Metric Multi-View Graph Clustering

Yuze Tan, Yixi Liu, Hongjie Wu, Jiancheng Lv, Shudong Huang*

College of Computer Science, Sichuan University, Chengdu 610065, China
{tanyuze, liuyixi}@stu.scu.edu.cn, wuhongjie0818@gmail.com {lvjiancheng, huangsd}@scu.edu.cn

Abstract

Graph-based methods have hitherto been used to pursue coherent patterns of data due to their ease of implementation and efficiency. These methods have been increasingly applied in multi-view learning and achieved promising performance in various clustering tasks. However, despite their noticeable empirical success, existing graph-based multi-view clustering methods may still suffer the suboptimal solution considering that multi-view data can be very complicated in raw feature space. Moreover, existing methods usually adopt the similarity metric by an ad hoc approach, which largely simplifies the relationship among real-world data and results in an inaccurate output. To address these issues, we propose to seamlessly integrate metric learning and graph learning for multi-view clustering. Specifically, we employ a useful metric to depict the inherent structure with linearity-aware of affinity graph representation learned based on the self-expressiveness property. Furthermore, instead of directly utilizing the raw features, we prefer to recover a smooth representation such that the geometric structure of the original data can be retained. We model the above concerns into a unified learning framework, and hence complement each learning subtask in a mutual reinforcement manner. The empirical studies corroborate our theoretical findings and demonstrate that the proposed method is able to boost the multi-view clustering performance.

Introduction

Learning the underlying similarity relationships contributes to capturing the inherent information of data and thus further facilitates the performance of downstream tasks. Considering the data points typically lying in latent subspaces, the subspace clustering technique plays an essential role to group data with respect to their underlying subspaces. These methods first exploit a similarity graph to represent the relationship of data by means of the self-expressiveness property and then perform the spectral clustering on the obtained graph to achieve the final clustering result (Ma et al. 2020). These methods are essentially graph-based methods, among which Sparse Subspace Clustering (SSC) (Elhamifar and Vidal 2013), Least Squares Regression (LSR) (Lu et al. 2012) and Low-Rank Representation (LRR) (Liu et al. 2012) are frequently employed in various practical fields of machine learning.

It is widely known that real-world data generally contain multiple views that are collected from multiple channels (Bickel and Scheffer 2004; Gao et al. 2015; Huang, Kang, and Xu 2020). For instance, a face image might consist of various features such as ambient light and a person's emotion, where one type of feature corresponds to a distinct view of data. Unlike traditional single-view algorithms, multi-view subspace clustering methods can precisely reshape the correlation between data samples. Assuming that a comprehensive latent representation can be generated from multi-view data, (Zhang et al. 2017) designed a multi-view subspace clustering method by searching the underlying latent representation from multiple views. To better fit the real-life data, the work (Luo et al. 2018) exploited one common consistent structure and a group of view-specific representations simultaneously to construct subspace representations. In (Huang et al. 2021), the diversity of multi-view data is redefined to represent a much broader concept than the noises and a holistic design of the clustering method is adopted by simultaneously considering the diverse parts and consistent pure components. (Zhang et al. 2020) introduced a one-step multi-view subspace clustering algorithm by integrating the steps of similarity learning, multi-view information fusion, and final clustering, which is distinguished from existing methods with isolated learning. To reduce the computational complexity of multi-view clustering, (Liu et al. 2022) employed the optimal anchor graph to directly output the clustering labels and incorporated the processes of graph construction and anchor learning into an integrated model to facilitate subsequent clustering tasks.

Although the aforementioned multi-view clustering algorithms showcase prominent performance, there are still two major drawbacks. For one thing, they assume that each sample can be well represented from the original data space, which may yield insufficient representation as real-world data generally consist of redundancy and are easily corrupted. Therefore, it is elusive to recover a separable representation to facilitate the subsequent clustering module. For another, blindly utilizing the Euclidean distance to measure the similarity between samples ignores the inherent correlation between samples to a large extent and thus fails to explore the underlying data distribution accurately. In this

*Corresponding author.

paper, we propose to seamlessly integrate metric learning and graph learning for multi-view clustering. We first recover a smooth representation by exploring the geometric structure of data. Then we employ a useful metric to depict the inherent structure with linearity-aware of the affinity graph obtained based on the smooth representation. We model the above concerns into a unified learning framework and hence complement each learning subtask in a mutual reinforcement manner. Our empirical experiments demonstrate the effectiveness and the superiority of the proposed method. The main contributions of our work are outlined as follows:

- We propose to recover a separable representation and better fit the inherent structure with the help of smooth representation learning and linearity-aware metric module respectively. We model the above concerns into a unified learning framework and hence lead to a novel model termed metric multi-view graph clustering.
- An efficient algorithm is introduced to solve the optimization problem. By leveraging the subtasks of smooth representation learning, multiple similarity graphs recovering, and the consensus graph fusing in our joint model, each subtask is alternately boosted towards an overall optimal solution.
- Extensive experiments on benchmark datasets are conducted to demonstrate the superiority of our model, compared with other state-of-the-arts.

Notations. In this paper, we utilize the normal italic symbols (e.g, z), boldface lowercase symbols (e.g, $\mathbf{z}$), and bold capital symbols (e.g, $\mathbf{Z}$) to represent scalars, vectors, and matrixes, respectively. $\mathbf{1}$ represents a column vector with all its elements equal to 1. $\mathbf{I}$ is an identity matrix. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

Related Work

It is well-known that subspace clustering methods aim to exploit underlying subspaces in the form of the affinity graph (Lu et al. 2018). In general, subspace learning is based on the self-expression of data samples, i.e., each sample can be reconstructed by a linear combination of the others in a union of subspaces (Elhamifar and Vidal 2013; Huang et al. 2019). Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n\times d}$, where n is the number of instances and d denotes the feature dimension. According to the self-expressiveness property, we have

$$\begin{aligned} &\mathbf{x}_i = \textstyle\sum_j \mathbf{x}_j z_{ij} \\ &\text{s.t.} \quad \forall i, z_{ij} \geq 0, \end{aligned} \tag{1}$$

where the combination coefficient z_{ij} denotes the similarity between the original data sample x_i and x_j. Accordingly, one can recover the coefficient matrix $\mathbf{Z} = [z_{ij}] \in \mathbb{R}^{n\times n}$ by solving

$$\begin{aligned} &\min\nolimits_{\mathbf{Z}} \textstyle\sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_j \mathbf{x}_j z_{ij} \right\|_F^2 \\ &\text{s.t.} \quad \forall i, z_{ij} \geq 0, \end{aligned} \tag{2}$$

where $\mathbf{Z}$ can also be treated as a similarity graph.

According to Eq. (2), the subspace clustering model can be formulated as

$$\begin{aligned} &\min\nolimits_{\mathbf{Z}} \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 + \phi \|\mathbf{Z}\|_F^2 \\ &\text{s.t.} \quad \mathbf{Z} \geq 0, \end{aligned} \tag{3}$$

where ϕ is a parameter to balance the corresponding regularization term. Based on Eq. (3), plenty of subspace clustering algorithms have been studied up to now (You, Robinson, and Vidal 2016; Yang et al. 2020).

It is clear that Eq. (3) can be extended to the multi-view domain. Given a multi-view data with m views $\left\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)} \mid \mathbf{X}^{(v)} \in \mathbb{R}^{n\times d_v}\right\}$ where d_v is the feature number in the v-th view, the multi-view version of Eq. (3) is

$$\begin{aligned} &\min\nolimits_{\mathbf{Z}^{(v)}} \textstyle\sum_{v=1}^{m} \left\|\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)}\right\|_F^2 + \phi \left\|\mathbf{Z}^{(v)}\right\|_F^2 \\ &\text{s.t.} \quad \mathbf{Z}^{(v)} \geq 0, \end{aligned} \tag{4}$$

where $\mathbf{Z}^{(v)}$ represents the affinity matrix of the v-th view.

The Proposed Method

Considering that the raw data might not be separable into subspaces due to the natural existence of redundancy and corruption, we propose to recover a smooth representation for each view, inspired by the graph filtering technique (Shuman et al. 2013). It is based on the fact that a signal is smooth by nature and associated adjacent nodes often share similar feature values. Since the smoother signals are often equipped with lower frequencies, retaining the low-frequency components is much preferred rather than high-frequency ones as they represent the smoothly changing structure (e.g. background) and the rapidly changing details (e.g. corruptions) respectively. Hence we employ a low-pass filter to obtain a smooth representation for each view. By taking each column of $\mathbf{X}^{(v)}$ as a graph signal, the formulation of k-order graph filtering on $\mathbf{X}^{(v)}$ can be designed as

$$\bar{\mathbf{X}}^{(v)} = \left(\mathbf{I} - \frac{\mathbf{L}^{(v)}}{2}\right)^k \mathbf{X}^{(v)}, \tag{5}$$

where $\bar{\mathbf{X}}$ represents a smoothed representation, $\mathbf{L}^{(v)}$ is the normalized graph Laplacian and k is a positive integer to consider the k-hop neighborhood relations (Zhang et al. 2019).

It is common knowledge that the core of clustering is to measure the similarity and dissimilarity among the given samples. Therefore, what kind of metric should be utilized to measure the distance between two data points is critical for the clustering task. As shown in Figure 1, Shiba inu 1 (S1) is closer to the golden retriever 1 (G1) than Shiba inu 3 (S3) under the metric of Euclidean distance, i.e., $d1 > d2$. However, it is obvious that both S1 and S3 belong to the same kind of dog, while S1 and G1 are not. In this paper, we employ a useful metric to depict the inherent similarity relationship with linearity-aware of the affinity graph. Here we first introduce the Pearson Correlation Coefficient (Benesty et al. 2009; Schober, Boer, and Schwarte 2018; Xu et al. 2022) as follows:

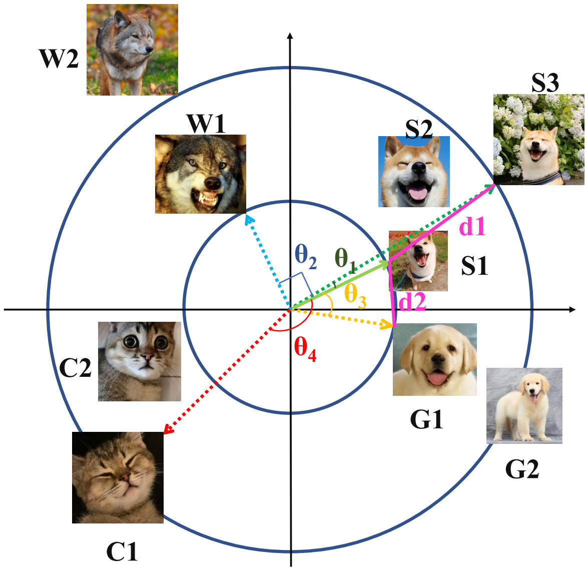


Figure 1: According to the picture, $d2 < d1$ shows that S1 is closer to G1 than to S3 under the Euclidean distance metric. However, it is obvious that S1 and S3 are both Shiba inu, while G3 is a golden retriever. On the contrary, under our proposed metric, $\theta_1 < \theta_3$ leads to $\mathcal{PCC}(\mathbf{S1}, \mathbf{S3}) > \mathcal{PCC}(\mathbf{S1}, \mathbf{G1})$. Therefore, S1 accurately owns a larger similarity with S3 than with G1.

Definition 1 ***(Pearson Correlation Coefficient)*** *Assumed a data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, where m stands for the feature dimension and n represents the number of instances. Choosing two arbitrary samples $\mathbf{x}_i, \mathbf{x}_j$ from $\mathbf{X}$, making sure $\mathbf{x}_i - \bar{x}_i, \mathbf{x}_j - \bar{x}_j \neq \mathbf{0}$, where $\bar{x}_i = \frac{1}{m}\sum_{k=1}^{m} x_i^k$. The Pearson Correlation Coefficient between them is formally defined as:*

$$\mathcal{PCC}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \frac{\left(\mathbf{x}_i - \bar{x}_i\right)^\top \left(\mathbf{x}_j - \bar{x}_j\right)}{\left\|\mathbf{x}_i - \bar{x}_i\right\|_2 \left\|\mathbf{x}_j - \bar{x}_j\right\|_2}. \tag{6}$$

The boundedness of Eq. (6) is given below.

Theorem 1 ***(Boundedness of the Coefficient)*** *As for two arbitrary instances $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$, one of the most fundamental properties is that:*

$$-1 \leq \mathcal{PCC}\left(\mathbf{x}_i, \mathbf{x}_j\right) \leq 1. \tag{7}$$

Proof 1 *Focusing on the formulation of $\mathcal{PCC}$ and without loss of generality, we reformulate Eq. (6) as:*

$$\mathcal{PCC} = \frac{\left(\mathbf{x}_i - \tilde{x}_i\right)^\top \left(\lambda\mathbf{x}_i - \lambda\tilde{x}_i\right)}{\left\|\mathbf{x}_i - \tilde{x}_i\right\|_2 \left\|\lambda\mathbf{x}_i - \lambda\tilde{x}_i\right\|_2} \tag{8}$$

where $\lambda\mathbf{x}_i$ equals $\mathbf{x}_j$ to depict the linear relationship between these two samples. And by simple algebra, we have:

$$\mathcal{PCC} = \frac{\lambda\left(\mathbf{x}_i - \tilde{x}_i\right)^\top \left(\mathbf{x}_i - \tilde{x}_i\right)}{|\lambda|\left(\left\|\mathbf{x}_i - \tilde{x}_i\right\|_2^2\right)} = \frac{\lambda}{|\lambda|} \tag{9}$$

Therefore, the range of $\mathcal{PCC}$ varying from -1 to 1 is proven.

Based on the Pearson Correlation Coefficient($\mathcal{PCC}$) in Eq. (6), we take a step forward that considers the linear relationship as a metric to evaluate the similarity among samples. The linearity-aware metric is explained in Definition 2.

Definition 2 ***Linearity-Aware Metric*** *According to theorem 1, $\mathcal{PCC}$ succeed in depicting the linear relationship between instances. In light of this, to pursue a novel metric that is suitable for the following optimization framework, we have:*

$$\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right) = 1 - \mathcal{PCC}. \tag{10}$$

In this way, we establish a much more discriminating model and we are now able to separate those samples which are closer under the Euclidean distance metric but do not have much linear relation. Therefore, we will obtain a more accurate and linear distinguishable outcome. The illustration of the improvement obtained by our method is shown in Figure 1. The conditions for the linearity-aware metric are:

- $\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right) = 2$. It suggests that x_i and x_j are completely negative correlated.
- $1 < \mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right) < 2$. It suggests that there is a ”certain degree” of negative linear relationship between x_i and x_j.
- $\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right) = 1$. x_i and x_j are not correlated.
- $0 < \mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right) < 1$. It suggests that there is a certain degree of positive linear relationship between x_i and x_j.
- $\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right) = 0$. x_i and x_j are completely positive correlated.

Taking Figure 1 as an example, θ_2 between S1 and W1 is 90 degrees, which means they are not correlated. θ_4 between S1 and C1 is larger than 90 degrees but smaller than 180 degrees, which means they have a certain degrees of negative correlation. And θ_3 between S1 and G1 is smaller than 90 degrees. But θ_1 between S1 and S3 is smaller than θ_3. Therefore, S1 and S3 have a much stronger correlation. In this regard, this showcases the feasibility of the linearity-aware metric. Intuitively, we learn the inherent similarity graph $\mathbf{S}^{(v)}$ for each view with linearity-aware as follows

$$\begin{gathered}\min_{\mathbf{S}} \sum_{v=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathcal{L}^2\left(\mathbf{z}_i^{(v)}, \mathbf{z}_j^{(v)}\right) s_{ij}^{(v)} + \gamma\|\mathbf{S}^{(v)}\|_F^2 \\ \text{s.t. } \mathbf{s}_i^{v\top}\mathbf{1} = 1, 0 \leq s_{ij}^{(v)} \leq 1,\end{gathered} \tag{11}$$

where γ represents the trade-off parameter.

Combining Eq. (4), Eq. (5) and Eq. (11), we arrive at

$$\begin{gathered}\min_{\mathbf{Z}^{(v)},\mathbf{S}^{(v)}} \mathcal{L}_1\left(\bar{\mathbf{X}}^{(v)}; \mathbf{Z}^{(v)}, \mathbf{S}^{(v)}\right) = \\ \sum_{v=1}^{m}\left\|\bar{\mathbf{X}}^{(v)} - \bar{\mathbf{X}}^{(v)}\mathbf{Z}^{(v)}\right\|_F^2 + \phi\left\|\mathbf{Z}^{(v)}\right\|_F^2 \\ + \sum_{v=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathcal{L}^2\left(\mathbf{z}_i^{(v)}, \mathbf{z}_j^{(v)}\right) s_{ij}^{(v)} + \gamma\|\mathbf{S}^{(v)}\|_F^2 \\ \text{s.t. } \mathbf{s}_i^{v\top}\mathbf{1} = 1, 0 \leq s_{ij}^{(v)} \leq 1,\end{gathered} \tag{12}$$

One of the most critical procedures in multi-view clustering is to effectively integrate information from different views. In light of (Nie et al. 2017), we utilize an adaptive weight strategy to meet the consistency among all views:

$$\min_{\mathbf{C}} \mathcal{L}_2\left(\mathbf{S}^{(v)}; \mathbf{C}\right) = \sum_{v=1}^{m} w^{(v)}\left\|\mathbf{C} - \mathbf{S}^{(v)}\right\|_F^2 \tag{13}$$

where $\mathbf{C}$ is the consensus graph, and $w^{(v)}$ is a self-tuned parameter used to weigh the importance of $\mathbf{S}^{(v)}$ which can be defined as

$$w^{(v)} = \frac{1}{2\left\|\mathbf{C} - \mathbf{S}^{(v)}\right\|_F}. \tag{14}$$

Note that Eq. (14) is essentially an inverse distance weighting.

With the coefficient matrix and similarity graphs learned in Eq. (11), and the consensus graph achieved in Eq. (13), our model can be finally modeled as

$$\begin{aligned} \min\nolimits_{\mathbf{Z}^{(v)},\mathbf{S}^{(v)},\mathbf{C}} & \underbrace{\mathcal{L}_1\left(\bar{\mathbf{X}}^{(v)}; \mathbf{Z}^{(v)}, \mathbf{S}^{(v)}\right)}_{\textit{graph learning}} + \underbrace{\mathcal{L}_2\left(\mathbf{S}^{(v)}; \mathbf{C}\right)}_{\textit{graph fusion}} \\ \text{s.t.} \quad & \mathbf{Z}^{(v)} \geq 0, \mathbf{s}_i^{(v)\top}\mathbf{1} = 1, 0 \leq s_{ij}^{(v)} \leq 1, \\ & \mathbf{1}^\top \mathbf{c}_i = 1, 0 \leq c_{ij} \leq 1. \end{aligned} \tag{15}$$

Optimization

Since Eq. (15) is not jointly convex in all variables, we propose to solve this non-convex problem by adapting an alternative optimization algorithm.

update $\mathbf{Z}^{(v)}$: To obtain the coefficient matrix $\mathbf{Z}^{(v)}$ of each view, we need to solve

$$\begin{aligned} \min\nolimits_{\mathbf{Z}^v} \sum\nolimits_{v=1}^m \left\|\bar{\mathbf{X}}^{(v)} - \bar{\mathbf{X}}^{(v)}\mathbf{Z}^{(v)}\right\|_F^2 + \phi \left\|\mathbf{Z}^{(v)}\right\|_F^2 \\ \text{s.t.} \quad \mathbf{Z}^{(v)} \geq 0, \end{aligned} \tag{16}$$

since Eq. (16) is independent for each v, we can solve it separately

$$\begin{aligned} \min\nolimits_{\mathbf{Z}^v} \left\|\bar{\mathbf{X}}^{(v)} - \bar{\mathbf{X}}^{(v)}\mathbf{Z}^{(v)}\right\|_F^2 + \phi \left\|\mathbf{Z}^{(v)}\right\|_F^2 \\ \text{s.t.} \quad \mathbf{Z}^{(v)} \geq 0, \end{aligned} \tag{17}$$

Taking the derivative of Eq. (17) and setting it to zero, we then have

$$\mathbf{Z}^{(v)} = \max\left(\left(\bar{\mathbf{X}}^{(v)\top}\bar{\mathbf{X}}^{(v)} + \phi\mathbf{I}\right)^{-1}\bar{\mathbf{X}}^{(v)\top}\bar{\mathbf{X}}^{(v)}, 0\right). \tag{18}$$

update $\mathbf{S}^{(v)}$: With other variables fixed, the corresponding optimization problem of each view becomes

$$\begin{aligned} \min\nolimits_{\mathbf{S}^{(v)}} \sum\nolimits_{i=1}^n \sum\nolimits_{j=1}^n \mathcal{L}^2\left(\mathbf{z}_i^{(v)}, \mathbf{z}_j^{(v)}\right) s_{ij}^{(v)} + \\ \gamma\|\mathbf{S}^{(v)}\|_F^2 + w^{(v)}\left\|\mathbf{C} - \mathbf{S}^{(v)}\right\|_F^2. \end{aligned} \tag{19}$$

For the simplicity of calculation, we denote $\mathcal{L}^2\left(\mathbf{z}_i^{(v)}, \mathbf{z}_j^{(v)}\right)$ as $\mathbf{d}_{ij}^{(v)}$, thus Eq. (19) can be written in a convenient form

$$\begin{aligned} \mathbf{s}_{i(t+1)}^{(v)} = \underset{\mathbf{s}_{i(t)}^{(v)}}{\arg\min} \left\|\mathbf{s}_{i(t)}^{(v)} + \frac{\left(\mathbf{d}_i^{(v)} - 2w^{(v)}\mathbf{c}_i\right)}{2\left(\gamma + w^{(v)}\right)}\right\|_2^2 \\ \text{s.t. } \mathbf{s}_i^{(v)\top}\mathbf{1} = 1, 0 \leq s_{ij} \leq 1, \end{aligned} \tag{20}$$

Algorithm 1: Algorithm for Metric Multi-view Subspace Clustering

Input: Feature matrix of m views $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(m)}\}$; the number of clusters p; balance parameters ϕ and γ.
Initialize: $w^{(v)} = \frac{1}{m}$, $\mathbf{S} = \mathbf{I}$. Construct the initial subspace $\mathbf{Z}^{(v)}$ of $\mathbf{X}^{(v)}$ by solving Eq. (4).
Output: The consensus graph $\mathbf{C}$ and the final clustering result.

```
repeat
    Update $\bar{\mathbf{X}}^{(v)}$ according to Eq. (5), where $\mathbf{L}^{(v)}$ is the normalized graph Laplacian of $\mathbf{S}^{(v)}$.
    Update the coefficient matrix $\mathbf{Z}^{(v)}$ according to Eq. (18).
    Update the similarity graph $\mathbf{S}^{(v)}$ by solving Eq. (21).
    Update consensus similarity graph $\mathbf{C}$ by solving Eq.(25).
    Update the self-tuned weight parameter $w^{(v)}$ according to Eq. (14).
until converge
Perform the standard spectral clustering on $\mathbf{C}$ to obtain the final clustering results.
```

where $\mathbf{s}_{i(t+1)}^{(v)}$ means $\mathbf{s}_i^{(v)}$ in the $(t+1)$-th iteration. Since ϕ, $w^{(v)}$, $\mathbf{d}_i^v$ and $\mathbf{c}_i$ are constant values when updating $\mathbf{S}^{(v)}$, we further simplify Eq. (20) by letting $\lambda = \left(\gamma + w^{(v)}\right)$ and $\mathbf{h}_i^{(v)} = \left(\mathbf{d}_i^{(v)} - 2w^{(v)}\mathbf{c}_i\right)$. In this way, Eq. (20) can be written as

$$\begin{aligned} \mathbf{s}_{i(t+1)}^{(v)} = \underset{\mathbf{s}_{i(t)}^{(v)}}{\arg\min} \left\|\mathbf{s}_{i(t)}^{(v)} + \frac{1}{2\lambda}\mathbf{h}_i\right\|_2^2 \\ \text{s.t. } \mathbf{s}_i^{(v)\top}\mathbf{1} = 1, 0 \leq s_{ij} \leq 1. \end{aligned} \tag{21}$$

Eq. (21) can be solved by off-the-shelf algorithm proposed in (Duchi et al. 2008; Huang, Nie, and Huang 2015).

update $\mathbf{C}$: While other variables are fixed, Eq. (15) is equivalent to

$$\begin{aligned} \min\nolimits_{\mathbf{C}} \sum\nolimits_{v=1}^m w^{(v)}\left\|\mathbf{C} - \mathbf{S}^{(v)}\right\|_F^2 \\ \text{s.t. } \mathbf{1}^\top\mathbf{c}_i = 1, \mathbf{C} \geq 0. \end{aligned} \tag{22}$$

Eq. (22) can be reformulated in an element-wise form

$$\begin{aligned} \min\nolimits_{\mathbf{C}} \sum\nolimits_{v=1}^m w^{(v)} \sum\nolimits_{i,j=1}^n \left(c_{ij} - s_{ij}^{(v)}\right)^2 \\ \text{s.t. } \mathbf{1}^\top\mathbf{c}_i = 1, \mathbf{C} \geq 0. \end{aligned} \tag{23}$$

It's self-evident that each view of Eq. (23) is independent from one another. Therefore we optimize it by each row i:

$$\begin{aligned} \min\nolimits_{\mathbf{c}_i} \sum\nolimits_{v=1}^m \sum\nolimits_{j=1}^n w^{(v)} \left(c_{ij} - s_{ij}^{(v)}\right)^2 \\ \text{s.t. } \mathbf{1}^\top\mathbf{c}_i = 1, \mathbf{C} \geq 0, \end{aligned} \tag{24}$$

where c_{ij} stands for the j-th element with respect to the row vector $\mathbf{c}_i$. Thus, Eq. (24) can be rewritten as

$$\begin{aligned} \min\nolimits_{\mathbf{c}_i} \sum\nolimits_{v=1}^m w^{(v)}\left\|\mathbf{c}_i - \mathbf{s}_i^{(v)}\right\|_2^2 \\ \text{s.t. } \mathbf{1}^\top\mathbf{c}_i = 1, \mathbf{C} \geq 0. \end{aligned} \tag{25}$$

Eq. (25) can be solved by utilizing the algorithm proposed in (Wang, Yang, and Liu 2019).

The detailed algorithm to solve Eq. (15) is summarized in Algorithm 1.

Time Complexity Analysis

Obviously, there are five steps that mainly resolve the time complexity of Algorithm 1. Recall that n, m, and p represent the number of data samples, views, and clusters, respectively. Let N_v be the number of nonzero entries of $\mathbf{S}^{(v)}$ (both $\mathbf{S}^{(v)}$ and $\mathbf{Z}^{(v)}$ are sparse). The time complexity of each step is given in Table 1.

Steps	Calculation	Complexity
Eq. (5)	graph filtering	$\mathcal{O}\left(N_v mp\right)$
Eq. (18)	update $\mathbf{Z}^{(v)}$	$\mathcal{O}\left(n^2 mp\right)$
Eq. (21)	update each column of $\mathbf{S}^{(v)}$	$\mathcal{O}\left(np\right)$
Eq. (25)	update each column of $\mathbf{C}$	$\mathcal{O}\left(nmp\right)$
Eq. (14)	$w^{(v)} = \frac{1}{2\left\|\mathbf{C}-\mathbf{S}^{(v)}\right\|_F}$	$\mathcal{O}\left(n^2 m\right)$
Total	$\approx \mathcal{O}\left(n^2\right)$	

Table 1: Time complexity of Algorithm 1.

In practical, we have $m \ll n$ and $p \ll n$, thus the overall time complexity is bounded by $\mathcal{O}\left(n^2\right)$.

Experiment

In this section, we demonstrate the efficiency and the superiority of our proposed method on several benchmark data sets, compared with other state-of-the-art multi-view clustering methods.

Experimental Setup

Intending to achieve a comprehensive evaluation, we compare our proposed method with several competitors: Multi-view Spectral Clustering with Co-training strategy(Co-train) (Kumar and Daumé 2011), Multi-view Spectral Clustering with Co-regularized strategy(Co-reg) (Kumar, Rai, and Daume 2011), Self-Weighted Multi-view Clustering(SwMC) (Nie et al. 2017), Generalized Latent Multi-View Subspace Clustering(LRMSC) (Zhang et al. 2018), Consistent and Specific Multi View Subspace Clustering(CSMSC) (Luo et al. 2018), Multi-view Consensus Graph Clustering(MCGC) (Zhan et al. 2019), Graph based Multi-view Clustering(GMC) (Wang, Yang, and Liu 2019), Consensus One-step Multi-view Subspace Clustering(COMVSC) (Zhang et al. 2020), Large-scale Multi-view Subspace Clustering in Linear Time(LMVSC) (Kang et al. 2020), multi-view clustering via Cross-view Graph Diffusion(CGD) (Tang et al. 2020), Multi-view Subspace Clustering via Co-training Robust Data Representation(CoMSC) (Liu et al. 2021a), One-pass Multi-view Clustering for Large-scale Data(OPMC) (Liu et al. 2021b), Scalable Multi-view Subspace Clustering with Unified Anchors(SMVSC) (Sun et al. 2021), Fast Parameter-free Multi-view Subspace Clustering with Consensus Anchor Guidance(FPMVS) (Wang et al. 2021),m Consensus Graph Learning for Multi-view Clustering(CGL) (Li et al. 2021).

Method	ACC	NMI	Purity	F-score
SC(AllFea)	65.51	61.88	67.59	59.42
Co-train	54.42	47.60	55.48	45.26
Co-reg	59.52	55.94	59.82	53.85
SWMC	37.19	42.88	40.62	49.28
LRMSC	54.93	57.25	62.64	55.07
CSMSC	62.11	64.01	70.61	59.19
GMC	35.19	41.21	37.63	49.13
COMVSC	59.21	50.53	62.75	53.23
LMVSC	**70.59**	65.71	70.98	62.73
CGD	41.84	47.02	45.20	50.88
CoMSC	60.63	53.34	61.07	50.05
OPMC	59.09	55.74	70.60	52.52
SMVSC	45.72	43.28	47.73	48.09
FPMVS	42.46	43.83	45.18	48.00
CGL	39.86	44.64	47.63	46.23
Ours	67.90	**68.69**	**74.24**	**68.77**

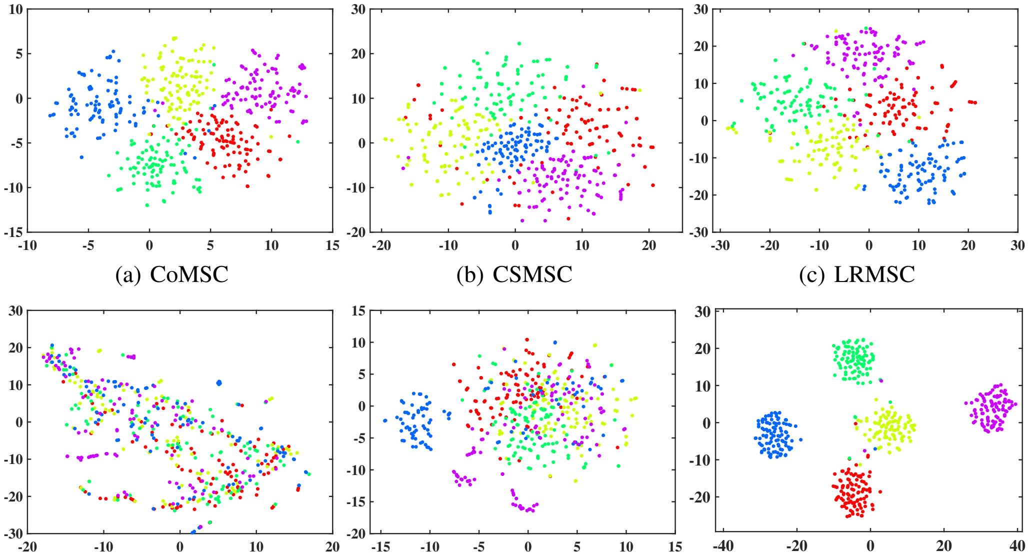
Table 2: The clustering results on HAR dataset (%)

Method	ACC	NMI	Purity	F-score
SC(AllFea)	55.14	45.04	57.19	42.13
Co-train	64.95	54.27	66.52	52.59
Co-reg	63.95	57.89	67.24	55.31
SWMC	72.38	71.78	77.14	66.19
LRMSC	71.81	62.42	74.14	60.08
CSMSC	80.43	71.36	80.43	70.06
MCGC	80.48	70.18	80.95	72.46
GMC	74.76	74.21	79.05	69.68
COMVSC	78.38	67.70	79.24	67.28
LMVSC	77.24	67.02	77.76	65.75
CGD	82.38	73.14	82.38	71.24
CoMSC	81.71	75.28	83.10	73.02
OPMC	84.33	72.95	84.43	72.47
SMVSC	81.43	70.18	81.43	69.36
FPMVS	78.57	66.84	78.57	68.36
CGL	82.86	75.78	82.86	73.30
Ours	**91.43**	**83.91**	**91.43**	**83.83**

Table 3: The clustering results on MSRC dataset (%)

We compare the above-mentioned methods with our proposed algorithm on several benchmark datasets: **HAR** is a Human Activity Recognition dataset which consists of 2941 samples and 6 classes. **Cora** is a popular dataset which is composed of 2708 instances and 7 categories, **MSRC** includes 240 images and 8 classes. Each class contains 30 images. Inspired by (Nie, Cai, and Li 2017), we extract 5 features from MSRC to comprehensively capture the intrinsic character within each image. **Newsgroups (NGs)** stands for one of the subsets of 20 Newsgroups datasets. It contains 500 newsgroups which are described from 3 different views. **ORL** is a well-known human face dataset that includes 400 images from 20 individuals. **Yale** contains 165 samples with 15 classes and is depicted from 3 different views.

To achieve the fairness of our experiment, the parameters of all compared algorithms are tuned to optimal values.



(a) CoMSC (b) CSMSC (c) LRMSC

(d) MCGC (e) SwMC (f) Ours

Figure 2: Visualization of the clustering results with t-SNE on NGs.

Method	ACC	NMI	Purity	F-score
SC(AllFea)	56.24	60.89	58.00	41.88
Co-train	53.94	57.90	55.52	38.68
Co-reg	56.36	60.30	57.64	41.43
SWMC	65.45	68.35	65.45	47.41
LRMSC	68.36	70.36	68.55	52.27
CSMSC	64.06	68.22	64.12	51.84
MCGC	67.27	68.92	67.27	48.33
GMC	65.45	67.36	66.06	48.01
COMVSC	71.27	71.70	71.33	55.01
LMVSC	61.45	61.96	68.24	40.57
CGD	47.88	55.70	49.15	37.61
CoMSC	63.58	65.17	70.36	44.35
OPMC	58.36	64.82	71.58	46.90
SMVSC	60.36	62.23	60.73	43.05
FPMVS	44.24	49.76	46.67	30.45
CGL	75.76	77.23	75.76	65.35
Ours	**78.00**	**80.72**	**78.00**	**69.17**

Table 4: The clustering results on Yale dataset (%)

Method	ACC	NMI	Purity	F-score
SC(AllFea)	63.27	79.82	67.38	52.33
Co-train	61.90	79.23	65.60	50.88
Co-reg	61.65	79.38	66.22	51.39
SWMC	70.75	83.31	76.75	43.33
LRMSC	79.63	90.49	83.42	73.99
CSMSC	77.57	89.20	80.77	71.69
MCGC	77.00	87.22	82.75	56.25
GMC	63.25	80.35	71.50	35.99
COMVSC	78.83	89.10	83.42	71.52
LMVSC	65.65	80.37	75.00	49.64
CGD	57.80	73.12	63.42	31.59
CoMSC	70.15	83.95	76.85	58.98
OPMC	61.90	78.73	68.83	49.39
SMVSC	58.28	76.07	61.75	43.96
FPMVS	55.45	73.72	59.27	41.02
CGL	86.00	92.31	87.48	80.61
Ours	**88.30**	**94.38**	**89.98**	**84.76**

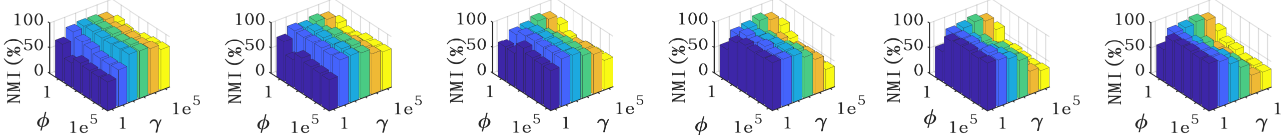
Table 5: The clustering results on ORL dataset (%)

Moreover, we run each method 10 times with the average results being recorded. The parameter setting of our method will be discussed in later subsection.

Results and Analysis

In order to adequately evaluate our method and compared algorithms, we adapt four widely used criteria including normalized mutual information (NMI), accuracy (ACC), Purity, and F-score. The clustering results are reported in Tables 2-7. We can arrive at a conclusion that our method is very effective and competitive, in view of the fact that the proposed method outperforms other competitors in the majority of cases. In detail, our method consistently obtains the best results in terms of NMI, Purity, and F-score on all datasets. While for ACC, the proposed method surpasses the compared methods except in one case on dataset HAR, where the second best performance is achieved.

We further visualize the clustering results of NGs with t-SNE (Van der Maaten and Hinton 2008). As shown in Figure 2, we can see CoMSC, CSMSC and LRMSC are able to divide the data samples into different clusters, yet there are no clear boundaries between them. What's worse, MCGC and SwMC even cannot find a clear clustering structure, which obviously fails to achieve a good performance. On the contrary, the clustering results obtained by our method showcase a more compact structure with clear boundaries,



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$ (d) $k = 4$ (e) $k = 5$ (f) $k = 6$

Figure 3: The influence of parameters γ, ϕ and k on ORL.

Method	ACC	NMI	Purity	F-score
SC(AllFea)	46.76	40.28	54.50	46.94
Co-train	89.62	78.75	90.54	83.28
Co-reg	84.76	70.32	85.46	74.53
LRMSC	96.42	88.90	96.42	92.97
CSMSC	98.40	94.61	98.40	96.82
GMC	98.20	93.92	98.20	96.43
COMVSC	38.32 2	26.44 2	42.26 2	41.44 5
LMVSC	44.96	24.66	57.62 1	34.35
CGD	95.20	85.51	95.20	90.68
CoMSC	98.40	94.61	98.40	96.81
OPMC	33.94	15.06	92.16	35.52
SMVSC	67.60	53.95	71.20	61.15
FPMVS	73.80	58.07	73.80	65.68
CGL	68.80	69.12	76.20	70.72
Ours	**99.20**	**97.21**	**99.20**	**98.40**

Table 6: The clustering results on NGs dataset (%)

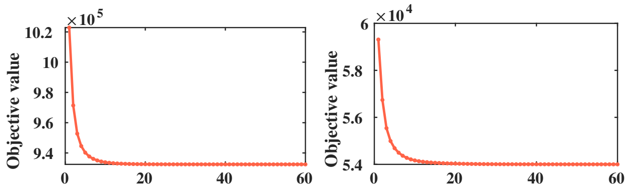
Method	ACC	NMI	Purity	F-score
SC(AllFea)	33.05	13.67	39.17	24.63
Co-train	50.95	31.32	56.45	36.64
Co-reg	35.41	20.05	43.37	27.44
LRMSC	48.88	32.89	54.40	36.27
CSMSC	54.39	45.68	61.57	47.48
LMVSC	43.40	25.33	49.14	31.59
CGD	43.94	21.02	46.42	35.86
CoMSC	64.16	46.49	68.67	49.46
OPMC	49.97	26.92	66.87	38.47
SMVSC	65.14	41.43	65.14	45.73
FPMVS	64.84	40.23	64.84	45.09
CGL	50.92	29.53	54.80	38.14
Ours	**71.83**	**50.80**	**73.19**	**57.95**

Table 7: The clustering results on Cora dataset (%)

which demonstrates the superiority of our method.

Sensitivity Analysis

With the aim of studying what impact different parameter settings will have on the clustering results, we vary three parameters: γ and ϕ in the ranges$[1, 1e1, 1e2, 1e3, 1e4, 1e5]$, and the filter order k in $[1, 2, 3, 4, 5, 6]$ respectively. Taking the ORL dataset as an example, we can see the clustering performance is relatively stable with respect to different k and ϕ, whereas it is a little sensitive to different γ, as shown in Figure 3. Considering that γ is the balance parameter for metric learning, we can come to a conclusion that the best metric differs from data to data. Generally, we can achieve superior performance when γ varies in the range [1e1,1e3].



(a) HAR (b) MSRC

Figure 4: Convergence curve of our algorithm.

Convergence Analysis

Since our model is essentially a nonconvex problem and is optimized with an iterative algorithm, it is critical to validate its convergence property. This section empirically showcases the convergence property and studies how fast the proposed algorithm can converge. As can be seen in Figure 4, our algorithm converges fast and the objective value varies little within a few iterations, which demonstrates the efficiency of the proposed algorithm. Note that despite the nonconvexity of our objective function described in Eq. (15), we can still search for the optimal solution for each variable and finally, the algorithm will converge to a local minima.

Conclusion

In this paper, we introduce a novel graph-based multi-view clustering method that measures the distance between instances from an innovative perspective. Unlike most existing methods considering Euclidean distance as the measurement of similarity between samples, our proposed method is capable of both the global structure of the input data and exploring the linear relationship between two local neighbors. Meanwhile, the graph filter that we used significantly boosts the robustness of the method. Finally, we integrate filter learning, subspace learning, and linearity-aware metric learning into one unified framework to achieve collaborative learning. Therefore, our method generates outstanding results on several authoritative benchmark datasets and is proven to outperform current state-of-the-art methods.

Acknowledgements

This work is supported by the Key Program of National Science Foundation of China (Grant No. 61836006), partially supported by the National Science Foundation of China under Grant 62106164, and the Sichuan Science and Technology Program under Grants 2021ZDZX0011 and 2022YFG0188.

References

Benesty, J.; Chen, J.; Huang, Y.; and Cohen, I. 2009. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*, 1–4. Springer.

Bickel, S.; and Scheffer, T. 2004. Multi-view clustering. In *Proceedings of the IEEE International Conference on Data Mining*, volume 4, 19–26.

Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; and Chandra, T. 2008. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning*, 272–279.

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.

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.

Huang, J.; Nie, F.; and Huang, H. 2015. A new simplex sparse learning model to measure data similarity for clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3569–3575.

Huang, S.; Kang, Z.; Tsang, I. W.; and Xu, Z. 2019. Auto-weighted multi-view clustering via kernelized graph learning. *Pattern Recognition*, 88: 174–184.

Huang, S.; Kang, Z.; and Xu, Z. 2020. Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognition*, 97: 1–11.

Huang, S.; Tsang, I. W.; Xu, Z.; Lv, J.; and Liu, Q. 2021. CDD: Multi-view Subspace Clustering via Cross-view Diversity Detection. In *Proceedings of the ACM International Conference on Multimedia*, 2308–2316.

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.

Kumar, A.; and Daumé, H. 2011. A co-training approach for multi-view spectral clustering. In *Proceedings of the International Conference on Machine Learning*, 393–400.

Kumar, A.; Rai, P.; and Daume, H. 2011. Co-regularized multi-view spectral clustering. *Advances in Neural Information Processing Systems*, 24: 1–9.

Li, Z.; Tang, C.; Liu, X.; Zheng, X.; Zhang, W.; and Zhu, E. 2021. Consensus graph learning for multi-view clustering. *IEEE Transactions on Multimedia*, 24: 2461–2472.

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*, 5177–5189.

Liu, J.; Liu, X.; Yang, Y.; Liu, L.; Wang, S.; Liang, W.; and Shi, J. 2021b. One-pass multi-view clustering for large-scale data. In *Proceedings of the IEEE International Conference on Computer Vision*, 12344–12353.

Liu, S.; Wang, S.; Zhang, P.; Xu, K.; Liu, X.; Zhang, C.; and Gao, F. 2022. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7576–7584.

Lu, C.; Feng, J.; Lin, Z.; Mei, T.; and Yan, S. 2018. Subspace clustering by block diagonal representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2): 487–501.

Lu, C.-Y.; Min, H.; Zhao, Z.-Q.; Zhu, L.; Huang, D.-S.; and Yan, S. 2012. Robust and efficient subspace segmentation via least squares regression. In *Proceedings of the European Conference on Computer Vision*, 347–360.

Luo, S.; Zhang, C.; Zhang, W.; and Cao, X. 2018. Consistent and specific multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 3730–3737.

Ma, Z.; Kang, Z.; Luo, G.; Tian, L.; and Chen, W. 2020. Towards clustering-friendly representations: subspace clustering via graph filtering. In *Proceedings of the ACM International Conference on Multimedia*, 3081–3089.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2408–2414.

Nie, F.; Li, J.; Li, X.; et al. 2017. Self-weighted Multiview Clustering with Multiple Graphs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2564–2570.

Schober, P.; Boer, C.; and Schwarte, L. A. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5): 1763–1768.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal Processing Magazine*, 30(3): 83–98.

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 ACM International Conference on Multimedia*, 3528–3536.

Tang, C.; Liu, X.; Zhu, X.; Zhu, E.; Luo, Z.; Wang, L.; and Gao, W. 2020. CGD: Multi-view clustering via cross-view graph diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5924–5931.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11): 2579–2605.

Wang, H.; Yang, Y.; and Liu, B. 2019. GMC: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6): 1116–1129.

Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2021. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31: 556–568.

Xu, Y.; Chen, S.; Li, J.; and Qian, J. 2022. Linearity-aware subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8770–8778.

Yang, J.; Liang, J.; Wang, K.; Rosin, P. L.; and Yang, M.-H. 2020. Subspace clustering via good neighbors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6): 1537–1544.

You, C.; Robinson, D.; and Vidal, R. 2016. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3918–3927.

Zhan, K.; Nie, F.; Wang, J.; and Yang, Y. 2019. Multiview consensus graph clustering. *IEEE Transactions on Image Processing*, 28(3): 1261–1270.

Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2018. Generalized latent multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1): 86–99.

Zhang, C.; Hu, Q.; Fu, H.; Zhu, P.; and Cao, X. 2017. Latent multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4279–4287.

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*, 4676–4689.

Zhang, X.; Liu, H.; Li, Q.; and Wu, X.-M. 2019. Attributed graph clustering via adaptive graph convolution. In *Proceedings of the International Conference on International Joint Conferences on Artificial Intelligence*, 4327–4333.