Scalable Multi-view Subspace Clustering with Unified Anchors

Anonymous Author(s)
Submission Id: 1866

ABSTRACT

Multi-view subspace clustering has received widespread attention to effectively fuse multi-view information among multimedia applications. Considering that most existing approaches' cubic time complexity makes it challenging to apply to realistic large-scale scenarios, some researchers have addressed this challenge by sampling anchor points to capture distributions in different views. However, the separation of the heuristic sampling and clustering process leads to weak discriminate anchor points. Moreover, the complementary multi-view information has not been well utilized since the graphs are constructed independently by the anchors from the corresponding views. To address these issues, we propose a Scalable Multi-view Subspace Clustering with Unified Anchors (SMVSC). To be specific, we combine anchor learning and graph construction into a unified optimization framework. Therefore, the learned anchors can represent the actual latent data distribution more accurately, leading to more discriminative clustering structure. Most importantly, the linear time complexity of our proposed algorithm allows the multi-view subspace clustering approach to be applied to large-scale data. Then, we design a four-step alternative optimization algorithm with proven convergence. Compared with state-of-the-art multi-view subspace clustering methods and large-scale oriented methods, the experimental results on several datasets demonstrate that our SMVSC method achieves comparable or better clustering performance much more efficiently.

CCS CONCEPTS

• **Computing methodologies → Cluster analysis**; • **Theory of computation → Unsupervised learning and clustering**.

KEYWORDS

Multi-view clustering, subspace clustering, scalable graph clustering

ACM Reference Format:

Anonymous Author(s). 2021. Scalable Multi-view Subspace Clustering with Unified Anchors. In *Proceedings of ACM Conference (MM'21).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Multi-view clustering, which integrates the diverse and complementary information among views for clustering, is an important unsupervised learning method in the machine learning and multimedia data mining community [16, 19, 22, 24, 28, 31]. Many multi-view clustering algorithms have been proposed in existing literature, among which multi-view subspace clustering is quite popular [18, 21, 25, 34, 39]. Multi-view subspace clustering (MVSC) aims to seek unified subspaces from fused multi-view data representation and then separates data in the corresponding subspace with the two-step learning schedules: i) graph construction: obtaining low-dimensional subspace representation from the given multi-view data, and ii) spectral clustering: performing spectral clustering on the respective fused graph. By capturing nonlinear structure and preserving pairwise similarity in graphs, MVSC has been widely applied in various applications, e.g. image classification[9, 29, 37], face clustering [32, 38], community detection [12, 13, 30].

Although existing MVSC approaches achieve great success in improving clustering performance, one major drawback of MVSC from being further applied is the cubic time complexity respecting the sample number n. The first graph construction stage needs to solve n convex quadratic program subproblems whose time complexity is at least $O(n^3)$ per iteration. Moreover, the second spectral clustering process needs $O(n^3)$ for Singular Value Decomposition (SVD). Therefore, designing scalable MVSC algorithms to handle large-scale multi-view data remains to be an open question.

In recent years, anchor-based MVSC has been proposed to alleviate the high complexity of the traditional subspace method [6, 11, 15]. By independently sampling k selected landmarks, the original global graphs with size $n \times n$ are replaced with the corresponding anchor graphs with size $n \times k$. The anchor graphs are equally weighted fused into the consensus graph and then spectral clustering is performed to get the final clustering result. The whole time complexity of anchor-based subspace approaches is reduced to $O(n)$ per iteration and can be applied to large-scale tasks. However, as can be seen from Figure. 2, existing anchor-based multi-view subspace clustering strategy can be further improved with the following considerations. Firstly, each view's anchor points are independently generated by k-means clustering or random sampling,

Figure 1: Multi-view learning is known as fusing data from multimedia or multiple features. For examples, multiple features can be extract from an image to form multi-view data. Text can be presented in different languages as different views. The video, corresponding voice and content description make up multi-view cross modal representations.

MM'21, OCT 2021, Chengdu, China

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

which are isolated from other views. Moreover, the separation of the heuristic sampling and graph construction process leads to weak discriminate anchor points. As a result, the selected anchors may not reflect the actual data distribution and generate imprecise graph structures. Secondly, the complementary multi-view information has not been well utilized without sufficient information fusion since the graphs are constructed independently by the anchors from the corresponding views. Both of the above limitations of existing methods adversely degrade the clustering performance.

To address the above issues, we propose a novel anchor-based multi-view subspace clustering method in this paper, termed Scalable Multi-view Subspace Clustering with Unified Anchors (SMVSC). To be specific, we combine anchor learning and graph construction into a unified framework in which the consensus anchors are jointly optimized with respective view permutation matrices. Therefore, the learned anchors can accurately represent the actual latent data distribution leading to better graph structure construction. The importance of each view is also adaptively measured by the individual view contribution to the unified graph. Most importantly, inheriting from anchor strategy, our proposed algorithm's linear time complexity allows it to be applied to large-scale multi-view data. Then, a four-step alternative optimization algorithm with proven convergence is proposed to solve the resulting optimization problem. Compared with state-of-the-art multi-view subspace clustering methods and large-scale oriented methods, the experimental results on several datasets demonstrate that our SMVSC method achieves comparable or better clustering performance much more efficiently. The main contributions of this paper can be summarized as follows.

(1) Different from traditional heuristic anchor sampling strategy, we integrate anchor learning and graph construction into a unified framework. The two counterparts contribute to each other and are jointly optimized together so that the learned anchors can capture the actual data distribution more accurately. Therefore, the subspace graph structure is obtained with more discrimination and further improve the clustering performance.
(2) Comparing to the existing anchor graph fusion strategy, we adaptively utilize the unified anchor graph to capture the complementary information across views and simultaneously learn the importance of different views. More importantly, our proposes method with linear time complexity is proved to be more efficient and effective to large-scale subspace clustering problems.
(3) We design an alternating optimization algorithm to solve the resulting optimization problem with proven convengence. Extensive experimental results demonstrate the superiority of our clustering performance and running time. Furthermore, to the best of our knowledge, compared to traditional multi-view subspace-based clustering methods, we are the first to run MVSC on more than 100,000 samples both efficiently and effectively.

In the rest of the paper, we introduce the related work in Section 2. Then, we depict the proposed method and its optimization process in Section 3. All the experimental results are displayed and analyzed in Section 4. At the end of the paper, we conclude and propose a prospect in Section 5.

2 RELATED WORK

2.1 Notations

In this paper, bold uppercase letters such as $\mathbf{A}, \mathbf{B}, \mathbf{C}$ represent matrices. $\|\cdot\|_F^2$ denotes the square of Frobenius norm. $\mathbf{I}_n$ represents the n-dimension identity matrix. $\mathbf{1}$ represents the vector with all elements being one. We denote a matrix with subscript i as the i-th view of matrix. Therefore, given a v views dataset $\{\mathbf{X}_i\}_{i=1}^{v} = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \cdots, \mathbf{x}_i^{(n)}] \in \mathbb{R}^{d_i \times n}$, where n is the number of samples, d_i is the view-specific dimension, $\mathbf{X}_i$ refers to the i-th view of original data matrix. Moreover, $\mathbf{X}_{i[:,j]}$ denotes the j-th column of original

2.2 Multi-view Subspace Clustering

As an extension of subspace clustering, multi-view subspace clustering assumes that data usually lie in underlying low-dimension subspaces rather than distribute uniformly in the entire space. Although many strategies are existing in subspace clustering, such as low-rank representation subspace clustering (LRR) [17], sparse subspace clustering (SSC) [7], most of the multi-view subspace clustering methods adopt self-representation to obtain the subspace representation.

Mathematically, given the multi-view data $\{\mathbf{X}_i\}_{i=1}^{v} \in \mathbb{R}^{d_i \times n}$ with d_i dimension feature in the i-th view, n the number of samples, the typical framework of multi-view subspace clustering can be expressed as follows,

$$\begin{aligned} \min_{\mathbf{S}} \sum_{i=1}^{v} \underbrace{\|\mathbf{X}_i - \mathbf{X}_i\mathbf{S}_i\|_F^2}_{Graph\ Construction} + \underbrace{\Omega(\mathbf{S}, \mathbf{S}_i)}_{Fusion}, \\ \text{s.t. } diag(\mathbf{S}_i) = 0, \mathbf{S}_i^{\top}\mathbf{1} = \mathbf{1}, \end{aligned} \tag{1}$$

where Ω refers to the consensus regularization term that could co-train a global graph among different views. After obtaining the fused global graph $\mathbf{S}$, the final clustering result can be reached by performing spectral clustering on $\mathbf{S}$.

Many multi-view subspace clustering methods are proposed along with this framework due to their ability to capture the global structure. However, Gao *et al.* [8] think it is unreasonable to align multi-view graphs into a unified one directly since the magnitude of $\mathbf{S}_i$ is different. They incorporate spectral clustering into their objective by using graphs of all views to obtain a uniform partition matrix. Considering that learning representation independently cannot ensure the complementary information, Cao *et al.* [4] propose to induce Hilbert-Schmidt Independence Criterion (HSIC) to explore diverse subspace representations. Zhang *et al.* [33, 35] reconstruct the data points in a latent space which could make subspace representation more accurate and robust. Since most of the multi-view subspace methods only explore diversity or consistency information, Luo *et al.* [20] propose to learn the subspace representation with consistency and specificity jointly. Some methods aim at finding a proper constraint for the subspace, such as low-rank or sparse or both [2]. Another theme in this field is fusing multiple information in partition level [10, 27, 36].

The approaches mentioned above have promising results in terms of clustering performance. However, regardless of the level

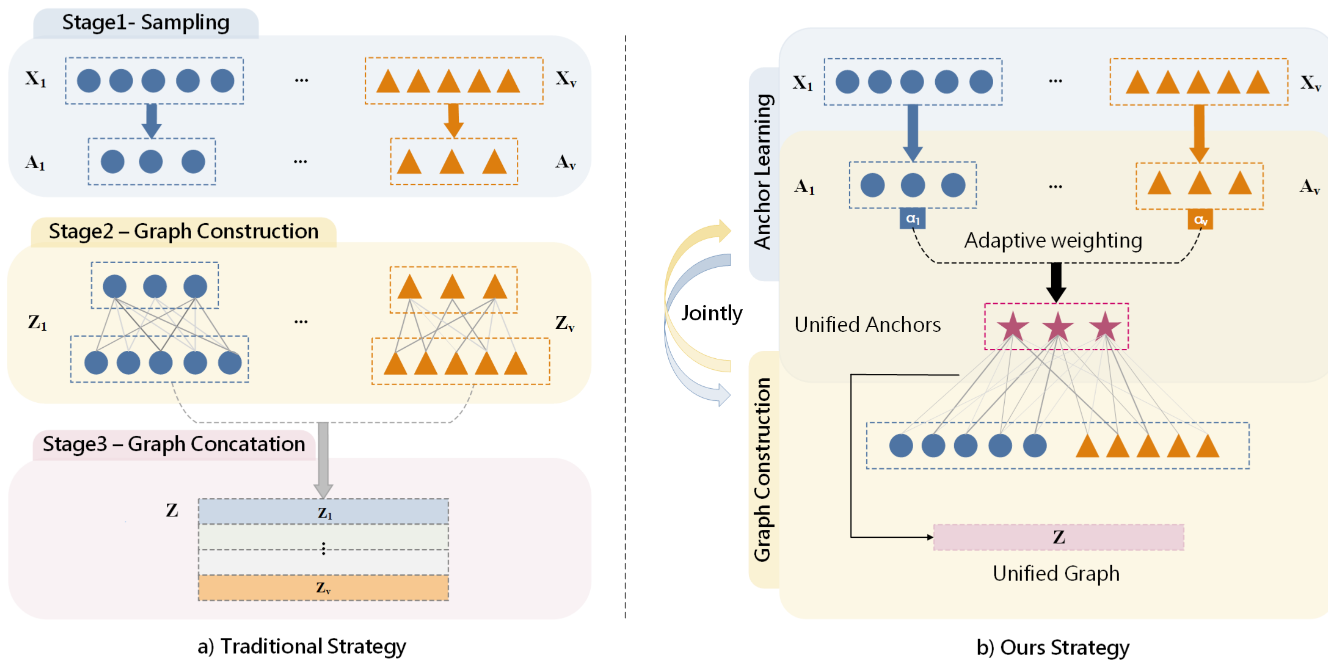


Figure 2: The framework of traditional anchor-based multi-view subspace strategy and our SMVSC strategy. In the first stage of traditional strategy, anchors are heuristically selected and then fixed by sampling from original data among each view. In the second stage, anchor graphs are constructed independently from each view without information exchange. At last, the view-specific anchor graphs are directly equally concatenated into a unified anchor graph. The three stages are independent of each other and there is no interaction between views. Unlike existing competitors, in our novel SMVSC framework, we adaptively learn a unified anchor graph considering the different importance of views. Most importantly, the anchor learning and graph construction are jointly optimized and boost each other to achieve a better clustering structure.

of fusion, multi-view subspace clustering cannot avoid performing graph construction, which requires a time complexity of $O(n^3)$ and a space complexity of at least $O(n^2)$. This dramatically limits the scalability of the multi-view subspace clustering approach.

2.3 Efficient Anchor-based Subspace Clustering

Sampling has been a long hotspot technique widely used in efficient multi-view spectral clustering [15] and multi-view subspace clustering [11], etc. In these methods, to avoid doing complex operations on computational and storage inefficient $n \times n$ affinity graphs, researchers select only a small number of the instances as anchors then efficiently learn a sub-graph $\mathbf{Z} \in \mathbb{R}^{m \times n}$ between the anchor points and the data, where m is the number of anchor points. As proved by these methods, the sampling operation can help reduce both storage and computational time while providing comparable clustering performance.

Recently, to extend the multi-view subspace clustering to large-scale datasets, Kang *et al.* [11] propose a Large-scale Multi-view Subspace Clustering in Linear Time (LMVSC) algorithm. They perform k-means to obtain the view-specific anchors and construct a small graph between anchors and all samples on each view. This strategy reaches good performance on several datasets. However, as shown in Fig. 2, in the LMVSC framework, the sampling procedure is isolated from the multi-view clustering process and the anchors do not participate in optimization. It is also performed independently in each view, leading to less discriminative anchor points and less complementary information across multiple views. In the next section, we propose a scalable multi-view subspace clustering with unified anchors to solve this problem.

3 METHODOLOGY

In this section, we introduce our proposed SMVSC method and give a detailed optimization procedure. An analysis of time complexity and space complexity is then conducted to illustrate the time and space efficiency of our method.

3.1 The Proposed method

Researchers utilize all the original data points to represent each point in the self-representation strategy, which is widely used in multi-view subspace clustering. Although a global relationship is well explored, the optimization time and storage cost related to the global graph restrict the scalability of multi-view subspace clustering. Besides, depicting one point with all samples is unnecessary and redundant. Therefore, we adopt the anchor strategy [11, 15] to select a small set of data points called anchor points or landmarks to reconstruct the underlying subspace and capture the manifold structure. In the existing literature, the selection of anchor points can be obtained by random sampling or uniform sampling from original data space or using the clustering centers obtained by performing k-means. However, in the previous strategy, anchors are fixed once they are initialized, making the anchor learning isolated from graph construction. Our algorithm integrates the two processes into a common framework, leading to more discriminative anchors. Moreover, generating anchors from independent views will result in different anchor sets, making the graph fusion difficult. And the complementary information among views is not well explored. In view of these issues, we adaptively learn a common graph by the projected unified anchors, resulting in a unified anchor graph with complementary view information and discriminative anchor structure. Mathematically, we formulate our objective function as follows,

$$\begin{aligned} &\min_{\boldsymbol{\alpha},\mathbf{W}_i,\mathbf{A},\mathbf{Z}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{Z}\|_F^2 + \|\mathbf{Z}\|_F^2, \\ &\text{s.t. } \boldsymbol{\alpha}^\top \mathbf{1} = 1, \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_d, \mathbf{A}^\top \mathbf{A} = \mathbf{I}_m, \mathbf{Z} \geq 0, \mathbf{Z}^\top \mathbf{1} = \mathbf{1}. \end{aligned} \tag{2}$$

In Eq. (2), $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ is the i-th view of original data where d_i is the dimension of corresponding view, n is the number of samples. $\mathbf{A} \in \mathbb{R}^{d \times m}$ is the unified anchor matrix where d is the common dimension across the view and m is the number of anchors. In this paper, we select k as the common dimension and choose the number of anchors $m \in \{k, 2k, 3k\}$. The common dimension, together with the orthogonal constraint, restricts $\mathbf{A}$ to be more discriminative. $\mathbf{W}_i$ is the anchor projection matrix from i-th view, which could project the unified anchor to corresponding original data space. $\mathbf{Z}$ is the unified anchor graph with $m \times n$ dimension.

According to the conclusion drawn from [1, 6, 11], the left singular vector of the anchor graph $\mathbf{Z}$ equals to that of complete graph $\mathbf{S} = \mathbf{Z}^\top \mathbf{Z}$. Therefore, we obtain the left singular vector $\mathbf{U}$ by performing SVD on $\mathbf{Z}$ and execute k-means on $\mathbf{U}$ to get the final clustering result.

3.2 Optimization

The optimization problem in Eq.(2) is not jointly convex when all variables are considered simultaneously. Therefore, we propose an alternating optimization algorithm to optimize each variable with the other variables been fixed. After that, we provide the total framework of optimization algorithm and time/space complexity analysis.

3.2.1 Update $\mathbf{W}_i$. When $\mathbf{A}$, $\mathbf{Z}$ and α_i are fixed, the objective function w.r.t. $\mathbf{W}_i$ can be formulated as

$$\min_{\mathbf{W}_i} \sum_{i=1}^{v} {\alpha_i}^2 \, \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{Z}\|_F^2, \text{ s.t. } \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_d. \tag{3}$$

Since each $\mathbf{W}_i$ is separated from each other in terms of corresponding views, thus we can transform Eq.(3) into the following equivalent problem by expanding the Frobenius norm by trace and removing the items that are not related to $\mathbf{W}_i$.

$$\max_{\mathbf{W}_i} \operatorname{Tr}(\mathbf{W}_i^\top \mathbf{B}_i), \text{ s.t. } \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_d, \tag{4}$$

where $\mathbf{B}_i = \mathbf{X}_i \mathbf{Z}^\top \mathbf{A}^\top$. Supposing the Singular value decomposition (SVD) result of $\mathbf{B}_i$ is $\mathbf{U}\Sigma\mathbf{V}^\top$, the optimal $\mathbf{W}_i$ can be easily obtained by calculating $\mathbf{U}\mathbf{V}^\top$ according to [26].

3.2.2 Update $\mathbf{A}$. With $\mathbf{W}_i$, $\mathbf{Z}$ and α_i being fixed, the optimization for $\mathbf{A}$ can be transformed into solving the following problem

$$\min_{\mathbf{A}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{Z}\|_F^2, \text{ s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_m. \tag{5}$$

Similar to the optimization of $\mathbf{W}_i$, the optimization of $\mathbf{A}$ in Eq.(5) equals to the following form

$$\max_{\mathbf{A}} \operatorname{Tr}(\mathbf{A}^\top \mathbf{C}), \text{ s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_m, \tag{6}$$

where $\mathbf{C} = \sum_{i=1}^{v} {\alpha_i}^2 \mathbf{W}_i^\top \mathbf{X}_i \mathbf{Z}^\top$. Similarly, the optimal solution of updating variable $\mathbf{A}$ can be attained the multiply of left singular matrix and the right singular matrix of $\mathbf{C}$.

3.2.3 Update $\mathbf{Z}$. Fixing other variables $\mathbf{W}_i$, $\mathbf{A}$ and α_i, the optimization problem for updating variable $\mathbf{Z}$ can be rewritten as,

$$\begin{aligned} &\min_{\mathbf{Z}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{Z}\|_F^2 + \|\mathbf{Z}\|_F^2, \\ &\text{s.t. } \mathbf{Z} \geq 0, \ \mathbf{Z}^\top \mathbf{1} = \mathbf{1}. \end{aligned} \tag{7}$$

The above optimization problem of $\mathbf{Z}$ can be easily formulated as the following Quadratic Programming (QP) problem

$$\begin{aligned} &\min \frac{1}{2} \mathbf{Z}_{:,j}^\top \mathbf{H} \mathbf{Z}_{:,j} + f^\top \mathbf{Z}_{:,j}, \\ &\text{s.t. } \mathbf{Z}_{:,j}^\top \mathbf{1} = 1, \ \mathbf{Z} \geq 0, \end{aligned} \tag{8}$$

where $\mathbf{H} = 2(\sum_{i=1}^{v} \alpha_i^2 + 1)\mathbf{I}$, $f^\top = -2\sum_{i=1}^{m} \mathbf{X}_{i[:,j]}^\top \mathbf{W}_i \mathbf{A}$. Optimization can be performed by solving the QP problem for each column of $\mathbf{Z}$.

3.2.4 Update α_i. Fixing the irrelevant variables, we can obtain the optimization problem for updating α_i.

$$\min_{\alpha_i} \sum_{i=1}^{v} \alpha_i^2 \mathbf{M}_i^2, \text{ s.t. } \boldsymbol{\alpha}^\top \mathbf{1} = 1, \tag{9}$$

where $\mathbf{M}_i = \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{Z}\|_F$. According to Cauchy-Buniakowsky-Schwarz inequality, the optimal α_i can be directly obtained by $\alpha_i = \frac{\mathbf{Q}}{\mathbf{M}_i}$, where $\mathbf{Q} = \frac{1}{\frac{1}{\mathbf{M}_1} + \frac{1}{\mathbf{M}_2} + \cdots + \frac{1}{\mathbf{M}_v}}$.

As the iteration proceeds, the four variables in the above optimization are solved separately with other variables fixed. Since each sub-problem is strictly convex, the objective value will decrease monotonically until the minimum is found or the convergence condition is reached. And the lower bound of the objective function can be easily proven to be zero. The entire procedures of the above optimization are listed in the following Algorithm 1.

3.3 Complexity Analysis

Firstly, we will analyze the time complexity during the total optimization. Then a comparison is conducted between the compared method in terms of main space complexity.

3.3.1 Time complexity. The computational complexity is composed of the cost of optimization of each variable. When updating $\mathbf{W}_i$, it cost $O(d_i d^2)$ to perform SVD on $\mathbf{B}_i$ and $O(d_i dk^2)$ to execute matrix multiplication to get the optimal $\mathbf{W}_i$. Similar to updating $\mathbf{W}_i$, updating $\mathbf{A}$ needs $O(md^2)$ and $O(dmk^2)$ for SVD and matrix multiplication. When solving the QP problem of updating $\mathbf{Z}$, it costs $O(nm^3)$ for all columns. The time cost of calculating α_i is

Algorithm 1 SMVSC

Input: Input v views dataset $\{\mathbf{X}_i\}_{i=1}^{v}$ and the number of cluster k.
Initialize: Initialize $\mathbf{A}$, $\mathbf{Z}$, $\mathbf{W}$ with zero matrix. Initialize α_i with $\frac{1}{v}$.

while not converged **do**
    Update $\mathbf{W}_i$ by solving the problem in Eq. (4).
    Update $\mathbf{A}$ by solving the problem in Eq. (6).
    Update $\mathbf{Z}$ by solving the problem in Eq. (8).
    Update α_i by solving Eq. (9).
end while
Obtain $\mathbf{U}$ by performing SVD on $\mathbf{Z}$.
Output: Perform k-means on $\mathbf{U}$ to achieve the final clustering result.

only $O(1)$. Therefore, the total time cost of the optimization process is $\sum_{i=1}^{v}(d_i d^2 + d_i dk^2) + md^2 + dmk^2 + nm^3$. Consequently, the computational complexity of our proposed optimization algorithm is linear complexity $O(n)$.

After the optimization, we perform SVD on $\mathbf{Z}$ to obtain its left singular matrix $\mathbf{U}$ and get the final clustering result by k-means. In the post-process, the computational complexity is $O(nm^2)$, which is also a linear complexity. Consequently, we achieve a linear-time algorithm in both optimization process and post-process. However, as shown in Table 1, most of the subspace-based multi-view clustering methods hold a $O(n^3)$ time complexity in the processes mentioned above.

3.3.2 Space complexity. In this paper, the major memory costs of our method are matrices $\mathbf{W}_i \in \mathbb{R}^{d_i \times k}$, $\mathbf{A} \in \mathbb{R}^{k \times m}$ and $\mathbf{Z} \in \mathbb{R}^{m \times n}$. Thus the space complexity of our SMVSC is $mn + (h+m)k$, where $h = \sum_{i=1}^{v} d_i$. In our algorithm, $m \ll n$ and $k \ll n$. Therefore, the space complexity of SMVSC is $O(n)$. We counted the major memory cost of the compared algorithms in the following Table 1. It is easy to observe that the space complexity of most state-of-the-art algorithms is $O(n^2)$, such as MVSC, AMGL, MLRSSC, FMR, etc. LMVSC method also performs $O(n)$ space complexity, but they have to construct a graph for each view, which will cost more than our consensus graph. The high time and space complexities limit the scalability of many multi-view subspace clustering, making them only applicable to relatively small datasets. We show in Table 1 the largest dataset reported in the comparison algorithm, which can reflect the efficiency of the algorithm to some extent.

Table 1: Complexity Analysis on compared methods

Method	Memory Cost	Time Complexity	Max Reported
RMKM	$(n+h)k$	$O(n)$	30,475
MVSC	$2vn^2 + nk$	$O(n^3)$	1,230
AMGL	$vn^2 + nk$	$O(n^3)$	12,613
MLRSSC	$(v+1)n^2$	$O(n^3)$	2,000
FMR	$n^2 + nm$	$O(n^3)$	10,158
PMSC	$2vn^2 + (v+1)nk$	$O(n^3)$	2,386
MLES	$n^2 + hm + mn$	$O(n^3)$	544
LMVSC	$vm(n+h)$	$O(n)$	30,000
Ours	$mn + (h+m)k$	$O(n)$	101,499

Table 2: Information of benchmark datasets

Dataset	Sample	View	Cluster	Feature
Caltech101-20	2386	6	20	48, 40, 254, 1984, 512, 928
CCV	6773	3	20	20, 20, 20
Caltech101-all	9144	5	102	48, 40, 254, 512, 928
SUNRGBD	10335	2	45	4096, 4096
NUSWIDEOBJ	30000	5	31	65, 226, 145, 74, 129
AwA	30475	6	50	2688, 2000, 252, 2000, 2000, 2000
YouTubeFace	101499	5	31	64, 512, 64, 647, 838

4 EXPERIMENT

In this section, we evaluate the clustering properties of the proposed method on seven widely used datasets. The performance of SMVSC is compared with six state-of-the-art multi-view subspace clustering methods and two large-scale oriented methods.

4.1 Datasets Information

The benchmark datasets used in this paper are introduced in Table 2 Caltech101-all and NUSWIDEOBJ are both object image datasets. CCV is a rich database of YouTube videos containing 20 semantic categories. The SUNRGBD dataset is densely annotated. The animal dataset with attributes is called AwA. YouTubeFace is a database of face videos obtained from YouTube.

4.2 Compared Algorithms

The following state-of-the-art multi-view clustering methods are compared with our proposed algorithm in the experiment.

- **Multi-view k-means clustering on big data (RMKM) [3]**. This work is a robust large-scale multi-view clustering method that integrates heterogeneous representations of large-scale data.
- **Multi-view Subspace Clustering (MVSC) [8]**. In this paper, an efficient multi-view subspace clustering method is proposed and the effectiveness of the algorithm is verified.
- **Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification (AMGL) [23]**. This work proposes a framework that automatically learns the optimal weights for each graph and obtains globally optimal results.
- **Multi-view low-rank sparse subspace clustering (MLR SSC) [2]**. This work learns subspace representations by constructing affinity matrices shared among all views and solves the associated low-rank and sparse constrained optimization problems.
- **Flexible Multi-View Representation Learning for Subspace Clustering (FMR) [14]**. This work flexibly encodes complementary information from different views, thus avoiding the use of partial information for data reconstruction.
- **Partition level multiview subspace clustering (PMSC) [10]**. This work proposes a unified multi-view subspace clustering model and verifies the effectiveness of the algorithm.
- **Multi-view clustering in latent embedding space (MLE S) [5]**. The algorithm can simultaneously learn the global structure and the clustering indicator matrix and then cluster multi-view data in the potential embedding space.

Table 3: Clustering performance of compared methods under datasets with less than 10,000 samples. '–' means out-of-memory failure. The best results are colored in red and the blue indicates the second-best or no statistical difference.

Dataset	Metric	RMKM	MVSC	AMGL	MLRSSC	FMR	PMSC	MLES	LMVSC	Ours
Caltech101-20	ACC	0.3345	0.5080	0.1876	0.3600	0.3873	0.5981	0.3495	0.4304	0.6132
	NMI	0.0000	0.5271	0.1101	0.2008	0.5276	0.5244	0.3158	0.5553	0.5873
	Purity	0.3345	0.7125	0.6313	0.4476	0.7163	0.6480	0.5268	0.7125	0.6999
	Fscore	0.2799	0.4329	0.4661	0.3069	0.3521	0.5474	0.2972	0.3414	0.6699
CCV	ACC	0.1044	–	0.1102	0.1259	0.1671	–	–	0.2014	0.2182
	NMI	0.0000	–	0.0758	0.0471	0.1326	–	–	0.1657	0.1684
	purity	0.1044	–	0.2021	0.1307	0.2110	–	–	0.2396	0.2439
	Fscore	0.1084	–	0.1215	0.1095	0.1018	–	–	0.1194	0.1307
Caltech101-all	ACC	0.0875	–	0.0359	0.1365	–	–	–	0.2005	0.2750
	NMI	0.0000	–	0.0187	0.1066	–	–	–	0.4155	0.3510
	Purity	0.0875	–	0.4311	0.1371	–	–	–	0.3975	0.3395
	Fscore	0.0548	–	0.3617	0.0815	–	–	–	0.1586	0.2224
SUNRGBD	ACC	0.1836	–	0.0643	0.1741	–	–	–	0.1858	0.1930
	NMI	0.2612	–	0.0371	0.1108	–	–	–	0.2607	0.2007
	purity	0.3771	–	0.2411	0.1741	–	–	–	0.3818	0.2971
	Fscore	0.1168	–	0.1894	0.1453	–	–	–	0.1201	0.1279

- **Large-scale multi-view subspace clustering in linear time (LMVSC) [11].** The algorithm is designed to handle large-scale data and has linear complexity.

4.3 Experimental Setup

In our experimental setup, the initialization of $\mathbf{W}$, $\mathbf{A}$, and $\mathbf{Z}$ are set to zero matrices. Following the principle that the number of points required for the underlying subspace should not be less than the number of subspaces, we chose the number of anchors m in the range of $\{k, 2k, 3k\}$ and the common dimension $d = k$. For a fair comparison, we download the released code of comparison algorithms from their original websites. Since all methods need to utilize k-means to get the final clustering results, we run 50 times k-means to eliminate the randomness in k-means initialization for all compared methods. Then the clustering performance is evaluated by the widely used metrics accuracy (ACC), normalized mutual information (NMI), purity, and Fscore. By the way, the experimental environment is implemented on a desktop computer with an Intel Core i7-7820X CPU and 64GB RAM, MATLAB 2020b (64-bit).

4.4 Clustering Performance

We compare our proposed algorithm SMVSC with eight multi-view subspace clustering algorithms on seven widely used multi-view benchmark datasets. Table 3 and Table 4 show the detailed clustering performance results, and we mark the best results as red and the second results as blue in these tables.

4.4.1 Clustering performance on Datasets with sample size 10,000 and below. As shown in Table 3, we choose Caltech101-20, CCV, Caltech101-all and SUNGRBD datasets with different sample sizes.

Table 4: Clustering performance of compared methods under datasets with more than 30,000 samples. Other competitors are all out of memory. '–' means out-of-memory failure. The best results are colored in red and the blue indicates the second-best or no statistical difference.

Dataset	Metric	RMKM	LMVSC	Ours
NUSWIDEOBJ	ACC	0.1193	0.1583	0.1916
	NMI	0.0926	0.1337	0.1272
	Purity	0.2062	0.2488	0.2331
	Fscore	0.0750	0.0990	0.1365
AwA	ACC	0.0656	0.0770	0.0878
	NMI	0.0738	0.0879	0.1061
	Purity	0.0849	0.0957	0.0993
	Fscore	0.0359	0.0378	0.0636
YouTubeFace	ACC	–	0.1479	0.2587
	NMI	–	0.1327	0.2292
	Purity	–	0.2816	0.3321
	Fscore	–	0.0849	0.1287

In terms of ACC, our algorithm outperforms other state-of-the-art multi-view clustering algorithms. SMVSC exceeds the large-scale multi-view subspace clustering (LMVSC), which also has linear time complexity by 18%, 2%, 7%, and 1%, respectively, on four datasets. RMKM is a multi-view k-means clustering algorithm for

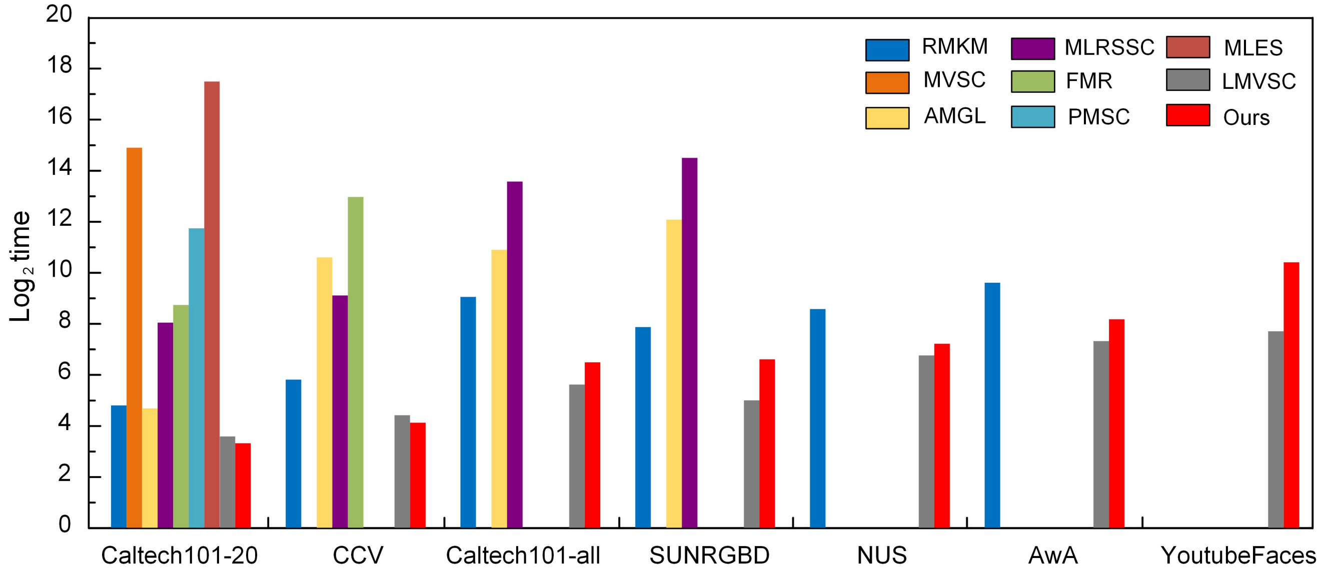
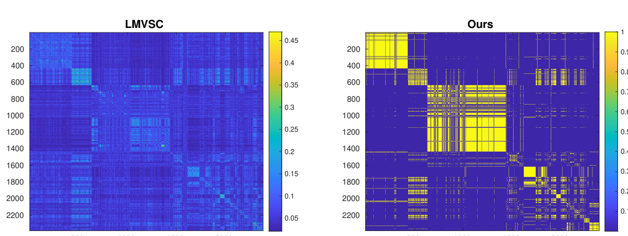


Figure 3: The running time of compared methods and ours over seven datasets. For clarity, the y axis is scaled by $\log$ to alleviate the gap between some methods and ours. Missing bars indicate that the method encounters an out-of-memory error on our experiment platform under this dataset.

handling large-scale clustering problems, and SMVSC is higher than RMKM by 27%, 11%, 19%, and 1%, respectively. In NMI, purity, and Fscore, SMVSC can achieve near parity with or even show better performance than other algorithms. In addition, there are no relevant results in Table 3 due to out-of-memory for some algorithms. For example, these two latest algorithms, MLES and PMSC, spend time in the tens of thousands per set of parameters on a dataset with a sample size of 2,000 from Fig. 3 and suffer from a out-of-memory error afterward. Therefore, we did not continue experiments with these algorithms.

Furthermore, we plot the complete graph to illustrate our better-learned clustering structure compared with LMVSC in Fig. 4.



(a) Complete graph of LMVSC **(b) Complete graph of ours**

Figure 4: Graph structure comparison between LMVSC and ours on Caltech101-20. Brighter color means a larger value. Compared with the graph of LMVSC, our graph shows clearer clustering structure with less noise.

Since the anchor graph of each view in LMVSC is learned independently, we concatenate them to form a $vm \times n$ anchor graph $\hat{\mathbf{S}}$ and then construct a complete graph by $\hat{\mathbf{S}}^\top\hat{\mathbf{S}}$ Our complete graph can be directly obtained by the unified anchor graph $\mathbf{Z}^\top\mathbf{Z}$. As can be seen in Fig. 4, our graph shows more clear block structures, while the one of LMVSC seems to be noisier and less clear.

4.4.2 Clustering performance on Datasets with sample sizes of more than 30,000. For better application to large-scale scenarios, we chose datasets NUSWIDEOBJ, AwA and YouTubeFace with sample sizes of 30,000 or more.

During the experiments, all of the multi-view subspace clustering algorithms, except for the algorithms that solve large-scale data, suffer directly from the "out-of-memory" problem. Therefore, there is no clustering performance for these algorithms in Table 4 for the above datasets.

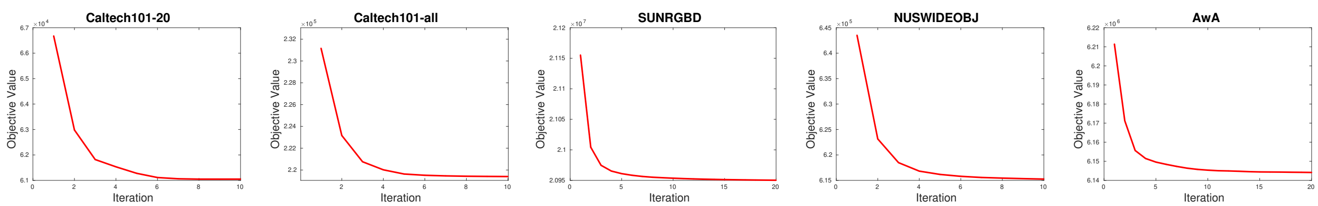
Based on the experimental results in Table 4, our proposed SMVSC still maintains excellent clustering performance on these larger datasets. On the 100,000 sample datasets YouTubeFace, SMKSC keeps running and performs better than LMVSC by 11%, 10%, 5%, and 4% on ACC, NMI, Purity and Fscore, respectively. These demonstrate that our algorithm has a lower spatial complexity when dealing with large-scale data and outperforms similar algorithms in terms of stability and accuracy.

4.5 Running Time

For a fair comparison, we uniformly set all algorithms to perform k-means 50 times and report the running time of the optimal set of parameters. Fig. 3 shows the distribution of running times over all datasets., Some of the algorithms do not have experimental results on some datasets and therefore, the corresponding histograms are not available in this figure. We can visually see that SMVSC has a

Table 5: Running time comparison of compared methods under datasets with less than 10,000 samples.

Method	Caltech101-20		CCV		Caltech101-all		SUNRGBD	
	Time(s)	Speed Up	Time(s)	Speed Up	Time(s)	Speed Up	Time(s)	Speed Up
RMKM	27.74	6,590.60 ×	55.84	142.21 ×	526.86	22.91 ×	232.39	99.04 ×
MVSC	30,259.00	6.04 ×	–	–	–	–	–	–
AMGL	25.60	7,139.90 ×	1,544.30	5.14 ×	1,897.20	6.36 ×	4,293.30	5.36 ×
MLRSSC	261.99	697.77 ×	549.44	14.45 ×	12,068.00	1.00 ×	23,015.00	1.00 ×
FMR	423.21	431.96 ×	7,940.90	1.00 ×	–	–	–	–
PMSC	3,399.70	53.77 ×	–	–	–	–	–	–
MLES	182,810.00	1.00 ×	–	–	–	–	–	–
LMVSC	11.95	15,294.07 ×	21.20	374.50 ×	48.85	247.04 ×	31.82	723.26 ×
Ours	9.91	18,451.31 ×	17.29	459.28 ×	89.24	135.24 ×	96.96	237.38 ×



(a) Caltech101-20 **(b) Caltech101-all** **(c) SUNRGBD** **(d) NUSWIDEOBJ** **(e) AwA**

Figure 5: The objective of our proposed method on five benchmark datasets

very advantageous running time and more details of the running time, as well as speed up, are recorded in Table 5 and Table 6.

As can be seen in Table 5 and Table 6, the speed up of SMVSC is significantly superior to the other algorithms and has almost twice of some algorithms, As far as AMGL and MLRSSC, although they work fine for datasets with 10,000 samples and below, they consume too much time cost. More importantly, the clustering performance of these two algorithms is not satisfactory. On larger datasets, the clustering performance of SMVSC and the large-scale oriented algorithms is more impressive, although both have linear complexity. Although LMVSC is the fastest among most of the datasets, it is easy to obtain that the simple k-means sampling strategy and the equal weight combination do not utilize complementary information with view interaction

These prove that the linear time complexity of SMVSC is more readily scalable to large-scale data, while some multi-view subspace clustering algorithms take a long time to process large-scale data.

Table 6: Running time of three large-scale methods under datasets with more than 30,000 samples.

Method	NUSWIDEOBJ		AwA		YouTubeFace	
	Time	Speed Up	Time	Speed Up	Time	Speed Up
RMKM	379.32	1.00 ×	774.64	1.00 ×	–	–
LMVSC	107.54	3.57 ×	158.57	4.89 ×	207.92	6.50 ×
Ours	147.75	2.57 ×	286.77	2.70 ×	1351.50	1.00 ×

4.6 Convergence

As mentioned in the method part, our algorithm can be theoretically guaranteed to converge to a local optimum. In this subsection, we record the objective values on each dataset to show our experimental convergence. Due to the limitation of the space, we only plot the evolution of the objective values on five datasets Caltech101-20, Caltech101-all, SUNRGBD, NUSWIDEOBJ and AwA. As shown in the Fig. 5, the objective values monotonically decrease at each iteration and usually converge in less than twenty iterations and most datasets can converge in less than ten iterations. These results verify our proposed algorithm's convergence experimentally.

5 CONCLUSION

In this paper, we propose a scalable multi-view subspace clustering with unified anchors to solve the clustering problem for large-scale data. The algorithm adaptively learns the weights of each view and combines anchor learning and graph construction into a unified optimization framework. This enables the learned anchors to more accurately represent the actual underlying data distribution and obtain a more discriminative clustering structure. Due to the linear time complexity of the SMVSC, the fast running time makes SMVSC better suited to realistic large-scale application scenarios. Compared with state-of-the-art multi-view subspace clustering methods and large-scale oriented methods, extensive experiments verify that the SMVSC maintains comparable or better clustering performance while having linear time complexity.

REFERENCES

[1] Séverine Affeldt, Lazhar Labiod, and Mohamed Nadif. 2020. Spectral Clustering via Ensemble Deep Autoencoder Learning (SC-EDAE). *Pattern Recognition* 108 (Dec. 2020), 107522.

[2] Maria Brbić and Ivica Kopriva. 2018. Multi-view low-rank sparse subspace clustering. *Pattern Recognition* 73 (2018), 247–258.

[3] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-view k-means clustering on big data. In *Twenty-Third International Joint conference on artificial intelligence*. Citeseer.

[4] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. 2015. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–594.

[5] Man-Sheng Chen, Ling Huang, Chang-Dong Wang, and Dong Huang. 2020. Multi-view clustering in latent embedding space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3513–3520.

[6] Xinlei Chen and Deng Cai. 2011. Large Scale Spectral Clustering with Landmark-Based Representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 25.

[7] Ehsan Elhamifar and René Vidal. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence* 35, 11 (2013), 2765–2781.

[8] Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. 2015. Multi-view subspace clustering. In *Proceedings of the IEEE international conference on computer vision*. 4238–4246.

[9] Quanxue Gao, Wei Xia, Zhizhen Wan, Deyan Xie, and Pu Zhang. 2020. Tensor-SVD based graph learning for multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3930–3937.

[10] Zhao Kang, Xinjia Zhao, Chong Peng, Hongyuan Zhu, Joey Tianyi Zhou, Xi Peng, Wenyu Chen, and Zenglin Xu. 2020. Partition level multiview subspace clustering. *Neural Networks* 122 (2020), 279–288.

[11] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. 2020. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4412–4419.

[12] Jinxing Li, Hongwei Yong, Feng Wu, and Mu Li. 2020. Online Multi-view Subspace Learning with Mixed Noise. In *Proceedings of the 28th ACM International Conference on Multimedia*. 3838–3846.

[13] Jianshu Li, Jian Zhao, Fang Zhao, Hao Liu, Jing Li, Shengmei Shen, Jiashi Feng, and Terence Sim. 2016. Robust face recognition with deep multi-view representation learning. In *Proceedings of the 24th ACM international conference on Multimedia*. 1068–1072.

[14] Ruihuang Li, Changqing Zhang, Qinghua Hu, Pengfei Zhu, and Zheng Wang. 2019. Flexible Multi-View Representation Learning for Subspace Clustering.. In *IJCAI*. 2916–2922.

[15] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[16] Weixuan Liang, Sihang Zhou, Jian Xiong, Xinwang Liu, Siwei Wang, En Zhu, Zhiping Cai, and Xin Xu. 2020. Multi-View Spectral Clustering with High-Order Optimal Neighborhood Laplacian Matrix. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[17] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. 2012. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2012), 171–184.

[18] Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, and Jianping Yin. 2017. Optimal neighborhood kernel clustering with multiple kernels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[19] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, Chang Tang, Jianping Yin, Dinggang Shen, Huaimin Wang, and Wen Gao. 2018. Late fusion incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence* 41, 10 (2018), 2410–2423.

[20] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. 2018. Consistent and specific multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[21] Zhengrui Ma, Zhao Kang, Guangchun Luo, Ling Tian, and Wenyu Chen. 2020. Towards Clustering-friendly Representations: Subspace Clustering via Graph Filtering. In *Proceedings of the 28th ACM International Conference on Multimedia*. 3081–3089.

[22] Feiping Nie, Guohao Cai, and Xuelong Li. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[23] Feiping Nie, Jing Li, Xuelong Li, et al. 2016. Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification.. In *IJCAI*. 1881–1887.

[24] Qiyuan Ou, Siwei Wang, Sihang Zhou, Miaomiao Li, Xifeng Guo, and En Zhu. 2020. Anchor-Based Multiview Subspace Clustering With Diversity Regularization. *IEEE MultiMedia* 27, 4 (2020), 91–101.

[25] Hua Wang, Feiping Nie, and Heng Huang. 2013. Multi-view clustering and feature learning via structured sparsity. In *International conference on machine learning*. PMLR, 352–360.

[26] Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. 2019. Multi-view Clustering via Late Fusion Alignment Maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 3778–3784.

[27] Xiaobo Wang, Zhen Lei, Xiaojie Guo, Changqing Zhang, Hailin Shi, and Stan Z Li. 2019. Multi-view subspace clustering with intactness-aware similarity. *Pattern Recognition* 88 (2019), 50–63.

[28] Yang Wang, Xuemin Lin, Lin Wu, Wenjie Zhang, and Qing Zhang. 2014. Exploiting correlation consensus: Towards subspace clustering for multi-modal data. In *Proceedings of the 22nd ACM international conference on Multimedia*. 981–984.

[29] Yang Wang and Lin Wu. 2018. Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. *Neural Networks* 103 (2018), 1–8.

[30] Bin Wu, Erheng Zhong, Andrew Horner, and Qiang Yang. 2014. Music emotion recognition by multi-label multi-layer multi-instance multi-view learning. In *Proceedings of the 22nd ACM international conference on Multimedia*. 117–126.

[31] Yu-Meng Xu, Chang-Dong Wang, and Jian-Huang Lai. 2016. Weighted multi-view clustering with feature selection. *Pattern Recognition* 53 (2016), 25–35.

[32] Fei Yan, Xiao-dong Wang, Zhi-qiang Zeng, and Chao-qun Hong. 2020. Adaptive multi-view subspace clustering for high-dimensional data. *Pattern Recognition Letters* 130 (2020), 299–305.

[33] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. 2018. Generalized latent multi-view subspace clustering. *IEEE transactions on pattern analysis and machine intelligence* 42, 1 (2018), 86–99.

[34] Changqing Zhang, Huazhu Fu, Si Liu, Guangcan Liu, and Xiaochun Cao. 2015. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE international conference on computer vision*. 1582–1590.

[35] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. 2017. Latent multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4279–4287.

[36] P. Zhang, X. Liu, J. Xiong, S. Zhou, W. Zhao, E. Zhu, and Z. Cai. 2020. Consensus One-step Multi-view Subspace Clustering. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1.

[37] Qinghai Zheng, Jihua Zhu, Zhongyu Li, Shanmin Pang, Jun Wang, and Yaochen Li. 2020. Feature concatenation multi-view subspace clustering. *Neurocomputing* 379 (2020), 89–102.

[38] Chengju Zhou, Changqing Zhang, Huazhu Fu, Rui Wang, and Xiaochun Cao. 2015. Multi-cue augmented face clustering. In *Proceedings of the 23rd ACM international conference on Multimedia*. 1095–1098.

[39] Sihang Zhou, Xinwang Liu, Miaomiao Li, En Zhu, Li Liu, Changwang Zhang, and Jianping Yin. 2019. Multiple kernel clustering with neighbor-kernel subspace segmentation. *IEEE transactions on neural networks and learning systems* 31, 4 (2019), 1351–1362.