SGL (MSGL) method can be formulated as

$$\min_{Z,F,\{\lambda^v\geq 0\}} \sum_{v=1}^{c}\lambda^v\left\|X^v-A^vZ^\top\right\|_F^2+\alpha\|Z\|_F^2+\beta Tr\left(F^\top LF\right) + \sum_{v=1}^{c}\left(\lambda^v\right)^\gamma \text{ s.t. } 0\leq Z,\ Z\mathbf{1}=\mathbf{1},\ F^\top F=I \tag{16}$$

Algorithm 2 MSGL for Subspace Clustering

Input: Data matrix $[X^1;\cdots;X^c]$, anchor matrix $[A^1;\cdots;A^c]$, cluster number k, parameter α, β, γ
Output: k clusters

1: Initialize the matrix F randomly and λ^v as $1/c$.
2: **while** convergence condition does not meet **do**
3: Update Z in Eq. (17).
4: Update F as Eq. (11).
5: Update λ^v by Eq. (20).
6: **end while**
7: Run K-means on matrix U to achieve final partition

where $\gamma < 0$. In this model, different anchor points are generated for different views. In a similar way as SGL, we can optimize the three variables alternatively.

A. Fix λ^v, F, and Update Z

The subproblem we are going to solve is

$$\min_{Z_{i,:}} \sum_{v=1}^{c}\lambda^v\left(Z_{i,:}\left(A^v\right)^\top A^vZ_{i,:}^\top-2\left(X_{:,i}^v\right)^\top A^vZ_{i,:}^\top\right)+\alpha Z_{i,:}Z_{i,:}^\top + \beta W_{i,:}Z_{i,:}^\top \text{ s.t. } 0\leq Z_{ij}\leq 1,\ \sum_j Z_{ij}=1. \tag{17}$$

This quadratic problem can be easily solved.

B. Fix λ^v, Z, and Update F

This subproblem would be the same as (11).

C. Fix F, Z, Update λ^v

For simplicity, we denote $\|X^v-A^vZ^\top\|_F^2$ as h^v. Our objection function becomes

$$H\left(\lambda^v\right)=\sum_{v=1}^{c}\lambda^vh^v+\sum_{v=1}^{c}\left(\lambda^v\right)^\gamma. \tag{18}$$

It can be solved by taking the derivative on $\lambda^{(v)}$ and setting it to zero

$$\frac{\partial H}{\partial\lambda^v}=h^v+\gamma\left(\lambda^v\right)^{\gamma-1}=0. \tag{19}$$

It yields

$$\lambda^v=\left(-\frac{h^v}{\gamma}\right)^{\frac{1}{\gamma-1}}. \tag{20}$$

Because of $h^v \geq 0$ and $\gamma < 0$, $\lambda^v \geq 0$ is met. The complete steps to multiview subspace clustering is outlined in Algorithm 2. It is worth pointing out that MSGL inherits all advantages of SGL, that is, explicit graph structure, a linear complexity, extensions to out-of-sample data, convergence guarantee.

VI. SINGLE-VIEW EXPERIMENTS

In this section, we conduct several experiments to evaluate our method on single-view datasets.

TABLE II
DESCRIPTION OF THE SINGLE-VIEW DATASETS

Data	Instance #	Feature #	Class #
BA	1,404	320	120
ORL	400	1,024	40
TR11	414	6,429	9
TR41	878	7,454	10
TR45	690	8,261	10
RCV1-4	9,625	29,992	4
MNIST	70,000	784	10
CoverType	581,012	54	7
Pokerhand	1,000,000	10	10

A. Datasets

Nine popular datasets are tested. Specifically, BA,[1] ORL,[2] and MNIST[3] are image data, while TR11, TR41, TR4,[4] and RCV1-4 [68] are text corpora. Pokerhand[5] is an evolutionary data and CoverType[6] is collected to predict forest cover type based on cartographic variable. Table II summarizes the statistics of these datasets. MNIST, CoverType, and Pokerhand are examined in the out-of-sample problem.

B. Comparison Methods

For a fair comparison, we select five representative scalable clustering methods.

ALRR: An accelerated LRR algorithm published in 2018 [30].

KMM: A K-multiple-means method based on the partition of a bipartite graph published in 2019 [69].

ESSC: An efficient SSC algorithm published in 2020 [29].

FNC: A directly solving normalized cut method published in 2018 [70]. It has lower computational complexity and is fast for large-scale data.

SSCOMP: A popular SSC method based on orthogonal matching pursuit published in 2016 [28].

We tune the parameters in these methods to achieve the best performance. For our approach, clustering performance varies depending on the initialization of K-means. Thus, we run 20 times and use a different seed to initialize K-means at each time. We report the mean and standard deviation values. The widely used clustering accuracy (ACC), normalized mutual information (NMI), and Purity are employed to evaluate the clustering performance [23]. We conduct all experiments on a computer with a 2.6 GHz Intel Xeon CPU and 64-GB RAM, MATLAB R2016a. The source code of our method is publicly available.[7]

C. Results

Table III shows the clustering results of various methods. It can be seen that our proposed SGL outperforms other state-of-the-art techniques in most cases. In particular, our method performs much better than the most two recent methods KMM and ESSC. In terms of ACC, NMI, and Purity, SGL improves KMM by 12.57%, 15.06%, and 14.49% in average, respectively. With respect to ESSC, our gain is 16.66%, 16.19%, and 12.17%, respectively. Among the five competitive methods, ALRR gives more stable performance than others; SSCOMP is quite unstable, which could be caused by the fact that its strong assumptions are often broke facing real-world data. FNC generates mediocre results.

TABLE III
CLUSTERING PERFORMANCE ON SIX DATASETS. THE BEST PERFORMANCE IS HIGHLIGHTED. TIME IS MEASURED IN SECONDS

Data	Metric	ALRR	KMM	ESSC	FNC	SSCOMP	SGL
BA	ACC	39.03	41.45	28.56	46.79	18.66	**48.51(0.52)**
	NMI	57.21	55.24	40.09	60.13	30.66	**60.47(0.22)**
	PURITY	53.20	45.47	32.76	**60.13**	21.36	53.81(0.11)
	TIME	8.07	0.78	18.30	1.68	0.31	11.63(0.12)
ORL	ACC	68.00	61.25	60.75	59.25	63.75	**68.83(0.39)**
	NMI	**82.73**	77.16	77.29	77.33	80.44	81.90(0.15)
	PURITY	77.00	65.00	67.50	64.25	67.50	**77.07(0.73)**
	TIME	4.55	0.24	4.69	0.40	0.19	3.23(0.15)
TR11	ACC	72.94	49.75	52.90	54.58	64.97	**75.31(0.86)**
	NMI	63.08	27.94	44.18	41.35	55.59	**67.66(0.59)**
	PURITY	75.84	51.93	57.73	41.35	78.01	**80.32(1.67)**
	TIME	4.51	0.15	25.53	1.69	3.19	5.31(0.08)
TR41	ACC	61.73	66.05	67.31	58.42	69.47	**77.79(0)**
	NMI	62.87	61.92	65.57	49.96	66.73	**72.11(0)**
	PURITY	73.00	72.09	73.35	49.96	80.52	**84.16(0)**
	TIME	7.75	0.36	74.74	6.83	83.36	7.63(0.23)
TR45	ACC	72.31	73.80	67.25	48.40	73.04	**74.41(2.44)**
	NMI	**70.10**	67.73	60.85	31.22	69.96	69.53(1.82)
	PURITY	77.68	82.34	68.84	31.22	**83.76**	76.74(1.71)
	TIME	9.41	0.35	62.25	4.83	87.64	6.34(0.18)
RCV1-4	ACC	66.79	47.63	39.11	65.37	30.35	**70.52(0)**
	NMI	40.76	17.12	12.30	35.00	0.10	**45.76(0)**
	PURITY	78.79	47.63	78.21	35.00	30.4	**79.37(0)**
	TIME	1175.80	4.1	4078.2	85.90	350.62	98.86(0.67)

We also list the consumed time for each method in Table III. We can observe that our method achieves comparable efficiency on those small size datasets, including AR, ORL, TR11, TR41, and TR45. Though KMM demonstrates high efficiency, its clustering performance is degraded. SSC-based methods, that is, ESSC and SSCOMP, cost much more time than others on TR11, TR41, and TR45. For medium size data RCV1-4, ALRR and ESSC run a long time. For example, ESSC takes 4078 s to finish the RCV1-4 data, while our method just needs 98.86 s. This indicates that our method is efficient yet effective.

D. Parameter Analysis

There are three parameters α and β and anchor number m in our model. We can observe that the dictionary A is constituted of anchor points, which is important to the proposed model. Recently, some advanced anchor points selection methods are developed. For example, Abdolali *et al.* [71] proposed to select a few sets of anchors points based on randomized hierarchical clustering. Consequently, it selects a different set of anchors for each data point. You *et al.* [72] adopted an exemplar selection method FFS to find a subset that best reconstructs all data points based on the ℓ_1 norm of the representation coefficients. Compared to them, our adopted K-means approach is simple

[1] http://www.cs.nyu.edu/roweis/data.html
[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[3] http://yann.lecun.com/exdb/mnist/
[4] http://www-users.cs.umn.edu/han/data/tmdata.tar.gz
[5] https://archive.ics.uci.edu/ml/datasets/Poker+Hand
[6] http://archive.ics.uci.edu/ml/datasets/covertype
[7] https://github.com/sckangz/SGL

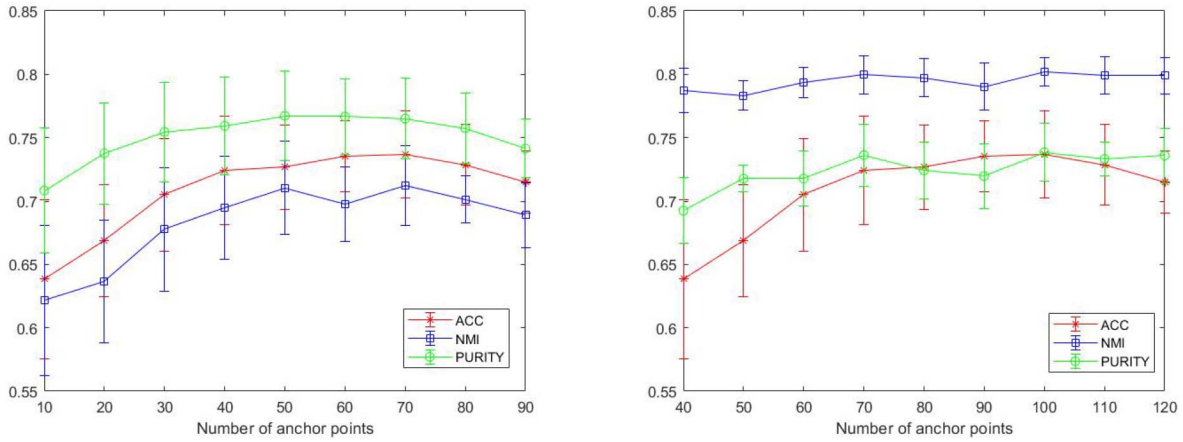

Fig. 2. Impact of anchor number and initialization on TR45 (the left) and ORL (the right) data sets. The average performance and error bar are displayed.

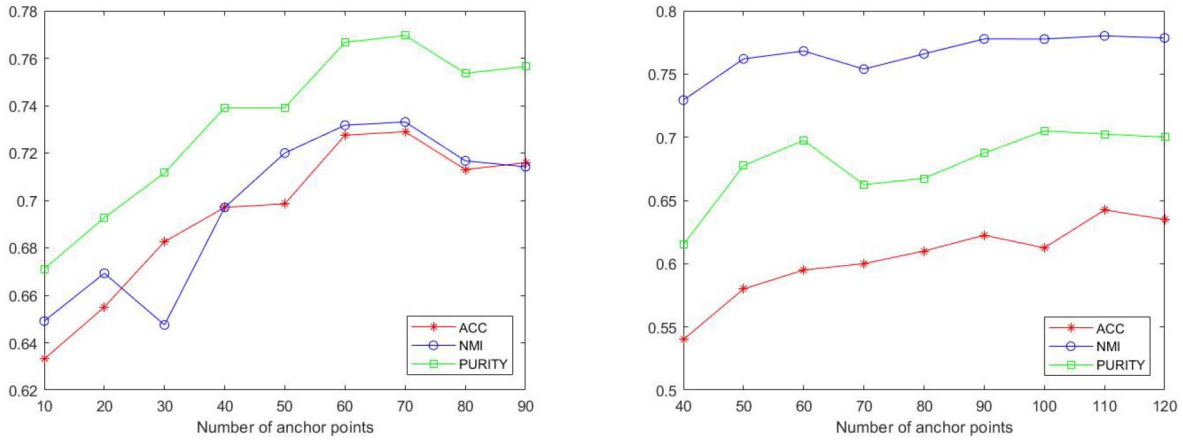

Fig. 3. Performance of SGL on TR45 (the left) and ORL (the right) datasets, where anchors are chosen by FFS.

and straightforward. However, K-means is sensitive to initialization. Therefore, we discuss the clustering effect caused by the variations of anchor number and initialization. It is reasonable to assume that we at least need k anchors to reveal the underlying subspaces. In Fig. 2, we let the number of anchor points vary over the range $[40, \ldots, 120]$ and $[10, \ldots, 90]$ for ORL and TR45, respectively. It shows that the clustering performance indeed changes along with the variation of anchor number and initialization. However, it can be seen that we do not require too many anchors for a good performance. Furthermore, the standard variation suggests that our results change slightly due to the initialization.

Furthermore, we adopt FFS [72] to select anchors and use them to build an anchor matrix A. Then, we plot the performance of our method in Fig. 3. We can see that its performance is inferior to the K-means-based approach. This could be caused by the fact that these anchors are close to a few cluster centroids. Thus, K-means is more appropriate than FFS. FFS could well handle the case when anchors lie close to a union of subspaces [72].

Next, we fix the number of anchor points and analyze the sensitive of α and β on TR45 and ORL datasets. β varies over the range of $[0.0001, 0.001, 0.01, 0.1, 1, 10]$. α varies over $[0.001, 0.01, 0.1, 1, 10, 50]$ and $[0.1, 1, 10, 20, 30, 40, 50]$ on TR45 and ORL, respectively. Fig. 4 displays how the clustering results of our method vary with α and β. We can find that the performance of our method is very stable with respect to a large range of α and β values. In practice, we can fix α and tune β.

E. Out-of-Sample Experiment

In this section, we conduct experiments on MNIST, CoverType, and Pokerhand to evaluate our method for addressing the out-of-sample problem. Following the setting in SLSR [32], we randomly select 1000 data points as in-sample data for CoverType and Pokerhand, while 2000 data points for MNIST. The rest data are regarded as out-of-sample data for testing. As described in Section III-B, we apply the kNN algorithm to test out-of-sample data. Concretely, we adopt 3NN and 1NN to the anchor points and their labels to predict the clusters of test data. As a baseline, we also apply 1NN to the in-sample data. Moreover, we compare with SLSR [32], to our best knowledge, which is the only method proposed to address the out-of-sample problem for subspace clustering.

We report the results in Table IV. As for the accuracy, we can find that our approach consistently outperforms other methods. In particular, our accuracy surpasses the SLSR

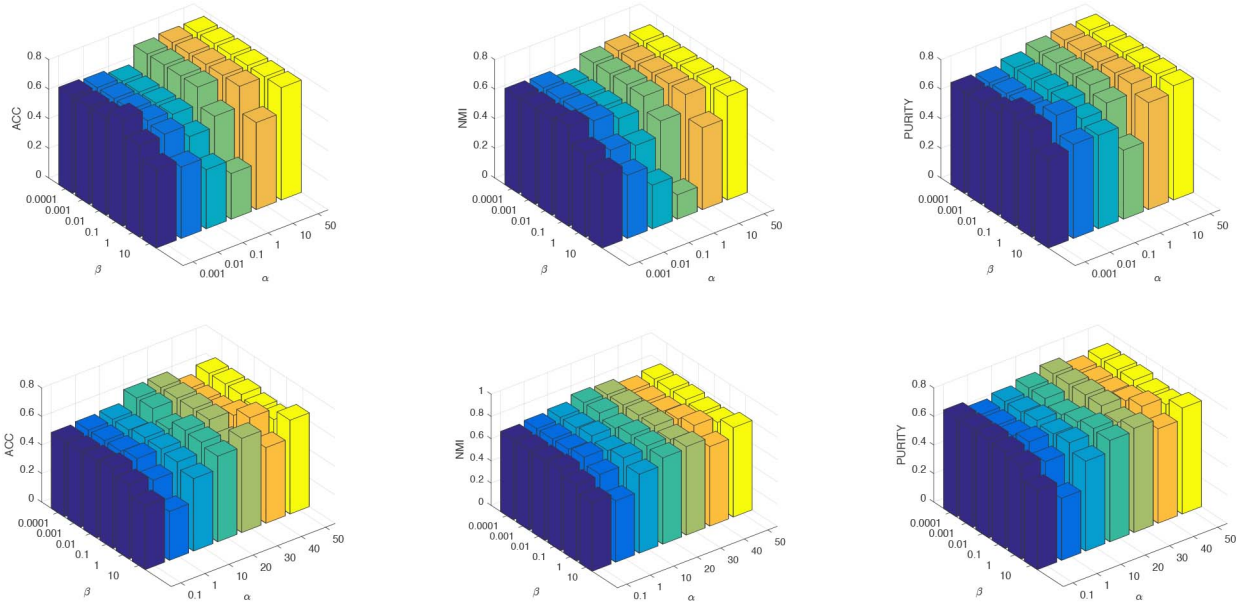

Fig. 4. Sensitivity demonstration of parameters α and β for our method over TR45 (the first row) and ORL (the second row) datasets.

TABLE IV
CLUSTERING PERFORMANCE ON OUT-OF-SAMPLE PROBLEM

Data	Method	Acc	NMI	PURITY	Time (s)
MNIST	3NN+anchor points	57.23(1.31)	53.72(0.84)	60.57(0.64)	1.04(0.02)
	1NN+anchor points	**58.23(0.35)**	**54.26(0.24)**	**62.37(0.19)**	0.67(0.01)
	1NN+in-sample data	56.49(0.75)	51.91(0.79)	59.61(0.96)	22.69(0.66)
	SLSR	52.26	47.72	57.06	252.28
CoverType	3NN+anchor points	**39.01(0.92)**	**8.57(0.78)**	50.15(0.41)	3.05(0.04)
	1NN+anchor points	32.10(1.48)	6.69(0.52)	50.21(0.13)	0.64(0.01)
	1NN+in-sample data	31.41(1.21)	6.47(0.57)	**52.36(0.28)**	8.24(0.11)
	SLSR	26.5	7.3	49.42	178.72
Pokerhand	3NN+anchor points	**27.97(1.71)**	0.27(0.10)	50.73(0.45)	8.81(0.07)
	1NN+anchor points	22.15(1.48)	**0.33(0.10)**	**50.90(0.47)**	0.88(0.01)
	1NN+in-sample data	19.73(1.36)	0.17(0.10)	50.23(0.38)	2.34(0.12)
	SLSR	15.82	0.06	50.20	284.52

method by about 10% on CoverType and Pokerhand data. Our method obtains comparable performance with baseline on NMI and Purity. These results suggest that our anchor points can well represent the structure of the raw data. We can also see that the number of neighbors in kNN has a big influence to the final performance. This confirms the previous observation in the literature.

In terms of running time, we can observe that our approach outperforms others by a large margin and can finish 1M samples in less than 1 s. Compared to SLSR, our method is at least 200 times faster. With respect to the conventional kNN approach, our method also runs much faster since we use fewer points. In summary, our method is a promising approach to process out-of-sample problem. On the other hand, this approach demonstrates itself to be an alternative way to tackle big data challenge.

VII. MULTIVIEW EXPERIMENTS

In this section, we assess the effectiveness and efficiency of our multiview model (16).

TABLE V
STATISTICAL INFORMATION OF THE MULTIVIEW DATASETS. THE NUMBER IN PARENTHESIS DENOTES DIMENSION

View	Caltech-7	Citeseer	NUS
1	Gabor (48)	Content (3703)	Color Histogram (65)
2	Wavelet moments (40)	Citation (3312)	Color moments (226)
3	CENTRIST (254)	-	Color correlation (145)
4	HOG (1984)	-	Edge distribution (74)
5	GIST (512)	-	Wavelet texture (129)
6	LBP (928)	-	-
Data points	1474	3312	30000
Class number	7	6	31

A. Setup

We perform experiments on three benchmark datasets: 1) Caltech-7[8]; 2) NUS[9]; and 3) Citeseer.[10] Both Caltech-7 and NUS are object recognition database, while Citeseer is a document data with content and citations. The details of them are summarized in Table V. We compare our proposed MSGL method with four other state-of-the-art multiview methods.

AMGL [73] is a popular multiview spectral clustering method proposed in 2016. Though it is parameter free, it has a high complexity since it involves SVD implemented on $n \times n$ matrix in each iteration.

MLRSSC [59] is a multiview subspace clustering method developed in 2018. It has good performance since it combines low-rank and sparse model, but it has $\mathcal{O}(n^3)$ complexity.

MSC_IAS [61] is proposed to learn a better graph in latent space in 2019. It surpasses a number of multiview subspace clustering methods, but it also has $\mathcal{O}(n^3)$ complexity.

LMVSC [54] is a linear multiview subspace clustering method published in 2020. It shows superior performance and high efficiency.

[8]http://www.vision.caltech.edu/ImageDatasets/Caltech101/

[9]https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

[10]http://lig-membres.imag.fr/grimal/data.html

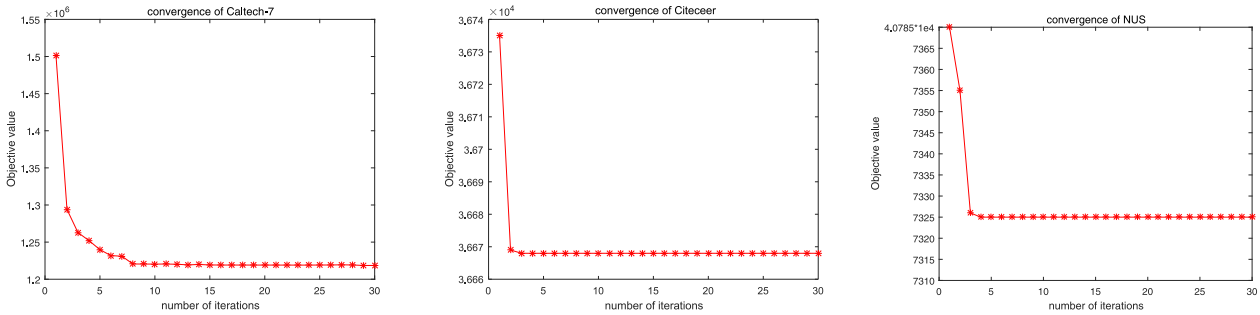

Fig. 5. Evolution of objective function value of (16) on all multiview datasets.

TABLE VI
CLUSTERING PERFORMANCE ON CALTECH-7 DATA

Method	ACC	NMI	PURITY	Time (s)
AMGL	45.18	42.43	46.74	20.12
MLRSSC	37.31	21.11	41.45	22.26
MSC_IAS	39.76	24.55	44.44	57.18
LMVSC	72.66	51.93	75.17	135.79
MSGL	**73.31(0.96)**	**52.47(2.23)**	**77.34(2.87)**	395.63(21.33)

TABLE VII
CLUSTERING PERFORMANCE ON CITESEER DATA

Method	ACC	NMI	PURITY	Time (s)
AMGL	16.87	0.23	16.87	449.07
MLRSSC	25.09	02.67	63.70	106.1
MSC_IAS	34.11	11.53	**80.76**	191.29
LMVSC	52.26	25.71	54.46	21.33
MSGL	**54.47(0.78)**	**26.54(0.94)**	57.49(1.13)	62.23(4.72)

TABLE VIII
CLUSTERING PERFORMANCE ON NUS DATA

Method	ACC	NMI	PURITY	Time (s)
MSC_IAS	15.48	**15.21**	16.75	45386
LMVSC	15.53	12.95	19.82	165.39
MSGL	**16.31(0.42)**	12.26(0.28)	**20.51(0.37)**	547.58(32.11)

We use grid search to find the best parameters for all methods. For MSGL, γ is searched from $[-1, -2, -3, -4, -5]$.

B. Results

Tables VI–VIII report the clustering performance on the three datasets. Both AMGL and MLRSSC raise out-of-memory exception on the NUS data. We can observe that our method outperforms other methods in most cases, including our closest competitor LMVSC. In particular, our method MSGL constantly outperforms LMVSC on accuracy and Purity. In terms of NMI, our method also achieves comparable or even better performance than LMVSC. AMGL, MLRSSC, and MSC_IAS produce poor accuracy and NMI on Caltech-7 and Citeseer datasets. There are some possible causes for this. AMGL uses losses of different views to weight each view, which might not be flexible to distinguish the contributions of diverse views. Moreover, some real-world datasets cannot be simply characterized by low rank or sparse structure as used in MLRSSC and MSC_IAS. In contrast, we directly consider the cluster structure, which is a target-oriented solution.

For running time comparison, the proposed method is usually faster than the baseline methods except LMVSC. Though both MSGL and LMVSC have linear time complexity, LMVSC is faster since it is iteration free. To be precise, LMVSC does not consider the structure of graph and distinguish views, so it is a one-pass approach. For large-scale data NUS, MSC_IAS costs about 12 h, while our method can finish it in 10 min. Hence, in terms of efficiency and effectiveness, MSGL and LMVSC are appealing in practice applications.

C. Convergence Analysis

As mentioned earlier, MSGL is a convergent algorithm. To verify this, we demonstrate the behavior of the objective value of (16) in Fig. 5. It shows that our algorithm converges within ten iterations on all three real-world datasets. Once again, this supports that our method is efficient.

VIII. CONCLUSION

In this article, we proposed a novel graph-based subspace clustering framework to cope with single view and multiview data. We simultaneously consider graph structure, scalability, and out-of-sample problems by making use of the anchor idea, bipartite graph, and spectral graph property. Consequently, a graph with explicit cluster structure is learned in linear complexity. Theoretical analysis builds the connection between our method and K-means clustering. Extensive experimental results demonstrate that our method can reduce time complexity without sacrificing the clustering performance. In the future, we plan to investigate the new anchor selection strategy to improve the stableness of the proposed approach.

REFERENCES

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
[2] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Adv. Neural Inf. Proc. Syst.*, 2001, pp. 849–856.
[3] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, pp. 226–231.
[5] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4857–4868, Nov. 2020.
[6] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.
[7] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[8] G. Liu, Z. Zhang, Q. Liu, and H. Xiong, "Robust subspace clustering with compressed data," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5161–5170, Oct. 2019.

[9] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2988–3001, Jun. 2017.

[10] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[11] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[12] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 347–360.

[13] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 24–33.

[14] J. Zhang *et al.*, "Self-supervised convolutional subspace clustering network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5473–5482.

[15] X. Zhu, Y. Zhu, and W. Zheng, "Spectral rotation for deep one-step clustering," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107175, doi: 10.1016/j.patcog.2019.107175.

[16] Z. Kang, C. Peng, and Q. Cheng, "Twin learning for similarity and clustering: A unified kernel approach," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2080–2086.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] Z. Kang, H. Pan, S. C. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1833–1843, May 2020.

[19] Z. Ren, S. X. Yang, Q. Sun, and T. Wang, "Consensus affinity graph learning for multiple kernel clustering," *IEEE Trans. Cybern.*, early access, Jun. 25, 2020, doi: 10.1109/TCYB.2020.3000947.

[20] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.

[21] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nystrom method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.

[22] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4129–4138.

[23] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 313–318.

[24] X. Chen, R. Chen, Q. Wu, Y. Fang, F. Nie, and J. Z. Huang, "LABIN: Balanced min cut for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31 no. 3, pp. 725–736, Mar. 2020.

[25] A. Adler, M. Elad, and Y. Hel-Or, "Linear-time subspace clustering via bipartite graph modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2234–2246, Oct. 2015.

[26] S. Wang, B. Tu, C. Xu, and Z. Zhang, "Exact subspace clustering in linear time," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2113–2120.

[27] J. Li, H. Zhao, Z. Tao, and Y. Fu, "Large-scale subspace clustering by fast regression coding," in *Proc. IJCAI*, 2017, pp. 2138–2144.

[28] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 3918–3927.

[29] F. Pourkamali-Anaraki, J. Folberth, and S. Becker, "Efficient solvers for sparse subspace clustering," *Signal Process.*, vol. 172, Jul. 2020, Art. no. 107548.

[30] J. Fan, Z. Tian, M. Zhao, and T. W. S. Chow, "Accelerated low-rank representation for subspace clustering and semi-supervised classification on large-scale data," *Neural Netw.*, vol. 100, pp. 39–48, Apr. 2018.

[31] S. Xiao, W. Li, D. Xu, and D. Tao, "FaLRR: A fast low rank representation solver," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4612–4620.

[32] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2499–2512, Dec. 2016.

[33] L. Huang, C.-D. Wang, H.-Y. Chao, and P. S. Yu, "MVStream: Multiview data stream clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3482–3496, Sep. 2020.

[34] P. Zhou, Y.-D. Shen, L. Du, F. Ye, and X. Li, "Incremental multi-view spectral clustering," *Knowl. Based Syst.*, vol. 174, pp. 73–86, Jun. 2019.

[35] H. Tao, C. Hou, D. Yi, J. Zhu, and D. Hu, "Joint embedding learning and low-rank approximation: A framework for incomplete multiview learning," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1690–1703, Mar. 2021.

[36] C. Tang *et al.*, "Learning a joint affinity graph for multiview subspace clustering," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1724–1736, Jul. 2019.

[37] 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, Oct. 2017.

[38] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 851–864, Mar. 2019.

[39] S. Sun, Y. Liu, and L. Mao, "Multi-view learning for visual violence recognition with maximum entropy discrimination and deep features," *Inf. Fusion*, vol. 50, pp. 43–53, Oct. 2019.

[40] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1984–1993, Oct. 2019.

[41] M.-S. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3513–3520.

[42] S. Yao, G. Yu, J. Wang, C. Domeniconi, and X. Zhang, "Multi-view multiple clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4121–4127.

[43] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, Jun. 2020. [Online]. Available: https://doi.org/10.1109/TKDE.2019.2903810

[44] 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, Sep. 2017.

[45] 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. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2153–2159.

[46] Y. Wang, "Survey on deep multi-modal data analytics: Collaboration, rivalry and fusion," 2020. [Online]. Available: arXiv:2006.08159

[47] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[48] X. Zhang, H. Sun, Z. Liu, Z. Ren, Q. Cui, and Y. Li, "Robust low-rank kernel multi-view subspace clustering based on the schatten p-norm and correntropy," *Inf. Sci.*, vol. 477, pp. 430–447, Mar. 2019.

[49] Y. Chen, X. Xiao, and Y. Zhou, "Jointly learning kernel representation tensor and affinity matrix for multi-view clustering," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1985–1997, Aug. 2020.

[50] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 586–594.

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

[52] H. Tao, C. Hou, Y. Qian, J. Zhu, and D. Yi, "Latent complete row space recovery for multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 29, pp. 8083–8096, 2020.

[53] Z. Kang *et al.*, "Partition level multiview subspace clustering," *Neural Netw.*, vol. 122, pp. 279–288, Feb. 2020.

[54] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proc. AAAI*, 2020, pp. 4412–4419.

[55] Z. Kang, X. Lu, Y. Lu, C. Peng, W. Chen, and Z. Xu, "Structure learning with similarity preserving," *Neural Netw.*, vol. 129, pp. 138–148, Sep. 2020.

[56] Z. Kang, X. Lu, J. Liang, K. Bai, and Z. Xu, "Relation-guided representation learning," *Neural Netw.*, vol. 131, pp. 93–102, Nov. 2020.

[57] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[58] B. D. Haeffele and R. Vidal, "Structured low-rank matrix factorization: Global optimality, algorithms, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1468–1482, Jun. 2020.

[59] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognit.*, vol. 73, pp. 247–258, Jan. 2018.

[60] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.

[61] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, and S. Z. Li, "Multi-view subspace clustering with intactness-aware similarity," *Pattern Recognit.*, vol. 88, pp. 50–63, Apr. 2019.
[62] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, "Scalable semi-supervised learning by efficient anchor graph regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1864–1877, Jul. 2016.
[63] J. Han, K. Xiong, and F. Nie, "Orthogonal and nonnegative graph reconstruction for large scale clustering," in *Proc. IJCAI*, 2017, pp. 1809–1815.
[64] F. R. Chung and F. C. Graham, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1997.
[65] Z. Kang *et al.*, "Structured graph learning for clustering and semi-supervised classification," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107627.
[66] X. Wang and Z. Kang, "Smooth representation semi-supervised classification," *Comput. Sci.*, vol. 48, no. 3, pp. 124–129, 2021.
[67] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural Parallel Sci. Comput.*, vol. 11, no. 4, pp. 351–368, 2003.
[68] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Apr. 2004.
[69] F. Nie, C.-L. Wang, and X. Li, "K-multiple-means: A multiple-means clustering method with specified k clusters," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining.*, 2019, pp. 959–967.
[70] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining.*, 2018, pp. 1206–1215.
[71] M. Abdolali, N. Gillis, and M. Rahmati, "Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs," *Signal Process.*, vol. 163, pp. 166–180, Oct. 2019.
[72] C. You, C. Li, D. P. Robinson, and R. Vidal, "Self-representation based unsupervised exemplar selection in a union of subspaces," 2020. [Online]. Available: arXiv:2006.04246
[73] 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. IJCAI*, 2016, pp. 1881–1887.

Zhao Kang received the Ph.D. degree in computer science from Southern Illinois University Carbondale, Carbondale, IL, USA, in 2017.

He is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has published over 60 research papers in top-tier conferences and journals, including AAAI, IJCAI, ICDE, CVPR, SIGKDD, ICDM, CIKM, SDM, ACML, IEEE TRANSACTIONS ON CYBERNETICS, *ACM Transactions on Intelligent Systems and Technology*, *ACM Transactions on Knowledge Discovery from Data*, *Neural Networks, and Pattern Recognition*, and *Public Relations Review*. His research interests are machine learning, pattern recognition, and deep learning.

Dr. Kang has been a SPC/PC Member or a Reviewer for a number of top conferences, such as AAAI, IJCAI, CVPR, ICCV, MM, ICDM, CIKM, and ECCV. He regularly serves as a Reviewer for the *Journal of Machine Learning Research*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE TRANSACTIONS ON MULTIMEDIA.

Zhiping Lin received the B.S. degree in electrical engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019, where he is currently pursuing the M.S. degree in computer science and engineering.

His current research interests include graph neural networks and clustering.

Xiaofeng Zhu (Senior Member, IEEE) received the Ph.D. degree in computer science from The University of Queensland, Australia.

He is a Professor with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include large-scale multimedia retrieval, feature selection, sparse learning, data preprocess, and medical image analysis.

Wenbo Xu received the B.S. degree in biomedical engineering from Chongqing University, Chongqing, China, in 1994, the M.S. degree in communication and information systems from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 1998, and the Ph.D. degree in cartography and geographic information systems from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2004.

From 2004 to 2012, he was a Lecturer and an Associate Professor with the School of Automation Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. From 2013 to 2014, he was a Visiting Scholar with the Department of Geography, University of Iowa, Iowa City, IA, USA. Since 2014, he has been a Professor with the School of Resources and Environment, UESTC. He has authored a book, over 50 articles, more than ten inventions, and holds three patents. His research interests include environmental remote sensing, image processing and applications, and geographic information systems.

Dr. Xu was a recipient of the First prize of Chinese Science and Technology Progress Award of Surveying and Mapping, in 2019 and the First Prize of Science and Technology Progress Award of Surveying and Mapping Geographic Information of Sichuan Province in 2017.