Efficient One-Pass Multi-View Subspace Clustering with Consensus Anchors

Suyuan Liu,[1*] **Siwei Wang,**[1*] **Pei Zhang,**[1] **Kai Xu,**[1] **Xinwang Liu,**[1†] **Changwang Zhang,**[2] **Feng Gao**[3]

[1] School of Computer, National University of Defense Technology, Changsha, China, 410073
[2] CCF Theoretical Computer Science Technical Committee, Shenzhen, China, 518064
[3] School of Arts, Peking University, Beijing, China, 100871
suyuanliu@nudt.edu.cn, wangsiwei13@nudt.edu.cn, zhangpei@nudt.edu.cn, kevin.kai.xu@gmail.com,
xinwangliu@nudt.edu.cn, changwangzhang@foxmail.com, gaof@pku.edu.cn

Abstract

Multi-view subspace clustering (MVSC) optimally integrates multiple graph structure information to improve clustering performance. Recently, many anchor-based variants are proposed to reduce the computational complexity of MVSC. Though achieving considerable acceleration, we observe that most of them adopt fixed anchor points separating from the subsequential anchor graph construction, which may adversely affect the clustering performance. In addition, post-processing is required to generate discrete clustering labels with additional time consumption. To address these issues, we propose a scalable and parameter-free MVSC method to directly output the clustering labels with optimal anchor graph, termed as Efficient One-pass Multi-view Subspace Clustering with Consensus Anchors (EOMSC-CA). Specially, we combine anchor learning and graph construction into a uniform framework to boost clustering performance. Meanwhile, by imposing a graph connectivity constraint, our algorithm directly outputs the clustering labels without any post-processing procedures as previous methods do. Our proposed EOMSC-CA is proven to be linear complexity respecting to the data size. The superiority of our EOMSC-CA over the effectiveness and efficiency is demonstrated by extensive experiments. Our code is publicly available at https://github.com/Tracesource/EOMSC-CA.

Introduction

In the era of big data, high-dimensional datasets have much smaller inherent dimension than the respective ambient space. In fact, most of the data can be represented by samples drawn from the sets of subspaces with low-dimension. Based on the above assumptions, subspace clustering attends to cluster the data lied in the linear subspaces and determine the low-dimensional subspace corresponding to each cluster (Vidal 2011; Elhamifar and Vidal 2013). However, data usually are collected from different sources or diverse domains in many real-world problems (Wang et al. 2019; Zhang et al. 2020; Liu et al. 2021b; Wang et al. 2021a). For instance, the same news can be covered in multiple forms as text, image and video (Kang et al. 2019). Therefore, in order to take full advantage of information from multiple sources, many MVC(multi-view clustering) methods have been proposed recently to explore consistency and complementarity across views (Tang et al. 2018; Wang et al. 2020; Liu et al. 2021a; Zhang et al. 2021; Zhou et al. 2020a). The current MVC methods are mostly based on graph models, which first learn a common similarity matrix from the features of different views and then execute a spectral clustering algorithm to obtain the final clustering results (Liu et al. 2012; Ding and Fu 2014; Zhan et al. 2017; Xu et al. 2018; Zhou et al. 2021). Existing multi-view subspace clustering algorithms suffer from high time consumption, making it difficult to process large-scale data in reality. The high time complexity mainly comes from the construction of the similarity matrix, the calculation of the spectral embedding in spectral clustering, and the discretization of the spectral embedding.

In order to improve efficiency, it has recently been proposed to apply bipartite graph learning, using the relationship between anchor points and data points to represent the relationship between all data points (Chen and Cai 2011; Adler, Elad, and Hel-Or 2015; Li et al. 2020; Liu et al. 2021c). Then the graph size of each view is reduced from $n \times n$ to $n \times m$ and m is the number of anchor points. However, most anchor point selection applies heuristic sampling strategies, such as k-means or random sampling, which prevents the mutual negotiation between anchor point selection and graph construction to achieve optimal clustering. Besides, existing graph-based methods usually consist of two stages, which construct graphs from raw data first, and perform post-processing to obtain the final result then (Zhou et al. 2020b; Gao et al. 2020; Liu et al. 2019). The final cluster structure is not clearly shown in the graph constructed in the previous stage, and the clustering performance dependents heavily on the constructed graphs (Nie et al. 2016b, 2017). Moreover, multi-view clustering algorithms usually encounter a large number of hyper-parameters. How to determine these hyper-parameters which severely reduce the effectiveness of algorithms remains to be a challenge.

In this paper, we propose a scalable MVSC method named Efficient One-pass Multi-view Subspace Clustering with Consensus Anchors (EOMSC-CA) to directly output the clustering labels without post-processing procedure. Firstly, we learn the anchor points and fused graph together, and the two procedures can be negotiated mutually to improve clustering performance. At the same time, we adaptively learn

[*] First author with equal contribution
[†] Corresponding author

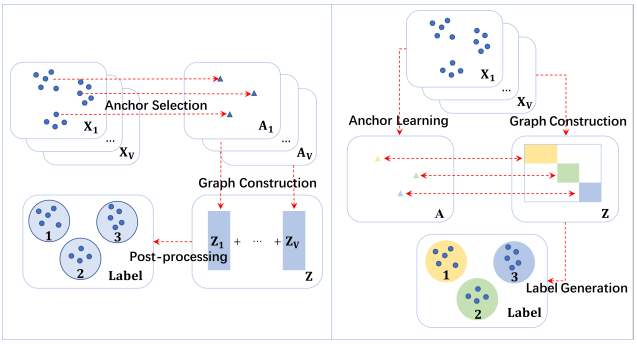


Figure 1: The framework of existing anchor-based methods(left) and our method(right). The existing methods first select anchor points from the data point of each view according to the sampling strategy, then construct the graph representation separately, and finally perform post-processing to obtain the final cluster label. In contrast, our method directly learns consistent anchor points and common graph representations from multi-view data. The process of learning anchors and the process of constructing graph influence each other, and finally clustering labels are directly generated through graphs with exactly k connected components.

view weights in the framework without additional hyper-parameters to balance the effects of the view. In addition, we impose rank constraints on the Laplacian graph to make sure that the final joint graph has exactly several connected components of clusters to indicate each cluster. The effectiveness and efficiency of our method can be verified by extensive experiments on various large-scale benchmark datasets.

We can summarize our contribution in the following points:

- Instead of existing fixed sampling anchors, we unify fused graph construction and anchor learning into a unified and flexible framework so that they seamlessly contribute mutually and boost performance.
- We propose a scalable MVSC method to output the optimal anchor graph with exactly k-connected components, which can be directly used for clustering without performing discretization procedures necessary for other graph-based clustering methods.
- The result of the experiment on multiple datasets validate the effectiveness and efficiency of our algorithm. Our method is proven to be linearly time complexity respecting to sample numbers without any hyper-parameters. These clearly make the proposed EOMSC-CA more suitable for large-scale tasks.

Background

In this section, we introduce the subspace clustering in single view and multi-view first, and then introduce the anchor-based MVC methods. Table 1 summarizes our mainly used notations.

Notation	Definition
n	Amount of data points
k	Amount of clusters
v	Amount of views
m	Amount of anchor points
l	Dimension of anchor matrix
d_p	Dimension of the p-th view
d	$\sum_{p=1}^{v} d_p$
$\boldsymbol{\beta}$	View coefficient
$\mathbf{X}_p \in \mathbb{R}^{d_p \times n}$	Data matrix
$\mathbf{W}_p \in \mathbb{R}^{d_p \times l}$	Projection matrix
$\mathbf{A} \in \mathbb{R}^{l \times m}$	Consensus anchor matrix
$\mathbf{Z} \in \mathbb{R}^{m \times n}$	Fused anchor graph

Table 1: Mainly used notations.

Subspace Clustering

Given data $\mathbf{X} \in \mathbb{R}^{d \times n}$, subspace-based methods assume that a linear combination of other data points in the same subspace can express each data point.

The first step of subspace clustering is to construct a graph:

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

where $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes the non-negative self-representation matrix, $f(\cdot)$ represents the regularization functions, and λ is a balance parameter. The $\mathbf{S}^\top \mathbf{1} = \mathbf{1}$ ensures that each column of $\mathbf{S}$ adds up to 1. With the size of $\mathbf{S}$, the graph construction in the first stage often takes at least $\mathcal{O}\left(n^3\right)$.

After obtaining the coefficient matrix $\mathbf{S}$, the second step is to get the spectral embedding $\mathbf{F} \in \mathbb{R}^{n \times k}$ with the symmetric similarity matrix $\mathbf{W}$ constructed as $\mathbf{W} = \frac{\mathbf{S}+\mathbf{S}^\top}{2}$,

$$\min_{\mathbf{F}} \operatorname{Tr}\left(\mathbf{F}^\top \mathbf{L}\mathbf{F}\right), s.t. \mathbf{F}^\top \mathbf{F} = \mathbf{I}_k, \quad (2)$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian with diagonal matrix $\mathbf{D}$ defined as $d_{ii} = \sum_{j=1}^{n} s_{ij}$, and F is the spectral embedding. The spectral clustering stage takes $\mathcal{O}\left(n^3\right)$ complexity.

Multi-view Subspace Clustering

For multi-view data $\{\mathbf{X}_p\}_{p=1}^{v}$, where $\mathbf{X}_p \in \mathbb{R}^{d_p \times n}$ represents the p-th view data with d_p dimensions, the MVSC model can be expressed as follows:

$$\min_{\mathbf{S}_p, \mathbf{S}} \|\mathbf{X}_p - \mathbf{X}_p \mathbf{S}_p\|_F^2 + \lambda f(\mathbf{S}, \mathbf{S}_p), s.t. \mathbf{S}_p \geq 0, \mathbf{S}_p^\top \mathbf{1} = \mathbf{1}, \quad (3)$$

where $f(\cdot)$ represents the unified regularization term. Performing spectral clustering on the fused global graph $\mathbf{Z}$ and then the final results can be reached.

Based on the above framework, many MVSC methods have recently been proposed. Gao et al. (2015) propose to learn an independent subspace representation in each view,

and then use a unified index matrix to get a common clustering result. In order to explore the complementarity information, Cao et al. (2015) propose to introduce Hilbert-Schmidt independence criterion as a regularization term. Considering that the subspace representation learned from multiple views may be redundant, Zhang et al. (2015) regard the subspace representation matrix of different views as a tensor, and explore the intersection information among views with low rank constraints. Another important point is to impose appropriate constraints on the subspace representation, such as low-rank constraints (Wang et al. 2018) or sparse constraints (Lu, Yan, and Lin 2016). Brbić and Kopriva (2018) suggest to build a consensus affinity matrix that satisfies both low-rank and sparsity.

However, the above methods poorly handle with scalability problem. Specifically, in addition to fusing multi-view information, multi-view subspace clustering requires graph construction and spectral embedding as the same as single-view.

Anchor-based Multi-view Clustering

In recent years, anchor graph is applied to solve large-scale data clustering problems (Kang et al. 2020; Li et al. 2020; Ou et al. 2020; Kang et al. 2021; Sun et al. 2021; Wang et al. 2021b). The principle of anchor based method is to choose some representative points from the original data called anchor points, and then use the relationship between anchor points and the entire data points to cover the complete affinity. Specifically, a small graph $\mathbf{Z} \in \mathbb{R}^{m\times n}$ is constructed to replace the original graph $\mathbf{S} \in \mathbb{R}^{n\times n}$. With the application of anchor points, the complexity of subspace clustering can be greatly reduced while maintaining considerable efficiency.

Some recent works have applied anchor graphs into multi-view clustering. Kang et al. (2020) propose to get anchors of each view with k-means at first, and then construct anchor graphs separately. Li et al. (2020) provide an anchor selection strategy based to importance build anchor graphs independently in each view and then fuse them into a consensus one.

Although anchor-based methods achieve considerable performance, there exists several limitations. For one thing, the selection of anchor points is separated in each view without mutual negotiation with the clustering process. For another, inevitable hyper-parameters affect their efficiency on large-scale data. Moreover, existing works need to preform a post-processing process to get the final result, and the cluster structure is not obvious in the constructed graph. In the next section, we propose EOMSC-CA to solve the above problems.

The Proposed Methodology

In this section, we will describe in detail the novel subspace clustering method termed as EOMSC-CA, including the motivation, formulation, optimization, and complexity analysis.

Motivation

Reducing the redundancy of data in large-scale data clustering is the key to improve efficiency. Actually, a small number of instances are enough to reconstruct the underlying subspaces. Therefore, the existing works propose to select some points from the original data as anchor points to reconstruct the relationship structure.

However, the existing anchor-based multi-view subspace methods all perform heuristic sampling strategies, which means that graph construction and anchor selection are separated. After selecting the anchor points separately in each view, graphs are then constructed as follows,

$$\min_{\mathbf{Z}_p} \sum_{p=1}^{v} \|\mathbf{X}_p - \mathbf{A}_p\mathbf{Z}_p\|_F^2, s.t. \mathbf{Z}_p \geq 0, \mathbf{Z}_p^\top \mathbf{1} = \mathbf{1}, \quad (4)$$

where $\mathbf{A}_p$ denotes anchor matrix on each view and $\mathbf{Z}_p$ is the anchor graph.

The graph constructed separately in each view cannot explore the complementary information well, which need to perform fusion algorithm to get a consensus graph. Besides, after constructing the graph with subspace clustering, it is usually necessary to perform spectral clustering to obtain the spectral embedding, and then adopts k-means to obtain the clustering results. This two-stage process causes the final cluster structure unclear in the constructed graph, and quality of the graph significantly affects the clustering results (Nie et al. 2016b).

Formulation of Problem

For the above challenges, we intend to **learn** anchors through optimization instead of sampling. Based on the assumption that high-dimensional data in different views share a consensus low-dimensional subspace, the learned anchors should be consistent in the consensus subspace. Define the projection matrix $\{\mathbf{W}_p\}_{p=1}^v$, we can align the consensus anchors $\mathbf{A}$ with the original data of the p-th view.

In order to make the obtained graph a clear clustering structure and get the cluster results directly, we hope to construct a graph containing precisely k connected parts. For the graph $\mathbf{Z} \in \mathbb{R}^{m\times n}$ constructed based on the anchor method, we define the augmented graph $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} & \mathbf{Z}^\top \\ \mathbf{Z} & \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}. \quad (5)$$

Then we get the normalized Laplacian matrix $\widetilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$, where $d_i = \sum_{j=1}^{n+m} s_{ij}$ is the i-th element of the diagonal matrix $\mathbf{D}$. According to the following Theorem, the rank constraint of Laplacian matrix ensures $\mathbf{S}$ to be k-connected.

Theorem 1 : The number of connected components in $\mathbf{S}$ equals to the cardinality of the eigenvalue zero of the normalized Laplacian matrix $\widetilde{\mathbf{L}}$.

Therefore, the graph $\mathbf{S}$ has precisely k connected components with rank constraints on $\widetilde{\mathbf{L}}$. Since $\mathbf{S}$ is composed of $\mathbf{Z}$, $\mathbf{Z}$ has the same amount of connected components as $\mathbf{S}$.

Eventually, our proposed EOMSC-CA has the following formulation,

$$\min_{\boldsymbol{\beta},\mathbf{W}_p,\mathbf{A},\mathbf{Z}} \sum_{p=1}^{v} \beta_p^2 \|\mathbf{X}_p - \mathbf{W}_p\mathbf{A}\mathbf{Z}\|_F^2, \quad (6)$$

$$s.t. \boldsymbol{\beta}^\top \mathbf{1} = 1, \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_l, \mathbf{A}^\top \mathbf{A} = \mathbf{I}_m,$$
$$\mathbf{Z} \geq 0, \mathbf{Z}^\top \mathbf{1} = \mathbf{1}, rank\left(\widetilde{\mathbf{L}}\right) = n + m - k,$$

where $\mathbf{X}_p \in \mathbb{R}^{d_p \times n}$ denotes the original data of the p-th view with d_p dimension, β_p is the weight coefficient that balances the influence of each view, $\mathbf{W}_p$ is the projection matrix and $\mathbf{A} \in \mathbb{R}^{l \times m}$ is the consensus anchor matrix with m anchors and l dimension.

Optimization

To solve the above optimization, all variables are updated alternatively, which means updating one variable with others being fixed.

Optimize Z by fixing other variables When $\mathbf{W}_p, \mathbf{A}, \boldsymbol{\beta}$ are fixed, optimizing $\mathbf{Z}$ is equal to solve the following equation:

$$\min_{\mathbf{Z}} \sum_{p=1}^{v} \beta_p^2 \operatorname{Tr}\left(\mathbf{Z}^\top \mathbf{Z} - 2\mathbf{X}_p^\top \mathbf{W}_p \mathbf{A} \mathbf{Z}\right), \tag{7}$$
$$s.t. \mathbf{Z} \geq 0, \mathbf{Z}^\top \mathbf{1} = \mathbf{1}, \operatorname{rank}\left(\widetilde{\mathbf{L}}\right) = n + m - k.$$

Define $\sigma_i\left(\widetilde{\mathbf{L}}\right)$ as the i-th smallest eigenvalue of $\widetilde{\mathbf{L}}$. Then $\sigma_i\left(\widetilde{\mathbf{L}}\right) \geq 0$ since $\widetilde{\mathbf{L}}$ is semi-definite. With λ to be large enough, Eq. (7) equals to the following problem:

$$\min_{\mathbf{Z}} \sum_{p=1}^{v} \beta_p^2 \operatorname{Tr}\left(\mathbf{Z}^\top \mathbf{Z} - 2\mathbf{X}_p^\top \mathbf{W}_p \mathbf{A} \mathbf{Z}\right) + \lambda \sum_{i=1}^{k} \sigma_i\left(\widetilde{\mathbf{L}}\right), \tag{8}$$
$$s.t. \mathbf{Z} \geq 0, \mathbf{Z}^\top \mathbf{1} = \mathbf{1}.$$

We can rewrite Eq. (8) as follows according Fan (1949),

$$\min_{\mathbf{Z},\mathbf{F}} \sum_{p=1}^{v} \beta_p^2 \operatorname{Tr}\left(\mathbf{Z}^\top \mathbf{Z} - 2\mathbf{X}_p^\top \mathbf{W}_p \mathbf{A} \mathbf{Z}\right) + \lambda \operatorname{Tr}\left(\mathbf{F}^\top \widetilde{\mathbf{L}} \mathbf{F}\right), \tag{9}$$
$$s.t. \mathbf{Z} \geq 0, \mathbf{Z}^\top \mathbf{1} = \mathbf{1}, \mathbf{F}^\top \mathbf{F} = \mathbf{I}_k,$$

where $\mathbf{F} \in \mathbb{R}^{(n+m) \times k}$ is the indicator matrix.

Fixing $\mathbf{Z}$, Eq. (9) becomes:

$$\min_{\mathbf{F}^\top \mathbf{F} = \mathbf{I}_k} \operatorname{Tr}\left(\mathbf{F}^\top \widetilde{\mathbf{L}} \mathbf{F}\right). \tag{10}$$

To solve Eq. (10), we need to compute the eigenvectors of $\widetilde{\mathbf{L}}$, which takes $\mathcal{O}(k(n+m)^2)$ complexity. In order to reduce the complexity, we compute the eigenvectors of $\mathbf{Z}$ instead of $\mathbf{S}$. Specifically, $\mathbf{F}$ and $\mathbf{D}$ in Eq. (10) can be decomposed as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{(n)} \\ \mathbf{F}_{(m)} \end{bmatrix}, \mathbf{D}_S = \begin{bmatrix} \mathbf{D}_{(n)} & \\ & \mathbf{D}_{(m)} \end{bmatrix}, \tag{11}$$

where $\mathbf{F}_{(n)}$ represents the indicator of data points and $\mathbf{F}_{(m)}$ denotes the indicator of anchor points. Then we can rewrite Eq. (10) as follows:

$$\min_{\mathbf{F}_{(n)}^\top \mathbf{F}_{(n)} + \mathbf{F}_{(m)}^\top \mathbf{F}_{(m)} = \mathbf{I}_k} \operatorname{Tr}\left(\mathbf{F}_{(n)}^\top \mathbf{D}_{(n)}^{\frac{1}{2}} \mathbf{Z}^\top \mathbf{D}_{(m)}^{\frac{1}{2}} \mathbf{F}_{(m)}^\top\right). \tag{12}$$

Eq. (12) can be easily solved by the following theorem.

Theorem 2 : Suppose $\mathbf{Q} \in \mathbb{R}^{n \times m}$, $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{B} \in \mathbb{R}^{m \times k}$, we have the follow problem

$$\max_{\mathbf{A}^T \mathbf{A} + \mathbf{B}^T \mathbf{B} = \mathbf{I}_k} \operatorname{Tr}\left(\mathbf{A}^T \mathbf{Q} \mathbf{B}\right). \tag{13}$$

The optimal solutions of the problem are $\mathbf{A} = \frac{\sqrt{2}}{2}\mathbf{U}$ and $\mathbf{B} = \frac{\sqrt{2}}{2}\mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are respectively corresponding to the top k left and right singular vectors.

After optimizing $\mathbf{F}$, we optimize $\mathbf{Z}$ in Eq. (9) with the optimal $\mathbf{F}$. To solve the problem, we noticed:

$$\operatorname{Tr}\left(\mathbf{F}^\top \widetilde{\mathbf{L}} \mathbf{F}\right) = \frac{1}{2} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} \left\| \frac{\mathbf{f}^i}{\sqrt{\mathbf{D}(i,i)}} - \frac{\mathbf{f}^j}{\sqrt{\mathbf{D}(j,j)}} \right\|_2^2 s_{ij}. \tag{14}$$

With the definition in Eq. (11), we can further transform the above formula as follows,

$$\operatorname{Tr}\left(\mathbf{F}^\top \widetilde{\mathbf{L}} \mathbf{F}\right) = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} \left\| \frac{\mathbf{f}_{(n)}^i}{\sqrt{\mathbf{D}_{(n)}(i,i)}} - \frac{\mathbf{f}_{(m)}^j}{\sqrt{\mathbf{D}_{(m)}(j,j)}} \right\|_2^2 z_{ji}. \tag{15}$$

Then Eq. (9) with respect to $\mathbf{Z}$ is equal to:

$$\min_{\mathbf{Z}^\top \mathbf{1} = \mathbf{1}, \mathbf{Z} \geq \mathbf{0}} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{p=1}^{v} \beta_p^2 \left(z_{ji}^2 - 2c_{pij} z_{ji}\right) + \lambda t_{ij} z_{ji}, \tag{16}$$

where $\mathbf{C}_p = \mathbf{X}_p^\top \mathbf{W}_p \mathbf{A}$ and $t_{ij} = \left\| \frac{\mathbf{f}_{(n)}^i}{\sqrt{\mathbf{D}_{(n)}(i,i)}} - \frac{\mathbf{f}_{(m)}^j}{\sqrt{\mathbf{D}_{(m)}(j,j)}} \right\|_2^2$. Denoting $\mathbf{Z}_{:,i}$ as a vector with the i-th element to be $\mathbf{z}_{ji}$, we optimize $\mathbf{Z}$ by column:

$$\min_{\mathbf{Z}_{:,i}^\top \mathbf{1} = 1, \mathbf{Z}_{:,i} \geq \mathbf{0}} \left\| \mathbf{Z}_{:,i}^\top - \left(\mathbf{C}_{i,:} - \frac{\lambda}{2} \mathbf{t}_{i,:}\right) / \sum_{p=1}^{v} \beta_p^2 \right\|_2^2, \tag{17}$$

where $\mathbf{C} = \sum_{p=1}^{v} \beta_p^2 \mathbf{C}_p$.

According to Nie, Wang, and Huang (2014), Eq. (17) has a closed form solution. We summarize the procedures of solving Eq. (7) in Algorithm 1.

Algorithm 1: Algorithm for optimizing $\mathbf{Z}$.

Input: $\mathbf{C} \in \mathbb{R}^{n \times m}$, cluster number k, a small value λ.
Output: Anchor graph $\mathbf{Z} \in \mathbb{R}^{m \times n}$.

1: Initialize $\mathbf{F}_{(n)}$ and $\mathbf{F}_{(m)}$, which are formed by the k left and right singular vectors of $\begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix} \in \mathbb{R}^{n \times m}$ corresponding to the k largest singular values.
2: **repeat**
3: Update each column of $\mathbf{Z}$ by solving Eq. (17);
4: Update $\mathbf{D}_{(n)}$ and $\mathbf{D}_{(m)}$ where $d_{(n)}(i,i) = \sum_{j=1}^{m} z_{ji}$ and $d_{(m)}(j,j) = \sum_{i=1}^{n} z_{ji}$;
5: Update $\mathbf{F}_{(n)}$ and $\mathbf{F}_{(m)}$ by solving Eq. (12);
6: Update λ due to $\widetilde{\mathbf{L}}$;
7: **until** $\mathbf{Z}$ has exactly k connected components.

Optimize $\mathbf{W}_p$ by fixing other variables With $\mathbf{Z}$, $\mathbf{A}$ and $\boldsymbol{\beta}$ being fixed, $\mathbf{W}_p$ can be optimized as

$$\min_{\mathbf{W}_p} \sum_{p=1}^{v} \beta_p^2 \left\|\mathbf{X}_p - \mathbf{W}_p\mathbf{A}\mathbf{Z}\right\|_F^2, s.t. \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_l. \quad (18)$$

Since $\mathbf{W}_p$ is independent in each view, we transform Eq. (18) into the following equivalent formulation,

$$\max_{\mathbf{W}_p} \mathrm{Tr}\left(\mathbf{W}_p^\top \mathbf{G}_p\right), s.t. \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_l, \quad (19)$$

where $\mathbf{G}_p = \mathbf{X}_p\mathbf{Z}^\top\mathbf{A}^\top$. The optimal solution of $\mathbf{W}_p$ is $\mathbf{U}\mathbf{V}^\top$ with $\mathbf{U}$ and $\mathbf{V}$ to be the singular matrix of $\mathbf{G}_p$.

Optimize $\mathbf{A}$ by fixing other variables Fixing $\mathbf{Z}, \mathbf{W}_p$ and $\boldsymbol{\beta}$, the optimization of $\mathbf{A}$ can be rewritten as,

$$\min_{\mathbf{A}} \sum_{p=1}^{v} \beta_p^2 \left\|\mathbf{X}_p - \mathbf{W}_p\mathbf{A}\mathbf{Z}\right\|_F^2, s.t. \mathbf{A}^\top \mathbf{A} = \mathbf{I}_m. \quad (20)$$

The above equation equals to the following form by removing unrelated items,

$$\max_{\mathbf{A}} \mathrm{Tr}\left(\mathbf{A}^\top \mathbf{B}\right), s.t. \mathbf{A}^\top \mathbf{A} = \mathbf{I}_m, \quad (21)$$

where $\mathbf{B} = \sum_{p=1}^{v} \beta_p^2 \mathbf{W}_p^\top \mathbf{X}_p \mathbf{Z}^\top$. Supposing $\mathbf{U}\Sigma\mathbf{V}^\top$ to be the singular value decomposition result of $\mathbf{B}$, the optimal solution of Eq. (21) is $\mathbf{U}\mathbf{V}^\top$.

Optimize $\boldsymbol{\beta}$ by fixing other variables When $\mathbf{Z}, \mathbf{W}_p$ and $\mathbf{A}$ are fixed, the objective function with respect to $\boldsymbol{\beta}$ can be formulated as

$$\min_{\boldsymbol{\beta}} \sum_{p=1}^{v} \beta_p^2 \mathbf{M}_p, s.t. \boldsymbol{\beta}^\top \mathbf{1} = 1, \boldsymbol{\beta} \geq 0, \quad (22)$$

where $\mathbf{M}_p = \left\|\mathbf{X}_p - \mathbf{W}_p\mathbf{A}\mathbf{Z}\right\|_F^2$. We can obtain the optimal β_p by Cauchy-Buniakowsky-Schwarz inequality as

$$\beta_p = \frac{\frac{1}{\mathbf{M}_p}}{\sum_{p=1}^{v} \frac{1}{\mathbf{M}_p}}, \quad (23)$$

The whole pipeline of solving Eq. (6) is summarized in Algorithm 2.

Algorithm 2: EOMSC-CA

Input: Multi-view dataset $\{\mathbf{X}_p\}_{p=1}^{v}$, cluster number k.
Output: $\mathbf{Z} \in \mathbb{R}^{m \times n}$ with exactly k connected components.

```
Initialize W,A,Z. Initialize β_p = 1/v.
repeat
    Compute C = Σ_{p=1}^{v} β_p^2 X_p^⊤ W_p A.
    Update Z by Algorithm 1;
    Update W_p by solving Eq. (19);
    Update A by solving Eq. (21);
    Update β by solving Eq. (23);
until converged.
```

Complexity Analysis

Due to the application of anchor strategy, EOMSC-CA has a low computational complexity. Specifically, it takes $\mathcal{O}\left(nm^2t + m^3t + nmlt + nmdt\right)$ to perform Algorithm 1 to construct $\mathbf{Z}$ with t being the iteration number, where solving Eq. (12) costs $\mathcal{O}\left(nm^2 + m^3\right)$ and solving Eq. (17) costs $\mathcal{O}\left(nl(d+m)\right)$ in each iteration. In the process of optimizing $\mathbf{W}_p$, performing SVD in each view needs $\mathcal{O}\left(d_p l^2\right)$ and matrix multiplication needs $\mathcal{O}\left(md_p(n+l)\right)$. When updating $\mathbf{A}$, it cost $\mathcal{O}\left(ml^2\right)$ for SVD and $\mathcal{O}\left(nl(m+d_p)\right)$ for matrix multiplication. And calculating β_p costs only $\mathcal{O}\left(1\right)$. Totally, the main computational complexity in Algorithm 2 is $\mathcal{O}\left(n(m^2t + mlt + mdt + dl) + m^3t + dl^2 + mdl + ml^2\right)$. In our algorithm, $m \ll n$, $d \ll n$, $l \ll n$ and $t \ll n$. Therefore, the optimization step in our algorithm is a linear complexity to sample numbers.

After obtaining a k-connected graph $\mathbf{Z}$, we perform a linear graph algorithm on it instead of performing SVD on graph and then k-means to get results. The computational complexity of this step is $\mathcal{O}\left(nml\right)$, which is also linear to the number of samples. By contrast, most of the MVSC algorithms have $\mathcal{O}\left(n^3\right)$ complexity.

Dataset	#Samples	#View	#Class
ORL_mtv	400	3	40
Caltech101-7	1474	6	7
Mfeat	2000	6	10
Caltech101-20	2386	6	20
Caltech101-all	9144	5	102
SUNRGBD	10335	2	45
NUSWIDEOBJ	30000	5	31
AWA	30475	6	50
YoutubeFace	101499	5	31

Table 2: Datasets used in our experiments.

Experiments

To evaluate the performance of EOMSC-CA, we conduct experiments in this section.

Benchmark Datasets

We perform experiments on nine widely used multi-view benchmark datasets: ORL_mtv, Caltech101-7, Mfeat, Caltech101-20, Caltech101-all, SUNRGBD, NUSWIDEOBJ, AWA, YoutubeFace. The details of them are shown in Table 2. Specifically, ORL_mtv contains 400 images in 40 classes. Caltech101-7 with 1474 instances in 7 categories and Caltech101-20 with 2386 subjects in 20 classes are both subsets of the image dataset Caltech101 (Fei-Fei, Fergus, and Perona 2004). Mfeat was generated from UCI machine learning repository, which consists of the digits from 0 to 9. SUNRGBD (Song, Lichtenberg, and Xiao 2015) consists of 10335 indoor scene images spread

Dataset	MLRSSC (2018)	AMGL (2016)	SFMC (2020)	RMKM (2013)	BMVC (2018)	LMVSC (2020)	MSGL (2021)	FPMVS (2021)	Ours
Hyper-parameter number	3	1	0	1	4	1	2	0	0
ACC									
ORL_mtv	0.0500	**[0.6951]**	0.6150	0.4350	0.4875	0.5862	0.2100	0.5251	**0.6225**
Caltech101-7	0.6090	0.3960	0.6526	0.2877	0.2239	0.3535	**0.7347**	0.5694	**[0.8351]**
Mfeat	0.2000	**[0.8262]**	0.7575	0.6710	0.6935	0.8170	0.7545	**0.8225**	**0.8220**
Caltech101-20	0.3600	0.2878	0.5947	0.3961	0.1689	0.2929	0.4790	**[0.6542]**	**0.6404**
Caltech101-all	0.1085	0.1401	0.1777	0.1650	0.2123	0.1449	0.1412	**[0.2855]**	**0.2232**
SUNRGBD	0.1391	0.0981	0.1136	0.1771	0.1669	0.1809	0.1310	**0.2335**	**[0.2370]**
NUSWIDEOBJ	N/A	N/A	0.1221	0.1328	0.1299	0.1476	0.1204	**0.1922**	**[0.1968]**
AWA	N/A	N/A	0.0390	0.0656	**0.0867**	0.0723	0.0802	**[0.0893]**	**0.0870**
YoutubeFace	N/A	N/A	N/A	N/A	0.0897	0.1403	0.1671	**0.2302**	**[0.2650]**
NMI									
ORL_mtv	0.1583	**0.8717**	**0.8269**	0.7010	0.6773	0.7880	0.4372	0.7443	**[0.8815]**
Caltech101-7	0.1788	0.4408	**[0.5629]**	0.1411	0.0470	0.3384	0.3794	**0.5391**	**0.5219**
Mfeat	0.2863	**0.8674**	**[0.8684]**	0.6533	0.6605	0.7609	0.7654	0.7930	**0.8319**
Caltech101-20	0.2008	0.4760	0.4285	0.5034	0.1626	0.4187	0.3113	**[0.6323]**	**0.5109**
Caltech101-all	0.0474	**0.3529**	0.2613	0.3494	**[0.4246]**	0.3332	0.2612	0.3415	0.2470
SUNRGBD	0.0421	0.1840	0.0230	0.2531	0.1954	**[0.2550]**	0.0933	**0.2418**	**0.2249**
NUSWIDEOBJ	N/A	N/A	0.0096	**[0.1435]**	0.1290	0.1276	0.0573	0.1326	**0.1327**
AWA	N/A	N/A	0.0034	0.0738	**[0.1372]**	0.0855	0.0792	**0.1047**	**0.0972**
YoutubeFace	N/A	N/A	N/A	N/A	0.0593	**0.1179**	0.0007	**[0.2339]**	0.0032
Fscore									
ORL_mtv	0.0582	**0.5123**	0.3066	0.3068	0.3054	0.4599	0.0517	0.3793	**[0.6200]**
Caltech101-7	0.5287	0.4035	0.6409	0.2879	0.2275	0.3790	**0.6524**	0.5632	**[0.7967]**
Mfeat	0.2739	**[0.8083]**	0.7111	0.5922	0.5879	0.7252	0.7011	0.7559	**0.7701**
Caltech101-20	0.3069	0.2182	0.3150	0.3565	0.1138	0.2564	0.4174	**[0.6904]**	**0.6471**
Caltech101-all	0.0502	0.0406	0.0462	0.1486	**0.1854**	0.1047	0.0864	**[0.2083]**	0.1083
SUNRGBD	**0.1291**	0.0644	0.1212	0.1168	0.1019	0.1159	0.0949	**[0.1597]**	**0.1539**
NUSWIDEOBJ	N/A	N/A	0.1140	0.0865	0.0881	0.0932	0.0856	**0.1343**	**[0.1367]**
AWA	N/A	N/A	0.0457	0.0359	**0.0559**	0.0365	0.0421	**[0.0626]**	**0.0599**
YoutubeFace	N/A	N/A	N/A	N/A	0.0579	0.0831	**0.1511**	0.1396	**[0.1641]**

Table 3: ACC, NMI and Fscore comparison of different clustering algorithms on datasets. The best results are highlighted in bold with brackets, and the second best and comparable results are bolded.

over 45 classes. NUSWIDEOBJ (Chua et al. 2009) is an object recognition database with 30000 objects. AWA contains 50 different animals with their six features. YoutubeFace is produced from YouTube with 101499 instances.

Experimental Setup

We compare our method with the following eight STOA MVC methods: **MLRSSC** (Brbić and Kopriva 2018); **AMGL** (Nie et al. 2016a); **SFMC** (Li et al. 2020); **RMKM** (Cai, Nie, and Huang 2013); **BMVC** (Zhang et al. 2018); **LMVSC** (Kang et al. 2020); **MSGL** (Kang et al. 2021); **FRMVS** (Wang et al. 2021b);.

The proposed method has no hyper-parameters to be tuned, but we need to determine the anchors number and the dimension of anchor matrix. In the experiments, the anchors number and the dimension of anchor matrix are both traverse $[k, 2k, ..., 7k]$ where k is the clustering number of each dataset. For the compared algorithms, we search their best parameters for fairness. Moreover, we run 50 times k-means and report the best result. To evaluate the clustering performance, we employ three widely used criteria including accuracy (ACC), normalized mutual information (NMI) and Fscore. All the experiments are performed on a desktop with Intel Core i9-10900X CPU and 64G RAM, MATLAB 2019b(64-bit).

Experimental Results

Table 3 compares the clustering performance of the EOMSC-CA with other methods on nine benchmark datasets. We mark the optimal results in red, use bold for the suboptimal and close results, and 'N/A' to indicate out-of-memory issue. According to the results, we have the following conclusions:

- The proposed algorithm can achieve the best on the nine datasets and is close to the best on the rest datasets, which proves its effectiveness in multi-view clustering.
- AMGL achieve the best performance among all methods on ORL_mtv. However, AMGL cannot handle large-scale tasks due to high complexity. Compared with the above method, EOMSC-CA not only has close performance on the corresponding datasets, but also can be performed on large-scale datasets.

The comparison of NMI and Fscore are also reported in Table 3. We conclude that our EOMSC-CA has considerable performance among baseline algorithms. To further prove

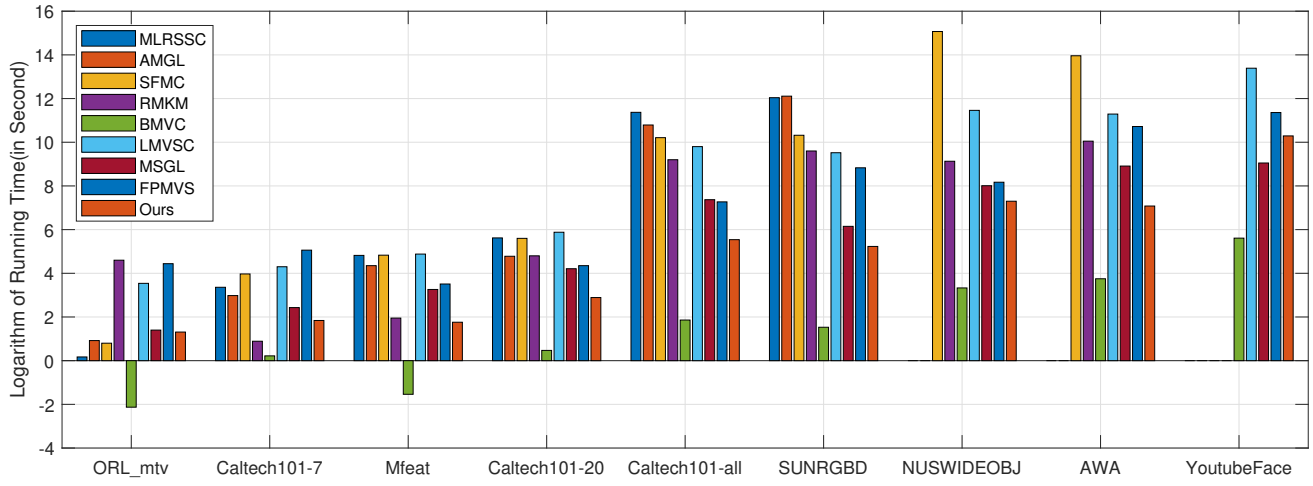
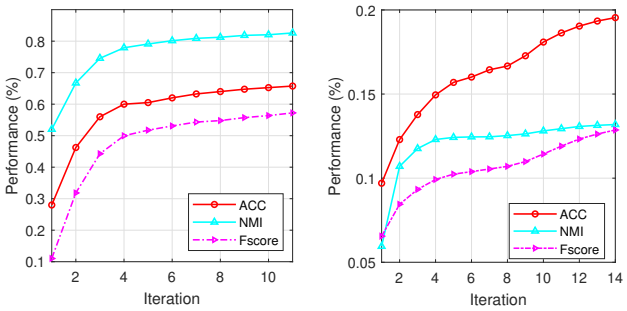
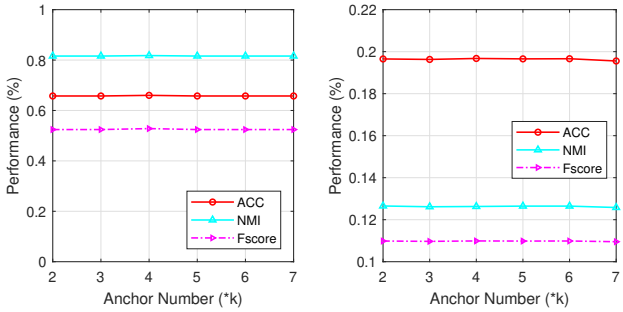


Figure 2: Run time of different algorithms on each baseline datasets.



(a) ORL_mtv (b) NUSWIDEOBJ

Figure 3: The clustering ACC, NMI and Fscore of our algorithm w.r.t the iterations.



(a) ORL_mtv (b) NUSWIDEOBJ

Figure 4: Sensitivity analysis of anchor number for our method over ORL_mtv and NUSWIDEOBJ.

the effectiveness of our algorithm optimization, we also conduct experiments on evolution of the clustering performance with variation iterations. As shown in Figure 3, the ACC, NMI and Fscore monotonically increase at each iteration and stabilize in the last few iterations. The results verify EOMSC-CA's effectiveness in optimization.

Running Time Comparison

The run time of various algorithms on nine datasets are compared in Figure 2. Compared with most STOA multi-view methods, EOMSC-CA has the better computational efficiency as shown in the figure. BMVC and MSGL run faster than other methods, while they have four and two hyper-parameters to be chosen with poorer clustering performance. Therefore, it can be demonstrated that our method has high efficiency on multi-view clustering, which is proved by theoretical and experimental results.

Sensitivity analysis

In order to analyse the influence of the amount of anchors on the clustering performance, we fix l and conduct a comparative experiment on ORL_mtv and NUSWIDEOBJ. As shown in Figure 4, our algorithm is not greatly affected by the anchor number.

Conclusion

In this paper, we propose a novel multi-view subspace clustering method termed as EOMSC-CA. In contrast to most anchor-based methods, the selection of anchor points and the construction of subspace graphs are optimized jointly to improve the clustering performance in our method. Moreover, by imposing a connectivity constraint, we directly generate the clustering result without post-processing. Extensive experiments on real-world datasets prove the effectiveness and efficiency of the proposed method.

Acknowledgements

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61922088, 61906020, 61872371 and 62006237).

References

Adler, A.; Elad, M.; and Hel-Or, Y. 2015. Linear-time subspace clustering via bipartite graph modeling. *IEEE transactions on neural networks and learning systems*, 26(10): 2234–2246.

Brbić, M.; and Kopriva, I. 2018. Multi-view low-rank sparse subspace clustering. *Pattern Recognition*, 73: 247–258.

Cai, X.; Nie, F.; and Huang, H. 2013. Multi-view k-means clustering on big data. In *Twenty-Third International Joint conference on artificial intelligence*.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–594.

Chen, X.; and Cai, D. 2011. Large scale spectral clustering with landmark-based representation. In *Twenty-fifth AAAI conference on artificial intelligence*.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, 1–9.

Ding, Z.; and Fu, Y. 2014. Low-rank common subspace for multi-view learning. In *2014 IEEE international conference on Data Mining*, 110–119. IEEE.

Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2765–2781.

Fan, K. 1949. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11): 652.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, 178–178. IEEE.

Gao, H.; Nie, F.; Li, X.; and Huang, H. 2015. Multi-view subspace clustering. In *Proceedings of the IEEE international conference on computer vision*, 4238–4246.

Gao, Q.; Xia, W.; Wan, Z.; Xie, D.; and Zhang, P. 2020. Tensor-SVD based graph learning for multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3930–3937.

Kang, Z.; Guo, Z.; Huang, S.; Wang, S.; Chen, W.; Su, Y.; and Xu, Z. 2019. Multiple Partitions Aligned Clustering. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 2701–2707. International Joint Conferences on Artificial Intelligence Organization.

Kang, Z.; Lin, Z.; Zhu, X.; and Xu, W. 2021. Structured Graph Learning for Scalable Subspace Clustering: From Single View to Multiview. *IEEE Transactions on Cybernetics*.

Kang, Z.; Zhou, W.; Zhao, Z.; Shao, J.; Han, M.; and Xu, Z. 2020. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4412–4419.

Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2020. Multi-view clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2012. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 171–184.

Liu, J.; Liu, X.; Yang, Y.; Guo, X.; Kloft, M.; and He, L. 2021a. Multiview subspace clustering via co-training robust data representation. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, X.; Liu, L.; Liao, Q.; Wang, S.; Zhang, Y.; Tu, W.; Tang, C.; Liu, J.; and Zhu, E. 2021b. One Pass Late Fusion Multiview Clustering. In *International Conference on Machine Learning*, 6850–6859. PMLR.

Liu, X.; Zhou, S.; Liu, L.; Tang, C.; Wang, S.; Liu, J.; and Zhang, Y. 2021c. Localized Simple Multiple Kernel K-Means. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9293–9301.

Liu, X.; Zhu, X.; Li, M.; Tang, C.; Zhu, E.; Yin, J.; and Gao, W. 2019. Efficient and effective incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4392–4399.

Lu, C.; Yan, S.; and Lin, Z. 2016. Convex sparse spectral clustering: Single-view to multi-view. *IEEE Transactions on Image Processing*, 25(6): 2833–2843.

Nie, F.; Li, J.; Li, X.; et al. 2016a. Parameter-free auto-weighted multiple graph learning: a framework for multi-view clustering and semi-supervised classification. In *IJCAI*, 1881–1887.

Nie, F.; Wang, X.; Deng, C.; and Huang, H. 2017. Learning a structured optimal bipartite graph for co-clustering. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4132–4141.

Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 977–986.

Nie, F.; Wang, X.; Jordan, M.; and Huang, H. 2016b. The constrained laplacian rank algorithm for graph-based clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Ou, Q.; Wang, S.; Zhou, S.; Li, M.; Guo, X.; and Zhu, E. 2020. Anchor-based multiview subspace clustering with diversity regularization. *IEEE MultiMedia*, 27(4): 91–101.

Song, S.; Lichtenberg, S. P.; and Xiao, J. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 567–576.

Sun, M.; Zhang, P.; Wang, S.; Zhou, S.; Tu, W.; Liu, X.; Zhu, E.; and Wang, C. 2021. Scalable Multi-view Subspace Clustering with Unified Anchors. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3528–3536.

Tang, C.; Zhu, X.; Liu, X.; Li, M.; Wang, P.; Zhang, C.; and Wang, L. 2018. Learning a joint affinity graph for multi-view subspace clustering. *IEEE Transactions on Multimedia*, 21(7): 1724–1736.

Vidal, R. 2011. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2): 52–68.

Wang, C.-D.; Chen, M.-S.; Huang, L.; Lai, J.-H.; and Philip, S. Y. 2020. Smoothness regularized multiview subspace clustering with kernel learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Wang, S.; Liu, X.; Liu, L.; Zhou, S.; and Zhu, E. 2021a. Late Fusion Multiple Kernel Clustering With Proxy Graph Refinement. *IEEE Transactions on Neural Networks and Learning Systems*.

Wang, S.; Liu, X.; Zhu, E.; Tang, C.; Liu, J.; Hu, J.; Xia, J.; and Yin, J. 2019. Multi-view Clustering via Late Fusion Alignment Maximization. In *IJCAI*, 3778–3784.

Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2021b. Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance. *IEEE Transactions on Image Processing*, 31: 556–568.

Wang, Y.; Wu, L.; Lin, X.; and Gao, J. 2018. Multiview spectral clustering via structured low-rank matrix factorization. *IEEE transactions on neural networks and learning systems*, 29(10): 4833–4843.

Xu, N.; Guo, Y.; Zheng, X.; Wang, Q.; and Luo, X. 2018. Partial multi-view subspace clustering. In *Proceedings of the 26th ACM International conference on multimedia*, 1794–1801.

Zhan, K.; Zhang, C.; Guan, J.; and Wang, J. 2017. Graph learning for multiview clustering. *IEEE transactions on cybernetics*, 48(10): 2887–2895.

Zhang, C.; Fu, H.; Liu, S.; Liu, G.; and Cao, X. 2015. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE international conference on computer vision*, 1582–1590.

Zhang, G.-Y.; Zhou, Y.-R.; Wang, C.-D.; Huang, D.; and He, X.-Y. 2021. Joint representation learning for multi-view subspace clustering. *Expert Systems with Applications*, 166: 113913.

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

Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2018. Binary multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1774–1782.

Zhou, S.; Liu, X.; Li, M.; Zhu, E.; Liu, L.; Zhang, C.; and Yin, J. 2020a. Multiple Kernel Clustering With Neighbor-Kernel Subspace Segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4): 1351–1362.

Zhou, S.; Liu, X.; Liu, J.; Guo, X.; Zhao, Y.; Zhu, E.; Zhai, Y.; Yin, J.; and Gao, W. 2020b. Multi-view spectral clustering with optimal neighborhood Laplacian matrix. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6965–6972.

Zhou, S.; Ou, Q.; Liu, X.; Wang, S.; Liu, L.; Wang, S.; Zhu, E.; Yin, J.; and Xu, X. 2021. Multiple Kernel Clustering With Compressed Subspace Alignment. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.