Efficient Orthogonal Multi-view Subspace Clustering

Man-Sheng Chen
Sun Yat-sen University
Guangzhou, China
chenmsh27@mail2.sysu.edu.cn

Chang-Dong Wang
Sun Yat-sen University
Guangzhou, China
Corresponding author
changdongwang@hotmail.com

Dong Huang
South China Agricultural University
Guangzhou, China
huangdonghere@gmail.com

Jian-Huang Lai
Sun Yat-sen University
Guangzhou, China
stsljh@mail.sysu.edu.cn

Philip S. Yu
University of Illinois at Chicago
Chicago, USA
psyu@cs.uic.edu

ABSTRACT

Multi-view subspace clustering targets at clustering data lying in a union of low-dimensional subspaces. Generally, an $n \times n$ affinity graph is constructed, on which spectral clustering is then performed to achieve the final clustering. Both graph construction and graph partitioning of spectral clustering suffer from quadratic or even cubic time and space complexity, leading to difficulty in clustering large-scale datasets. Some efforts have recently been made to capture data distribution in multiple views by selecting key anchor bases beforehand with k-means or uniform sampling strategy. Nevertheless, few of them pay attention to the algebraic property of the anchors. How to learn a set of high-quality orthogonal bases in a unified framework, while maintaining its scalability for very large datasets, remains a big challenge. In view of this, we propose an Efficient Orthogonal Multi-view Subspace Clustering (OMSC) model with almost linear complexity. Specifically, the anchor learning, graph construction and partition are jointly modeled in a unified framework. With the mutual enhancement of each other, a more discriminative and flexible anchor representation and cluster indicator can be jointly obtained. An alternate minimizing strategy is developed to deal with the optimization problem, which is proved to have linear time complexity *w.r.t.* the sample number. Extensive experiments have been conducted to confirm the superiority of the proposed OMSC method. The source codes and data are available at https://github.com/ManshengChen/Code-for-OMSC-master.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Cluster analysis** → Multi-view clustering.

KEYWORDS

multi-view clustering, large-scale, orthogonal bases, partition

ACM Reference Format:
Man-Sheng Chen, Chang-Dong Wang, Dong Huang, Jian-Huang Lai, and Philip S. Yu. 2022. Efficient Orthogonal Multi-view Subspace Clustering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539282

1 INTRODUCTION

In real-world scenarios, data can be collected from multiple sources or feature extractors [5, 6, 9, 16–18, 30, 32, 43]. For instance, many computer vision objects involve instances represented by image and video. News can be reported by different languages, such as Chinese, English and Spanish. A person can be described by face, fingerprint, iris and so on. How to effectively integrate these heterogeneous information to perform clustering is essential in an unsupervised learning task, motivating the development of multi-view clustering. In multi-view clustering, there are two important principles [42, 43], i.e., the consensus principle, aiming to maximize the agreements among multiple views, and the complementary principle, meaning that each view of the data contains some information the other views do not have.

In the recent few years, considerable efforts have been made in the field of multi-view clustering to explore the diverse and complementary information among multiple views [4, 13, 23, 27, 31, 35, 39, 45, 46]. Among them, studies on multi-view subspace clustering have attracted an increasing amount of attention, whose goal is to discover the underlying low-dimensional subspaces or groups. For instance, Cao et al. employed the Hilbert Schmidt Independence Criterion (HSIC) as a regularization term to explore the complementary information between different views [2]. Zhang et al. attempted to search for such an underlying latent subspace of multiple views [48]. The work in [5] constructed the global graph and the clustering indicator matrix based on the learned potential embedding space in a unified framework. In [24], Luo et al. simultaneously learned consistency and specificity in subspace representation. Extended from low-rank representation segmentation (LRR) [22], the work in [47] imposed unfolding based low rank on the tensor, which is stacked by multiple subspace representations from different views. However, most of the existing works suffer from the high computational cost (typically quadratic or even cubic complexity, see section 2 for details), limiting the their efficiency when dealing with large-scale multi-view data.

KDD '22, August 14–18, 2022, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539282

To address this issue, some multi-view clustering methods aiming at large-scale data have been developed, which can be divided into two main categories, i.e., matrix factorization based methods [1, 28, 36, 44] and anchor graph based models [15, 20, 21]. For the matrix factorization based algorithms equivalent to the relaxed k-means version, they can improve the computational efficiency due to the unnecessary construction of affinity graph. For instance, Cai et al. integrated heterogeneous representations of large-scale data efficiently [1]. In [34], the cluster structure was explored with a constrained factor matrix. Further, Han et al. regarded the intermediate factor matrix as a diagonal matrix in the 3-factor factorization, reducing the number of matrix multiplication in optimization [12]. The work in [28] simultaneously implemented clustering task on row and column of the input data. However, due to their direct factorization of the original data matrix, the efficiency of this kind of methods significantly decreases when there is large data feature dimension. The anchor graph based models often can achieve better performance, since they would adopt the generated anchor bases and original data to construct the corresponding anchor graph. For instances, Li et al. constructed a fusion graph for multiple anchor graphs by a local manifold fusion technology. In [20], Kang at al. proposed an anchor graph-based subspace clustering method, where a smaller graph for each view is studied according to the constructed anchor bases. Further, the work in [38] explored the common anchor graph with the guidance of consensus anchor bases in multi-view data. Likewise, Sun et al. learned anchors graph based on the underlying data distribution [29].

Briefly, the above anchor based algorithms aim at sampling a few anchors to encode the original multi-view data, where the anchors can also be viewed as a set of bases. Despite significant success, few of them have paid attention to the algebraic property of the bases. The set of bases (i.e., anchors) expressing data samples can be constrained to be orthogonal so that each base is independent and more discriminative from each other, which may further promote the expression of diverse samples. However, how to learn a set of high-quality orthogonal bases in a unified framework, while maintaining its scalability for very large datasets, remains a big challenge.

Aiming to address the aforementioned problem, in this paper, we propose a novel method for large-scale multi-view data, called Efficient Orthogonal Multi-view Subspace Clustering (OMSC), in which the anchor learning, graph construction and partition are jointly modeled in a unified framework. The three components boost each other to jointly promote clustering quality. In particular, a consensus anchor graph is adaptively learned by the projected unified anchors from multiple views, correlated with the cross-view complementary information and discriminative anchor structure. Further, under the orthogonality constraint of actual bases, a factor matrix with rigorous clustering interpretation is constrained to be cluster indicator matrix.

The main contributions of this paper are summarized as follows:

- We propose an efficient multi-view subspace clustering method called Efficient Orthogonal Multi-view Subspace Clustering (OMSC) where the flexible unified anchor representation, consensus graph and final clusterings are jointly explored in a unified framework.
- An alternate minimizing strategy is developed to deal with the optimization problem, by which the proposed method is proved to have linear time complexity *w.r.t.* the number of samples.
- Extensive experiments conducted on several big datasets have demonstrated the superiority of the proposed method compared with the state-of-the-art methods. Furthermore, to the best of our knowledge, it is the first time to jointly consider the orthogonal anchor learning, graph construction and partition in a unified model for large-scale data.

The rest of this paper is organized as follows. In Section 2, we provide the main notations and basic preliminaries of Multi-View Subspace Clustering. The proposed Efficient Orthogonal Multi-view Subspace Clustering approach is described in Section 3 in which the optimization algorithm and the time computational analysis as well as space complexity analysis are provided. In Section 4, the experimental results on seven datasets are reported. Finally, this paper is concluded in Section 5.

2 PRELIMINARIES

Notations. Throughout this paper, matrices are represented as uppercase letters. The i-th row of matrix B is written as $B_{i,:}$, with its j-th entry being B_{ij}. $\|B\|_F^2$ denotes the square of Frobenius norm of matrix B. The trace of B can be denoted as $Tr(B)$. We denote the v-th view representation of B as $B^{(v)}$. Particularly, I_d represents the d-th dimension identity matrix, and $\mathbf{1}$ represents the vector with all entries being 1.

Multi-View Subspace Clustering. Self-expressive subspace clustering assumes that each instance can be reconstructed as a linear combination of the other instances [14]. Mathematically, by minimizing the reconstruction loss, the corresponding self-representation matrix Z^* can be computed by

$$\begin{aligned}&\min_{Z} \gamma\|X - A(X)Z\|_q + \Omega(Z),\\ &\text{s.t. } Z \in C,\end{aligned} \tag{1}$$

where $\gamma > 0$ is the trade-off factor. $X \in \mathbb{R}^{d\times n}$ represents the input data matrix with its column being a sample vector. $A(X)$ is a learnable dictionary matrix, and we usually just set it as $A(X) = X$. $\|\cdot\|_q$ stands for a proper norm, $\Omega(Z)$ and C are respectively the regularization term and constraint set on Z. Specifically, Z is also named with affinity graph, and the existing literatures distinguish each other by adopting different regularization terms $\Omega(Z)$ or constraint sets C [2, 11, 40].

As an extension, more and more attention has been paid on multi-view subspace clustering. Accordingly, given a multi-view dataset $\left\{\mathbf{X}^{(v)} \in \mathbb{R}^{d_v\times n}\right\}_{v=1}^{m}$, the objective formulation can be commonly rewritten as [11, 20]

$$\begin{aligned}&\min_{Z^{(v)}} \gamma \sum_{v=1}^{m} \|X^{(v)} - X^{(v)}Z^{(v)}\|_F^2 + \Omega(Z^{(v)}),\\ &\text{s.t. } Z^{(v)} \geqslant 0, Z^{(v)}\mathbf{1} = 1,\end{aligned} \tag{2}$$

where $Z^{(v)} \in \mathbb{R}^{n\times n}$ is nonnegative and $\sum_{j=1} Z_{ij}^{(v)} = 1$. Due to the $n \times n$ graph construction, the existing algorithms require a lot of computational time and space storage. After obtaining the affinity representation, spectral clustering algorithm would be employed on it to achieve the final clustering performance, needing extra $O\left(n^2k\right)$ complexity. Hence, the scalability of multi-view subspace clustering methods is dramatically limited especially when the sample number n is large.

3 THE PROPOSED METHOD

In this section, the motivation and formulation of the proposed OMSC method are firstly introduced, followed by the detailed optimization process. In addition, we conduct an analysis about the computational time and space usage to demonstrate the time and space efficiency of OMSC.

3.1 Motivation and Formulation

In general, each data point is represented as a linear combination of other points by means of the self-representation strategy in multi-view subspace clustering, where a global relationship can be well explored. Nevertheless, the higher optimization time and storage cost associated with the global representation restrict the scalability of multi-view subspace clustering while facing real-world applications with big data. Additionally, there are a small number of instances that are sufficient for reconstructing the underlying subspaces [38], and it is unnecessary and redundant to depict one point with all instances. Hence, the anchor strategy is utilized to select a small set of instances as anchor bases or landmarks to capture the manifold structure. In particular, on the one hand, the existing multi-view anchor-based subspace methods almost select anchor bases by randomly (uniformly) sampling from the original data space, or the clustering centroids obtained by performing k-means, where the anchor bases are fixed, and the anchor learning is isolated from the latter graph construction. On the other hand, few of the existing anchor-based works have paid attention to the algebraic property of the bases (or anchors). How to learn a set of high-quality orthogonal bases in a unified framework, while maintaining its scalability for very large datasets, remains a considerably challenging issue.

Different from the previous works, the proposed OMSC method integrates the three parts, i.e., the anchor learning, graph construction and partition, into a unified framework, where discriminative anchors are learned automatically not based on sampling and the flexible affinity graph as well as final partition can be acquired. According to the assumption that multiple views are originated from one latent representation sharing a consensus underlying data distribution, the cross-view anchors should be consistent in the latent space. In view of this, the respective projection matrix $\left\{P^{(v)}\right\}_{v=1}^{m}$ are constructed aiming at the consensus anchor guidance, and a common affinity graph integrating complementary information across views can be adaptively learned from the projected unified anchors. Mathematically, the optimization goal can be formulated as

$$\begin{aligned}&\min_{\alpha,P^{(v)},A,S}\sum_{v=1}^{m}\alpha_v^2\|X^{(v)}-P^{(v)}AS\|_F^2,\\&\text{s.t. } \alpha^T\mathbf{1}=1, {P^{(v)}}^T P^{(v)}=I_d, A^TA=I_l, S\geq 0, S^T\mathbf{1}=1,\end{aligned}\tag{3}$$

where α_v^2 stands for the view coefficients adaptively learned by its contribution to the consensus affinity graph, and 2 (>1) herein aims to smoothen the weight coefficient distribution. $X^{(v)}\in\mathbb{R}^{d_v\times n}$ is the v-th view of the original data with d_v and n being the dimension of the corresponding view and the size of samples respectively. $P^{(v)}$ is the v-th view anchor projection matrix, enabling to project the unified anchor to the corresponding original feature space. $A\in\mathbb{R}^{d\times l}$ represents the unified anchor matrix, in which d and l are respectively the common dimension across different views and the number of anchors, and $S\in\mathbb{R}^{l\times n}$ is the common affinity matrix whose space complexity reduces to $O\left(l\times n\right)$ instead of $O\left(n^2\right)$ before.

In addition, in order to involve the partition information into the unified framework, nonnegative and orthogonal matrix factorization [7] is eagerly adopted to directly assign cluster labels to data without an extra post-processing step to recover cluster structures from the factor matrix. To be specific, under the orthogonality constraint of the actual bases, a factor matrix with rigorous clustering interpretation is constrained to be cluster indicator matrix. Therefore, we can formulate the overall objective function as follows

$$\begin{aligned}&\min_{\alpha,P^{(v)},A,S,G,F}\sum_{v=1}^{m}\alpha_v^2\|X^{(v)}-P^{(v)}AS\|_F^2+\lambda\|S-GF\|_F^2,\\&\text{s.t. } \alpha^T\mathbf{1}=1, {P^{(v)}}^T P^{(v)}=I_d, A^TA=I_l, S\geq 0, S^T\mathbf{1}=1,\\&G^TG=I_k, F_{ij}\in\{0,1\}, \sum_{i=1}^{k}F_{ij}=1, \forall j=1,2,\cdots,n,\end{aligned}\tag{4}$$

where $G\in\mathbb{R}^{l\times k}$ denotes the centroid matrix, and $F\in\mathbb{R}^{k\times n}$ is the cluster assignment matrix with $F_{ij}=1$ if the j-th instance is assigned to the i-th cluster and 0 otherwise. Throughout this paper, according to the principle that the number of instances required for the underlying subspace should not be less than the number of subspaces, the common dimension is chosen as k, and similarly the number of anchors is correlated with the number of clusters ($l\in\{k,2k,3k\}$). It is worth mentioning that the constraint of projection matrix $P^{(v)}$ avoids A to be pushed arbitrarily, and the common dimension as well as the orthogonal constraint restrict the unified anchor representation to be more discriminative. To the best of our knowledge, different from the existing anchor-based multi-view subspace clustering algorithms [20, 21, 29, 38], this could be the first time that the anchor selection, graph construction and partition are jointly combined into a unified framework aiming at large-scale data, where three parts boost each other in an interplay manner to recover multi-view information and achieve better clustering performance.

3.2 Optimization

In order to solve the optimization problem in Eq. (4), an alternate minimizing algorithm is designed to optimize each variable while fixing the others.

$P^{(v)}$-subproblem: By fixing the other variables, the objective function *w.r.t.* $P^{(v)}$ can be rewritten as

$$\min_{P^{(v)}}\sum_{v=1}^{m}\alpha_v^2\|X^{(v)}-P^{(v)}AS\|_F^2, \text{ s.t. } {P^{(v)}}^T P^{(v)}=I_d.\tag{5}$$

Since each $P^{(v)}$ is independent from each other in terms of distinct views, the optimization problem in Eq. (5) can be transformed into the following model by expanding the Frobenius norm by trace, i.e.,

$$\max_{P^{(v)}} Tr(P^{(v)^T}B^{(v)}), \text{ s.t. } P^{(v)^T}P^{(v)} = I_d, \tag{6}$$

where $B^{(v)} = X^{(v)}S^TA^T$. According to [37], supposing the Singular Value Decomposition (SVD) result of $B^{(v)}$ is $U_B\Sigma_BV_B^T$, the optimal solution of $P^{(v)}$ can be easily obtained by calculating $U_BV_B^T$.

A-**subproblem:** By fixing the other variables, A can be updated by solving the following problem

$$\min_{A} \sum_{v=1}^{m} \alpha_v^2 \|X^{(v)} - P^{(v)}AS\|_F^2, \text{ s.t. } A^TA = I_l. \tag{7}$$

Similar to the optimization of $P^{(v)}$, it is equivalent to solving A by the following form

$$\max_{A} Tr(A^TC), \text{ s.t. } A^TA = I_l, \tag{8}$$

where $C = \sum_{v=1}^{m} \alpha_v^2 P^{(v)^T} X^{(v)} S^T$. Hence, the optimal A is equal to $U_CV_C^T$, where $C = U_C\Sigma_CV_C^T$.

S-**subproblem:** By fixing the other variables, the objective function *w.r.t.* S can be formulated as

$$\begin{aligned} &\min_{S} \sum_{v=1}^{m} \alpha_v^2 \|X^{(v)} - P^{(v)}AS\|_F^2 + \lambda\|S - GF\|_F^2, \\ &\text{s.t. } S \geq 0, S^T\mathbf{1} = 1. \end{aligned} \tag{9}$$

To solve the above optimization problem of S, we can rewrite it as the following Quadratic Programming (QP) problem

$$\begin{aligned} &\min \frac{1}{2} S_{:,j}^T W S_{:,j} + h^T S_{:,j}, \\ &\text{s.t. } S_{:,j}^T\mathbf{1} = 1, S \geq 0, \end{aligned} \tag{10}$$

where $W = 2(\sum_{v=1}^{m} \alpha_v^2 + \lambda)I$ and $h^T = -2\sum_{v=1}^{m} X_{:,j}^{(v)^T} P^{(v)}A - 2\lambda F_{:,j}^T G^T$. Specifically, optimization can be achieved by tackling the QP problem for each column of S.

G-**subproblem:** By fixing the other variables, the optimization for G can be transformed into the following problem

$$\min_{G} \lambda\|S - GF\|_F^2, \text{ s.t. } G^TG = I_k. \tag{11}$$

Similar to $P^{(v)}$ and A, the optimization of G can be rewritten as

$$\max_{G} Tr(G^TJ), \text{ s.t. } G^TG = I_k, \tag{12}$$

where $J = SF^T$. Thus, the optimal solution of G is $U_JV_J^T$, where $J = U_J\Sigma_JV_J^T$.

F-**subproblem:** By fixing the other variables, the objective function *w.r.t.* F can be formulated as the following minimization problem

$$\begin{aligned} &\min_{F} \lambda\|S - GF\|_F^2, \\ &\text{s.t. } F_{ij} \in \{0, 1\}, \sum_{i=1}^{k} F_{ij} = 1, \forall j = 1, 2, \cdots, n. \end{aligned} \tag{13}$$

Since there is one and only one non-zero entry (i.e., 1) in each column of cluster assignment matrix F, it is difficult to optimize F as a whole directly. Fortunately, according to [8], the optimization problem for each object can be solved independently. Thus, for each object, we can have

Algorithm 1 Efficient Orthogonal Multi-view Subspace Clustering

```
Input: Multi-view dataset $\left\{\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}\right\}_{v=1}^{m}$, trade-off parameter
  $\lambda > 0$ and cluster number $k$.
Initialize $P^{(v)} = A = S = 0, G = F = I$.
while not converged do
    Update $P^{(v)}, \forall v$ by solving the problem in Eq. (6).
    Update $A$ by solving the problem in Eq. (8).
    Update $S$ by solving the problem in Eq. (10).
    Update $G$ by solving the problem in Eq. (12).
    Update $F$ via the optimal row obtained by Eq. (15).
    Update $\alpha_v$ by solving the problem in Eq. (17).
end while
  Output: The final consensus cluster assignment matrix $F$.
```

$$\begin{aligned} &\min_{F_{:,j}} \|S_{:,j} - GF_{:,j}\|^2, \\ &\text{s.t. } F_{:,j} \in \{0, 1\}^k, \|F_{:,j}\|_1 = 1. \end{aligned} \tag{14}$$

The above optimization problem can be settled by finding one optimal row i^* in $F_{:,j}$ such that $F_{i^*,j} = 1$ and 0 for the other entries in this column. The optimal row can be found by

$$i^* = \arg\min_{i} \|S_{:,j} - G_{:,i}\|^2. \tag{15}$$

In other words, the optimal cluster assignment can be received by minimizing the distance between the object and the cluster centroid.

α_v-**subproblem:** By fixing the other variables, it is equivalent to solving α_v by the following optimization form

$$\min_{\alpha} \sum_{v=1}^{m} \alpha_v^2 R^{(v)^2}, \text{ s.t. } \alpha^T\mathbf{1} = 1, \tag{16}$$

where $R^{(v)} = \|X^{(v)} - P^{(v)}AS\|_F$. According to Cauchy-Buniakowsky-Schwarz inequality, we can achieve the optimal solution of α_v by

$$\alpha_v = \frac{\frac{1}{R^{(v)}}}{\sum_{v=1}^{m} \frac{1}{R^{(v)}}}. \tag{17}$$

Due to the convex property and optimal solution of each subproblem, the objective formulation will decrease monotonically in each iteration until convergence. For clarity, we summarize the procedure of solving the proposed OMSC model in Algorithm 1.

3.3 Complexity Analysis

Time complexity. The computational burden of OMSC contains the cost of optimization of each variable. Specifically, the time cost of updating $P^{(v)}$ is $O\left(d_vd^2\right)$ and $O\left(d_vdk^2\right)$ for SVD and matrix multiplication. Likewise, to calculate A, it costs $O\left(dl^2\right)$ to perform SVD on C and $O\left(dlk^2\right)$ for matrix multiplication. To update S, the corresponding QP problem costs $O\left(nl^3\right)$ for all data vectors. Similar to calculate $P^{(v)}$ and A, updating G needs $O\left(lk^2\right)$ and $O\left(lk^3\right)$ for SVD and matrix multiplication. For updating F, the time cost needs $O\left(lnk\right)$. When updating α_v, it only costs $O\left(1\right)$.

Table 1: Statistics of the seven real-world datasets.

Datasets	#Object	#View	#Class	View dimension
Notting-Hill	550	3	5	2000, 3304, 6750
Caltech101-20	2386	6	20	48, 40, 254, 1984, 512, 928
VGGFace2-50	34027	4	50	944, 576, 512, 640
YTF-10	38654	4	10	944, 576, 512, 640
YTF-20	63896	4	20	944, 576, 512, 640
YTF-50	126054	4	50	944, 576, 512, 640
YTF-100	195537	4	100	944, 576, 512, 640

Table 2: Complexity analysis on compared algorithms and OMSC. “Max Reported” represents the largest dataset reported in the algorithm.

Method	Space Complexity	Time Cost	Max Reported
AMGL	$O\left(vn^2+nk\right)$	$O\left(n^3\right)$	12643
SwMC	$O\left(n^2+nk\right)$	$O\left(kn^2\right)$	2000
MLAN	$O\left(vn^2+nk\right)$	$O\left(vn^2\right)$	3000
PMSC	$O\left(2vn^2+(v+1)nk\right)$	$O\left(n^3\right)$	2386
LMVSC	$O\left(vk(n+p)\right)$	$O\left(n\right)$	30,000
SMVSC	$O\left(mn+(p+m)k\right)$	$O\left(n\right)$	101,499
FPMVS-CAG	$O\left(kn+(p+k)k\right)$	$O\left(n\right)$	101,499
OMSC	$O\left(n(m+k)+(p+2m)k\right)$	$O\left(n\right)$	195,537

In total, the time complexity of the proposed OMSC method is $O\left((pd^2+pdk^2+dl^2+dlk^2+nl^3+lk^2+lk^3+lnk)t\right)$, where $p=\sum_{v=1}^{m} d_v$ is the summation of feature dimensions, and t denotes the number of iterations of these six parts. Since $n \gg l, n \gg k = d$ in our model, the computational complexity of OMSC is nearly linear to the number of samples $O\left(n\right)$.

Space complexity. The major space usages of the proposed OMSC method are matrices $P^{(v)} \in \mathbb{R}^{d_v \times d}$, $A \in \mathbb{R}^{d \times l}$, $S \in \mathbb{R}^{l \times n}$, $G \in \mathbb{R}^{l \times k}$ and $F \in \mathbb{R}^{k \times n}$. Therefore, the major space complexity of OMSC is $O\left(n(l+k)+(p+2l)k\right)$, nearly linear to the number of samples $O\left(n\right)$ as well.

4 EXPERIMENT

In this section, the effectiveness of the proposed OMSC method is evaluated from the aspects of clustering performance and running time while comparing with five state-of-the-art multi-view clustering methods and four large-scale oriented algorithms. The experiments are conducted on seven widely used datasets, including several large-scale multi-view image datasets. All of the experiments are implemented in Matlab 2021a on a standard Window PC with an Intel 2.4-GHz CPU and 64-GB RAM (64-bit).

4.1 Datasets and Compared Methods

Seven datasets are employed in the evaluation: Notting-Hill, Caltech101-20, VGGFace2-50, YTF-10 (YouTube Faces), YTF-20, YTF-50 and YTF-100. Specifically, the Notting-Hill dataset is extracted from the movie “Notting-Hill” [3, 41]. Caltech101-20 is an object image dataset [10]. VGGFace2-50[1] is a face image dataset. YTF-10, YTF-20, YTF-50 and YTF-100 are different subsets of face videos obtained from YouTube[2]. More detailed descriptions of datasets are introduced in Table 1.

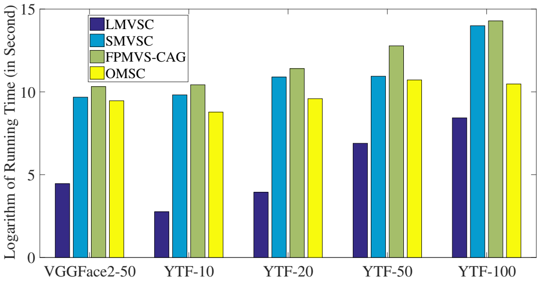


Figure 1: Comparison results on the last five large-scale datasets in terms of running time in seconds.

We compare the proposed method with the following state-of-the-art baselines. **Parameter-free Auto-weighted Multiple Graph Learning (AMGL)** [26] attempts to learn the optimal weights for each graph and achieve optimal performance. **Self-weighted Multiview Clustering with Multiple Graphs (SwMC)** [27] recovers a Laplacian rank constrained graph. **Multi-View Learning With Adaptive Neighbors (MLAN)** [25] simultaneously achieves clustering and local manifold by considering the adaptive neighborhood. **Partition Level Multiview Subspace Clustering (PMSC)** [19] is a unified multi-view subspace clustering model. **Large-scale Multi-view Subspace Clustering in Linear Time (LMVSC)** [20] is a large-scale multi-view subspace clustering where a smaller graph for each view is studied. **Scalable Multiview Subspace Clustering with Unified Anchors (SMVSC)** [29] considers the graph construction based on actual latent data distribution. **Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance (FPMVS-CAG)** [38] learns the subspace representation under consensus anchor guidance. In comparison, the major space complexity and time cost of the compared methods are summarized in Table 2.

4.2 Experimental Setup

For a fair comparison, experiments are conducted twenty times for each method including OMSC as well as the compared algorithms, and the average performance as well as the standard deviation (std. dev.) are reported. Meanwhile, for most of the methods, k-means algorithm is needed to achieve the final clustering performances, and thus it is run ten times to eliminate the random initialization in each experiment. For our method, the trade-off parameter λ is tuned from $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. For the baselines, the best parameters are tuned as suggested by the corresponding papers.

The clustering performance is comprehensively evaluated by four widely used metrics, i.e., accuracy (ACC), normalized mutual information (NMI), purity and Fscore. For all the above evaluation metrics, higher values indicate better clustering performances [33].

[1] https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/.
[2] https://www.cs.tau.ac.il/~wolf/ytfaces/.

Table 3: Comparison results: the mean and standard deviations achieved by different clustering methods on all the corresponding datasets. The best scores are highlighted in bold. N/A indicates that the compared method suffers out-of-memory error.

Datasets	Metric	AMGL	SwMC	MLAN	PMSC	LMVSC	SMVSC	FPMVS-CAG	OMSC
Notting-Hill	ACC	0.4931$_{\pm0.0891}$	0.8400$_{\pm0.0000}$	0.7127$_{\pm0.0000}$	0.8664$_{\pm0.0020}$	0.7927$_{\pm0.0000}$	0.9145$_{\pm0.0000}$	0.7054$_{\pm0.0000}$	**0.9164**$_{\pm0.0000}$
	NMI	0.3492$_{\pm0.1011}$	0.8300$_{\pm0.0000}$	0.7935$_{\pm0.0000}$	0.7354$_{\pm0.0048}$	0.7311$_{\pm0.0000}$	0.8276$_{\pm0.0000}$	0.7178$_{\pm0.0000}$	**0.8354**$_{\pm0.0000}$
	Purity	0.5042$_{\pm0.0869}$	0.8600$_{\pm0.0000}$	0.8455$_{\pm0.0000}$	0.8664$_{\pm0.0020}$	0.8945$_{\pm0.0000}$	0.9145$_{\pm0.0000}$	0.8272$_{\pm0.0000}$	**0.9164**$_{\pm0.0000}$
	Fscore	0.4721$_{\pm0.0640}$	0.8785$_{\pm0.0000}$	0.7461$_{\pm0.0000}$	0.7925$_{\pm0.0028}$	0.7837$_{\pm0.0000}$	0.8746$_{\pm0.0000}$	0.7029$_{\pm0.0000}$	**0.8797**$_{\pm0.0000}$
Caltech101-20	ACC	0.5043$_{\pm0.0272}$	0.5159$_{\pm0.0000}$	0.5996$_{\pm0.0003}$	0.5981$_{\pm0.0489}$	0.4304$_{\pm0.0000}$	0.6165$_{\pm0.0000}$	**0.6638**$_{\pm0.0000}$	0.6635$_{\pm0.0000}$
	NMI	0.5248$_{\pm0.0398}$	0.4287$_{\pm0.0000}$	0.6388$_{\pm0.0003}$	0.5244$_{\pm0.0868}$	0.5553$_{\pm0.0000}$	0.5891$_{\pm0.0000}$	0.6387$_{\pm0.0000}$	**0.6403**$_{\pm0.0000}$
	Purity	0.6685$_{\pm0.0257}$	0.6429$_{\pm0.0000}$	**0.7914**$_{\pm0.0003}$	0.6480$_{\pm0.0439}$	0.7125$_{\pm0.0000}$	0.7015$_{\pm0.0000}$	0.7460$_{\pm0.0000}$	0.7460$_{\pm0.0000}$
	Fscore	0.4123$_{\pm0.0172}$	0.3520$_{\pm0.0000}$	0.5264$_{\pm0.0003}$	0.5474$_{\pm0.0398}$	0.3414$_{\pm0.0000}$	0.6656$_{\pm0.0000}$	0.6917$_{\pm0.0000}$	**0.6940**$_{\pm0.0000}$
VGGFace2-50	ACC	N/A	N/A	N/A	N/A	0.1113$_{\pm0.0000}$	0.1127$_{\pm0.0000}$	0.1098$_{\pm0.0000}$	**0.1150**$_{\pm0.0000}$
	NMI	N/A	N/A	N/A	N/A	0.1332$_{\pm0.0000}$	0.1366$_{\pm0.0000}$	0.1365$_{\pm0.0000}$	**0.1464**$_{\pm0.0000}$
	Purity	N/A	N/A	N/A	N/A	0.1401$_{\pm0.0000}$	0.1157$_{\pm0.0000}$	0.1142$_{\pm0.0000}$	**0.1233**$_{\pm0.0000}$
	Fscore	N/A	N/A	N/A	N/A	0.0524$_{\pm0.0000}$	**0.0604**$_{\pm0.0000}$	0.0597$_{\pm0.0000}$	0.0595$_{\pm0.0000}$
YTF-10	ACC	N/A	N/A	N/A	N/A	0.7566$_{\pm0.0000}$	0.7389$_{\pm0.0000}$	0.7325$_{\pm0.0000}$	**0.7820**$_{\pm0.0000}$
	NMI	N/A	N/A	N/A	N/A	0.7670$_{\pm0.0000}$	0.7980$_{\pm0.0000}$	0.7740$_{\pm0.0000}$	**0.8275**$_{\pm0.0000}$
	Purity	N/A	N/A	N/A	N/A	0.8125$_{\pm0.0000}$	0.7726$_{\pm0.0000}$	0.7621$_{\pm0.0000}$	**0.8298**$_{\pm0.0000}$
	Fscore	N/A	N/A	N/A	N/A	0.7001$_{\pm0.0000}$	0.6982$_{\pm0.0000}$	0.6959$_{\pm0.0000}$	**0.7456**$_{\pm0.0000}$
YTF-20	ACC	N/A	N/A	N/A	N/A	0.7092$_{\pm0.0000}$	0.7125$_{\pm0.0000}$	0.6948$_{\pm0.0000}$	**0.7446**$_{\pm0.0000}$
	NMI	N/A	N/A	N/A	N/A	0.7751$_{\pm0.0000}$	0.7913$_{\pm0.0000}$	0.7790$_{\pm0.0000}$	**0.8170**$_{\pm0.0000}$
	Purity	N/A	N/A	N/A	N/A	0.7614$_{\pm0.0000}$	0.7700$_{\pm0.0000}$	0.7259$_{\pm0.0000}$	**0.7731**$_{\pm0.0000}$
	Fscore	N/A	N/A	N/A	N/A	0.6268$_{\pm0.0000}$	0.6518$_{\pm0.0000}$	0.6261$_{\pm0.0000}$	**0.6835**$_{\pm0.0000}$
YTF-50	ACC	N/A	N/A	N/A	N/A	0.6825$_{\pm0.0000}$	0.6681$_{\pm0.0000}$	0.6851$_{\pm0.0000}$	**0.7152**$_{\pm0.0000}$
	NMI	N/A	N/A	N/A	N/A	0.8098$_{\pm0.0000}$	0.8258$_{\pm0.0000}$	0.8364$_{\pm0.0000}$	**0.8527**$_{\pm0.0000}$
	Purity	N/A	N/A	N/A	N/A	0.7688$_{\pm0.0000}$	0.6926$_{\pm0.0000}$	0.7140$_{\pm0.0000}$	**0.7657**$_{\pm0.0000}$
	Fscore	N/A	N/A	N/A	N/A	0.5794$_{\pm0.0000}$	0.6157$_{\pm0.0000}$	0.6381$_{\pm0.0000}$	**0.6758**$_{\pm0.0000}$
YTF-100	ACC	N/A	N/A	N/A	N/A	0.6006$_{\pm0.0000}$	0.5906$_{\pm0.0000}$	0.5293$_{\pm0.0000}$	**0.6651**$_{\pm0.0000}$
	NMI	N/A	N/A	N/A	N/A	0.7804$_{\pm0.0000}$	0.7991$_{\pm0.0000}$	0.7532$_{\pm0.0000}$	**0.8337**$_{\pm0.0000}$
	Purity	N/A	N/A	N/A	N/A	0.6827$_{\pm0.0000}$	0.6103$_{\pm0.0000}$	0.5446$_{\pm0.0000}$	**0.7141**$_{\pm0.0000}$
	Fscore	N/A	N/A	N/A	N/A	0.5171$_{\pm0.0000}$	0.5035$_{\pm0.0000}$	0.3541$_{\pm0.0000}$	**0.5846**$_{\pm0.0000}$

Since each evaluation measure penalizes or favours specific property in the clustering results, a more comprehensive evaluation can be obtained by reporting the final clustering results via the four diverse metrics.

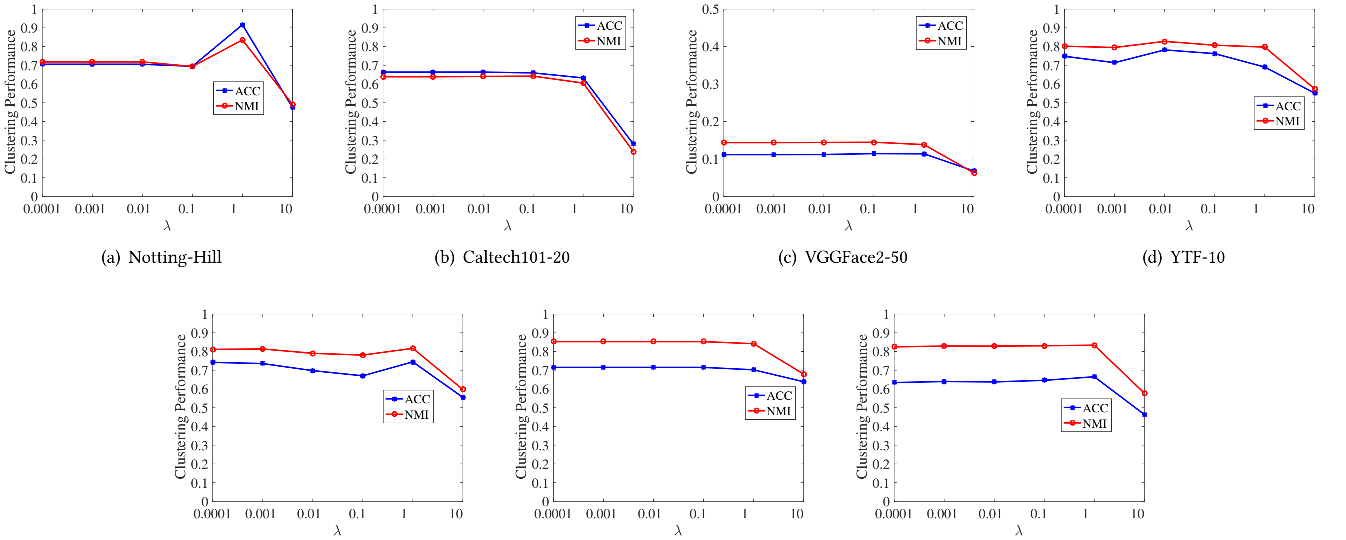
4.3 Comparison Results

4.3.1 Clustering Performance. The detailed clustering results in terms of ACC, NMI, Purity and Fscore obtained by multiple multi-view clustering methods on seven benchmark datasets are reported in Table 3, and the mean as well as the standard deviation (std. dev.) are reported over 20 runs. In the table, we highlight the best performance for distinct datasets in terms of each measure in boldface. Note that N/A indicates that the method suffers out-of-memory error on the corresponding dataset on our device, and therefore no result is reported.

According to the table, the proposed OMSC method generally achieves the best clustering performance on all the testing datasets, and almost always obtains better results than the other multi-view clustering methods for large-scale data. For instances, on the YTF-50 dataset, OMCBD significantly outperforms the other three methods aiming at the larger datasets by achieving improvements of 4.3%, 6.1% and 1.6% in terms of NMI. On the YTF-100 dataset, the performance improvements over the other three methods in terms of NMI are 5.3%, 3.5% and 8.1% respectively.

Meanwhile, compared with the traditional multi-view subspace clustering methods (AMGL, SwMC, MLAN, PMSC), the anchor based subspace methods (LMVSC, SMVSC, FPMVS-CAG and the proposed OMSC method) are more suitable for large-scale datasets, and therefore achieve better performance in most cases, demonstrating the virtue of anchor bipartite graph. Despite of this, the proposed OMSC method still outperforms the other three anchor based methods on the whole, which further verifies the necessity of taking the anchor learning, graph construction and partition jointly into consideration.

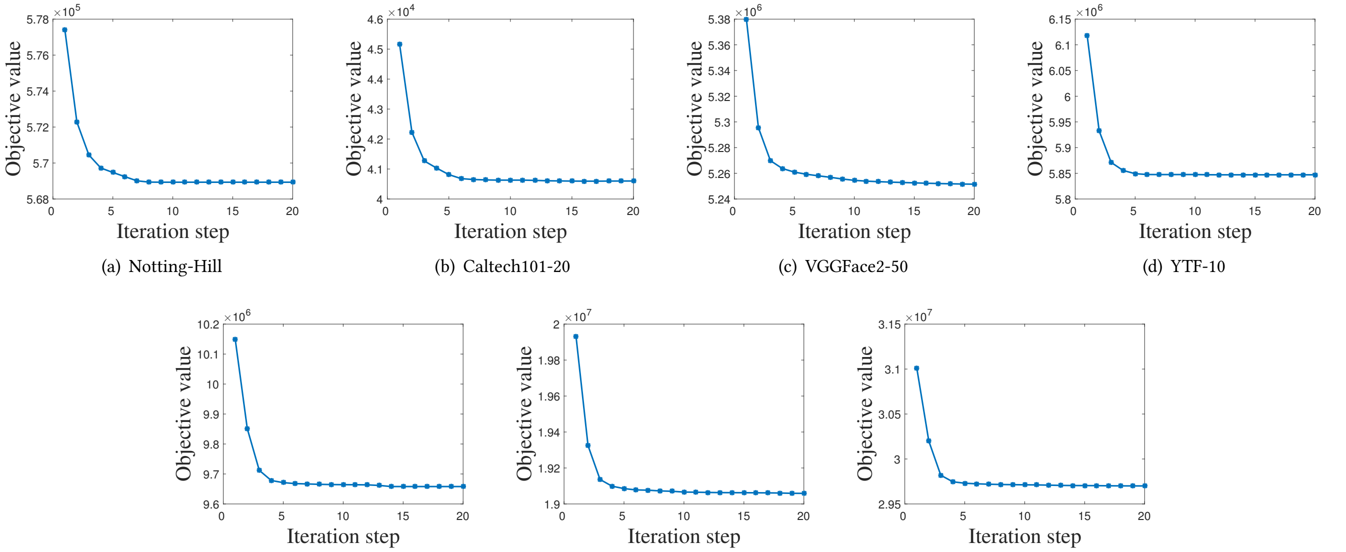
4.3.2 Running Time. Comprehensively, the comparison of running time in seconds consumed by various anchor based subspace algorithms on the last five large-scale datasets are reported in Figure 1. To alleviate the gap between the compared methods and OMSC, the Y-axis is scaled by logarithm in the figure. It can be observed that the proposed OMSC method achieves a relatively good trade-off between the clustering performance and computational burden, ranking the second best on computational time. Although LMVSC costs less computational time, its clustering performance is generally worse than OMSC as it constructs multiple view-specific



(a) Notting-Hill (b) Caltech101-20 (c) VGGFace2-50 (d) YTF-10

(e) YTF-20 (f) YTF-50 (g) YTF-100

Figure 2: Parameter analysis: The performance in terms of ACC and NMI when using different trade-off parameter λ.



(a) Notting-Hill (b) Caltech101-20 (c) VGGFace2-50 (d) YTF-10

(e) YTF-20 (f) YTF-50 (g) YTF-100

Figure 3: Convergence analysis: the objective value as a function of the iteration step.

graphs based on the fixed anchors from views, which neglects the cross-view complementary information and the underlying correlation between anchor bases and the graph. Hence, large-scale multi-view data clustering can be settled by the proposed OMSC method with efficient computation.

Consequently, the above experimental results have well demonstrated that the proposed OMSC method is meaningful and achieves relatively better performances compared with other state-of-the-art multi-view subspace clustering and large-scale oriented methods.

4.4 Parameter Analysis

In this subsection, we conduct parameter analysis on the trade-off parameter λ on different benchmark datasets. The influences of the parameter values in terms of ACC and NMI on all the benchmark datasets are illustrated in Figure 2. According to Figure 2, we can observe that the clustering performance of OMSC is generally stable over the corresponding ranges of parameter values on various datasets. Therefore, the proposed method is relatively robust to the parameter λ. In the meantime, the best results on different datasets can also be obtained from the figure. For instance, the optimal performance can be obtained on the YTF-10 dataset when λ is 0.01. Accordingly, the optimal results on the other datasets can also be acquired from the figure.

4.5 Convergence Analysis

An alternate minimizing algorithm is developed to solve the optimization problem in Eq. (4). Since the objective function in Eq. (4) is non-increasing with the iterations, the algorithm can be guaranteed to convergence finally. In this subsection, the objective values on benchmark datasets are recorded to verify the convergence property of the proposed method. The objective values on all the benchmark datasets are illustrated in Figure 3. Obviously, it can be observed that the corresponding objective value decreases sharply within the first 10 iterations and then stays steady with more iterations, implying that the proposed OMSC method is able to converge after just a few iterations.

5 CONCLUSION

In this paper, we develop a novel unified model termed Efficient Orthogonal Multi-view Subspace Clustering (OMSC) with almost linear complexity. To the best of our knowledge, it is the first time to jointly consider the anchor learning, graph construction and partition in a unified model for large-scale data. Specifically, the consensus anchor graph is adaptively learned by the projected unified anchor bases from multiple views, correlated with the cross-view complementary information and discriminate anchor structure. Meanwhile, under the orthogonality constraint of the actual bases, a factor matrix with rigorous clustering interpretation is constrained to be cluster indicator matrix so as to directly achieve the late partition. An alternate minimizing strategy is developed to deal with the optimization problem, which is proved to have linear time complexity *w.r.t.* the sample number. Extensive experiments conducted on several large-scale datasets have demonstrated the superiority of the proposed method compared with the state-of-the-art methods, including multi-view subspace clustering methods and large-scale oriented methods.

ACKNOWLEDGMENTS

The work was supported by National Key Research and Development Program of China (2021YFF1201200), NSFC (61876193 and 61976097), and NSF (III-1763325, III-1909323, III-2106758, and SaTC-1930941). Chang-Dong Wang is the corresponding author.

REFERENCES

[1] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-View K-Means Clustering on Big Data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2598–2604.

[2] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. 2015. Diversity-induced multi-view subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*. 586–594.

[3] Xiaochun Cao, Changqing Zhang, Chengju Zhou, Huazhu Fu, and Hassan Foroosh. 2015. Constrained Multi-View Video Face Clustering. *IEEE Trans. Image Process.* 24, 11 (2015), 4381–4393.

[4] Mansheng Chen, Ling Huang, Chang-Dong Wang, Dong Huang, and Jian-Huang Lai. 2021. Relaxed multi-view clustering in latent embedding space. *Inf. Fusion* 68 (2021), 8–21.

[5] Man-Sheng Chen, Ling Huang, Chang-Dong Wang, and Dong Huang. 2020. Multi-view Clustering in Latent Embedding Space. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*.

[6] Xiaojun Chen, Wenya Sun, Bo Wang, Zhihui Li, Xizhao Wang, and Yunming Ye. 2019. Spectral Clustering of Customer Transaction Data With a Two-Level Subspace Weighting Method. *IEEE Trans. Cybernetics* 49, 9 (2019), 3230–3241.

[7] C. Ding, Tao Li, Wei Peng, and Haesun Park. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[8] Chris H. Q. Ding and Xiaofeng He. 2005. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. 606–610.

[9] Guowang Du, Lihua Zhou, Yudi Yang, Kevin Lü, and Lizhen Wang. 2021. Deep Multiple Auto-Encoder-Based Multi-view Clustering. *Data Sci. Eng.* 6, 3 (2021), 323–338.

[10] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 178.

[11] Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. 2015. Multi-view subspace clustering. In *IEEE International Conference on Computer Vision*. 4238–4246.

[12] Junwei Han, Kun Song, Feiping Nie, and Xuelong Li. 2017. Bilateral k-Means Algorithm for Fast Co-Clustering. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 1969–1975.

[13] Chenping Hou, Feiping Nie, Hong Tao, and Dongyun Yi. 2017. Multi-View Unsupervised Feature Selection with Adaptive Similarity and View Weight. *IEEE Trans. Knowl. Data Eng.* 29, 9 (2017), 1998–2011.

[14] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. 2014. Smooth representation clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3834–3841.

[15] Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. 2022. Fast Multi-view Clustering via Ensembles: Towards Scalability, Superiority, and Simplicity. *CoRR* abs/2203.11572 (2022).

[16] Ling Huang, Hong-Yang Chao, and Chang-Dong Wang. 2019. Multi-View Intact Space Clustering. *Pattern Recognition* 86 (2019), 344–353.

[17] Ling Huang, Chang-Dong Wang, Hongyang Chao, and Philip S. Yu. 2020. MVStream: Multiview Data Stream Clustering. *IEEE Trans. Neural Networks Learn. Syst.* 31, 9 (2020), 3482–3496.

[18] Zhao Kang, Liangjian Wen, Wenyu Chen, and Zenglin Xu. 2019. Low-rank kernel learning for graph-based clustering. *Knowledge-Based Systems* 163 (2019), 510–517.

[19] 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.

[20] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. 2020. Large-Scale Multi-View Subspace Clustering in Linear Time. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*. 4412–4419.

[21] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. 2015. Large-Scale Multi-View Spectral Clustering via Bipartite Graph. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2750–2756.

[22] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1 (2013), 171–184.

[23] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, Chang Tang, Jianping Yin, Dinggang Shen, Huaimin Wang, and Wen Gao. 2019. Late Fusion Incomplete Multi-View Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 10 (2019), 2410–2423.

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

[25] Feiping Nie, Guohao Cai, Jing Li, and Xuelong Li. 2018. Auto-Weighted Multi-View Learning for Image Clustering and Semi-Supervised Classification. *IEEE Trans. Image Process.* 27, 3 (2018), 1501–1511.

[26] Feiping Nie, Jing Li, and Xuelong Li. 2016. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 1881–1887.

[27] Feiping Nie, Jing Li, and Xuelong Li. 2017. Self-weighted Multiview Clustering with Multiple Graphs. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 2564–2570.

[28] Feiping Nie, Shaojun Shi, and Xuelong Li. 2020. Auto-weighted multi-view co-clustering via fast matrix factorization. *Pattern Recognit.* 102 (2020), 107207.

[29] Mengjing Sun, Pei Zhang, Siwei Wang, Sihang Zhou, Wenxuan Tu, Xinwang Liu, En Zhu, and Changjian Wang. 2021. Scalable Multi-view Subspace Clustering

with Unified Anchors. In *ACM Multimedia Conference*. 3528–3536.

[30] Shiliang Sun, Liang Mao, Ziang Dong, and Lidan Wu. 2019. *Multiview Machine Learning*. Springer.

[31] Chang Tang, Xinzhong Zhu, Xinwang Liu, Miaomiao Li, Pichao Wang, Changqing Zhang, and Lizhe Wang. 2019. Learning a Joint Affinity Graph for Multiview Subspace Clustering. *IEEE Trans. Multimedia* 21, 7 (2019), 1724–1736.

[32] Chang-Dong Wang, Mansheng Chen, Ling Huang, Jian-Huang Lai, and Philip S. Yu. 2021. Smoothness Regularized Multiview Subspace Clustering With Kernel Learning. *IEEE Trans. Neural Networks Learn. Syst.* 32, 11 (2021), 5047–5060.

[33] Chang-Dong Wang, Jian-Huang Lai, and Philip S. Yu. 2016. Multi-View Clustering Based on Belief Propagation. *IEEE Trans. Knowl. Data Eng.* 28, 4 (2016), 1007–1021.

[34] Hua Wang, Feiping Nie, Heng Huang, and Fillia Makedon. 2011. Fast Nonnegative Matrix Tri-Factorization for Large-Scale Data Co-Clustering. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. 1553–1558.

[35] Hao Wang, Yan Yang, and Bing Liu. 2020. GMC: Graph-Based Multi-View Clustering. *IEEE Trans. Knowl. Data Eng.* 32, 6 (2020), 1116–1129.

[36] Jing Wang, Feng Tian, Hongchuan Yu, Chang Hong Liu, Kun Zhan, and Xiao Wang. 2018. Diverse Non-Negative Matrix Factorization for Multiview Data Representation. *IEEE Trans. Cybern.* 48, 9 (2018), 2620–2632.

[37] 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*. 3778–3784.

[38] Siwei Wang, Xinwang Liu, Xinzhong Zhu, Pei Zhang, Yi Zhang, Feng Gao, and En Zhu. 2022. Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance. *IEEE Trans. Image Process.* 31 (2022), 556–568.

[39] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li. 2017. Exclusivity-Consistency Regularized Multi-view Subspace Clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.

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

[41] Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji. 2013. Constrained Clustering and Its Application to Face Clustering in Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3507–3514.

[42] Yuan Xie, Dacheng Tao, Wensheng Zhang, Yan Liu, Lei Zhang, and Yanyun Qu. 2018. On Unifying Multi-view Self-Representations for Clustering by Tensor Multi-rank Minimization. *Int. J. Comput. Vis.* 126, 11 (2018), 1157–1179.

[43] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A Survey on Multi-view Learning. *CoRR* abs/1304.5634 (2013). arXiv:1304.5634

[44] Ben Yang, Xuetao Zhang, Feiping Nie, Fei Wang, Weizhong Yu, and Rong Wang. 2021. Fast Multi-View Clustering via Nonnegative and Orthogonal Factorization. *IEEE Trans. Image Process.* 30 (2021), 2575–2586.

[45] Hong Yu, Jia Tang, Guoyin Wang, and Xinbo Gao. 2021. A Novel Multi-View Clustering Method for Unknown Mapping Relationships Between Cross-View Samples. In *The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 2075–2083.

[46] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. 2020. Generalized Latent Multi-View Subspace Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 1 (2020), 86–99.

[47] Changqing Zhang, Huazhu Fu, Si Liu, Guangcan Liu, and Xiaochun Cao. 2015. Low-Rank Tensor Constrained Multiview Subspace Clustering. In *IEEE International Conference on Computer Vision*. 1582–1590.

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