East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2023-314.140824

Vol. **16**, No. 1, pp. 20-44
February 2026

Multi-View Data Clustering via Dynamical Optimization of Consensus Laplacian Matrix

Senwen Zhan[1], Hao Jiang[1,*], Dong Shen[1] and Wai-Ki Ching[2]

[1]*School of Mathematics, Renmin University of China, Beijing 100872, China.*
[2]*Department of Mathematics, The University of Hong Kong, Hong Kong, SAR, China.*

Received 8 December 2023; Accepted (in revised version) 14 August 2024.

Abstract. Multi-view data analysis has gained increasing popularity, in particular multi-view spectral clustering has attracted much attention for its outstanding performance in mining heterogeneity information in multi-view data. However, most spectral clustering methods exhibit the following disadvantages: firstly, learning consensus representation directly from multi-view data that may contain noise renders a distorted description; secondly, the traditional two-step process may fall into a suboptimal solution. To overcome these disadvantages, a novel multi-view spectral clustering method is proposed by unifying the optimization of consensus Laplacian matrix and the learning and discretization of spectral embedding into one step. We consider that the optimal Laplacian matrix is in the neighborhood of view-specific Laplacian matrix, as the view-specific Laplacian matrix only contains partial information from multi-view data, resulting in certain deviation from the optimal Laplacian matrix. The consensus Laplacian matrix was obtained in a dynamic optimization way with the spectral rotation and embedding information simultaneously determined. Extensive experiments have been conducted to demonstrate the effectiveness and superiority of our proposed method.

AMS subject classifications: 65M10, 78A48

Key words: Multi-view, dynamical optimization, spectral clustering, spectral rotation, unified framework.

1. Introduction

Multi-view data can capture wealth of information from various perspectives. However, due to the diverse sources of data acquisition, the features of multi-view data are of redundancy, correlation, diversity [5]. A key challenge is how to integrate the information of each view [36]. Early fusion is the most widely used method in multi-view clustering [17,30,32]. It offers a unified consensus representation for the multi-view data, and in

*Corresponding author. *Email addresses:* zhansenwen@ruc.edu.cn (S.W. Zhan), jiangh@ruc.edu.cn (H. Jiang), dshen@ruc.edu.cn (D. Shen), wching@hku.hk (W. Ching)

the later stage uses a single-view clustering method to capture heterogeneity in the data. Late fusion [22,23] is an alternative way to integrate the multi-sources of data. Each view is clustered separately to generate multiple base clustering results, which are then combined in the final stage to produce a consensus clustering outcome.

Although the existing multi-view spectral clustering methods can achieve promising clustering results, it is still challenging to fully extract information embedded in the multi-view data due to the following reasons. First, most multi-view spectral clustering methods are realized in two independent steps [19]. While each step can achieve its own optimization goal, the two-step strategy always yields a suboptimal result rather than the global optimal one. Second, most multi-view spectral clustering methods directly learn a consensus representation from original multi-view data, which would render a distorted description for the data, because the data obtained from various sources are usually noisy and correlated [39, 40]. For example, some methods linearly combine the Laplacian matrices of all views to obtain a consensus Laplacian matrix for the final multi-view spectral clustering [18,42]. Third, existing methods usually treat each view as equally important, ignoring the importance of different views [15,33].

To effectively solve the above issues, we propose a novel one-step multi-view spectral clustering framework by combining the learning of consensus Laplacian matrix and the discretization of spectral embedding in a unified framework. For the Laplacian matrix of each view, we assume that it contains partial information of the whole multi-view data and regard it as a minor deviation of the optimal Laplacian matrix (as shown in Fig. 1). Our goal is to dynamically optimize the optimal Laplacian matrix provided that it is located in the neighborhood of view-specific Laplacian matrices. In addition, each view is adaptively weighted in search process to reduce the noise effect.

To demonstrate the effectiveness of our method, we conducted extensive experiments on a number of widely used real-world datasets, and used well-known metrics: ARI, NMI, ACC, F-score, Purity to evaluate clustering performances, visualized the aggregated embedding data using t-stochastic neighborhood embedding (t-SNE). The proposed model converges in a fast manner and shows extraordinary performance in multi-view data integration.

The contributions of this study are summarized as follows:

- We search for the optimal Laplacian matrix in the neighborhood of Laplacian matrix of each view, allowing that the information contained in the Laplacian matrix of each view is not adequately accurate, and may slightly deviate from the optimal Laplacian matrix. The optimal Laplacian matrix and low dimensional embedding information can be obtained through ingenious design of the optimization objective and framework.

- By putting the learning and discretization of spectral embedding into a unified framework, we obtain the clustering result of multi-view data directly in optimization process. This one-step strategy can avoid loss of information and obtain more reasonable clustering result.

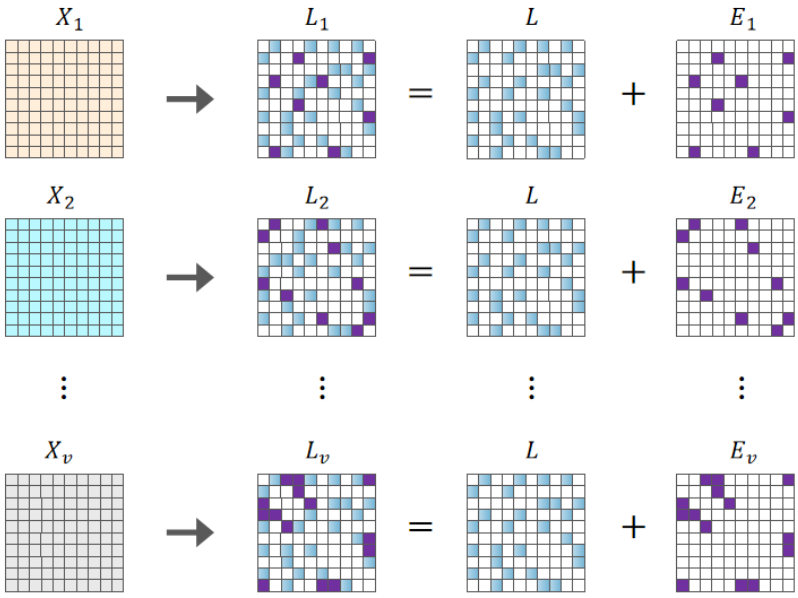


Figure 1: The relationship between the view-specific Laplacian matrix $L_i, i = 1, 2, \ldots, v$ and the optimal Laplacian matrix L modelling the relationship of multi-view data, where E_i represents the error matrix of the i-th view due to noise or incomplete information. Since view-specific Laplacian matrix contains only partial information about multi-view data, it can be considered as deviating from the optimal Laplacian matrix. We search for the optimal Laplacian matrix in the neighborhood of view-specific Laplacian matrix.

- We adaptively evaluate the importance of different views in optimization process. In particular, less weight is adaptively assigned to the noisy view and more weight is adaptively assigned to the informative view to make the learned consensus results more accurate.

The remainder of the paper is organized as follows. Section 2 provides related works in multi-view data clustering. In Section 3, we introduce the notations used in this paper and preliminary results of the spectral method. In Section 4, we describe our methodology in detail, derive the multi-view data analysis model and the corresponding optimization algorithm with theoretical analysis. A number of extensive experiments are conducted and analyzed in Section 5. Finally, we conclude this paper in Section 6.

2. Related Works

As a fundamental research topic in machine learning and data mining, data clustering aims to analyze the heterogeneity embedded in the data. Moreover, with the increasing wealth of information in multi-view data, researchers have moved beyond single-view models and developed many multi-view models [3, 4, 28]. Many attempts have been made on multi-view clustering, such as graph-context-aware refinement [21] and co-neighbor propagation. A new projection-based method [16] was proposed to mine high-order information

between views effectively. For multi-view spectral clustering, it is critical to effectively integrate the information of each view and avoid the information loss caused by the two-step process.

In the existing literature, most multi-view spectral clustering methods fall into early fusion category. For example, view specific Laplacian matrix was linearly combined [42], and the view-specific weights are fine tuned with spectral perturbation theory. The similarity matrix of each view is composed of consistent part and inconsistent part in [20], and the adjacency matrix of consistent graph should be close to the linear combination of consistent parts of each view while keeping the inconsistent part of each view sparse. A new multi-view clustering method based on bipartite graph was proposed [17], which integrated bipartite graphs of different views to form a consensus bipartite graph for clustering. Research has been conducted [37] using sparse subspace clustering to construct initial similarity matrices, and fine-grained similarity fusion was applied to obtain the integrated adjacency matrix for clustering. By considering clustering quality evaluation criterion and the view consistency loss [13], authors constructed a new consensus affinity matrix for multi-view spectral clustering, an auto-weighted mutual reinforcing approach [12] was proposed to simultaneously learn the similarity matrix of each view and consensus graph.

To deal with the suboptimal issue in traditional multi-view clustering pipelines, one-step frameworks [8] have been developed in the recent decades. JESER [27] was an initiative in exploring one-step framework for single view data. It performed spectral embedding and spectral rotation simultaneously. In [41], a unified framework integrating the learning of common affinity matrix and k-means was proposed for multi-view data. In [29], view-specific spectral embedding matrices and similarity graphs are linearly combined in a separate manner and jointly explored to generate the clustering results.

3. Notations and Preliminaries

3.1. Notations

We denote the matrix as uppercase letters, the vector as lowercase letters and scalars as normal italic letters. We also denote $X_1, X_2, \cdots, X_V$ as dataset of V views, where $X_i \in \mathbb{R}^{n \times d_i}$ represents the i-th view dataset, n is the number of data points and d_i is the dimension of features of i-th view. Further, we use X^T, tr(X), rank(X), X^{-1} to represent the transpose, trace, rank and inverse of matrix X, respectively.

3.2. Spectral clustering

Given a dataset containing n data points $\{x_1, x_2, \cdots, x_n\}$, spectral clustering represents them as nodes on graph G and obtain the corresponding weights to the edges between nodes according to their similarity. Generally, the similarity is described by a Gaussian kernel function. Spectral clustering achieves the purpose of clustering by reasonably dividing graph G into multiple subgraphs $G_1, G_2, \cdots, G_c$, where c is the number of clusters. Let

$W \in \mathbb{R}^{n\times n}$ be the adjacent matrix of graph G, W can be defined as follows:

$$w_{ij} = \begin{cases} \exp\left(-\dfrac{\|x_i - x_j\|^2}{2\sigma^2}\right), & \text{if vertices } (i,j) \text{ are connected,} \\ 0, & \text{otherwise.} \end{cases}$$

Spectral clustering uses the following formula to describe the compactness of two subgraphs G_i, G_j:

$$\text{cut}(G_i, G_j) = \sum_{t\in G_i, k\in G_j} w_{tk}.$$

Different graph cutting methods are generated according to the size evaluation criteria of subgraph. Here, we focus on the normalized cut, while details on RatioCut can be found in [7]. Let $D \in \mathbb{R}^{n\times n}$ be a diagonal matrix, where the i-th diagonal element satisfies $d_i = \sum_{i=1}^{n} w_{ij}$. We minimize the following index to achieve the most appropriate graph segmentation:

$$\text{Ncut}(G_1, G_2, \cdots, G_c) = \sum_{i=1}^{c} \frac{\text{cut}(G_i, \bar{G}_i)}{\text{vol}(G_i)}, \tag{3.1}$$

where $\bar{G}_i$ represents the complement of G_i in G, and $\text{vol}(G_i) = \sum_{i\in G_i} d_i$. To minimize Eq. (3.1), the cluster indicator matrix $F = [f_1, f_2, \cdots, f_c] \in \mathbb{R}^{n\times c}$ is defined as follows:

$$f_{ij} = \begin{cases} 1, & \text{if node } i \text{ belongs to the } j\text{-th cluster,} \\ 0, & \text{otherwise.} \end{cases}$$

We therefore obtain

$$\begin{aligned} \text{cut}(G_i, G_i) &= f_i^T W f_i, \\ \text{vol}(G_i) &= f_i^T D f_i. \end{aligned}$$

Let Laplacian matrix $L = D - W$. The Eq. (3.1) can be derived as follows:

$$\begin{aligned} \text{Ncut}(G_1, G_2, \cdots, G_c) &= \sum_{i=1}^{c} \frac{f_i^T D f_i - f_i^T W f_i}{f_i^T D f_i} = \sum_{i=1}^{c} \frac{f_i^T L f_i}{f_i^T D f_i} \\ &= \sum_{i=1}^{c} \left(f_i^T L f_i\right)\left(f_i^T D f_i\right)^{-1} \\ &= \text{tr}\left[(F^T L F)(F^T D F)^{-1}\right]. \end{aligned}$$

The matrix $\tilde{L} = D^{-1/2} L D^{-1/2} = I_n - D^{-1/2} W D^{-1/2}$ is called the normalized Laplacian. In the normalized cut method, we only need to resolve the following problem:

$$\begin{aligned} &\min_F \ \text{tr}(F^T \tilde{L} F), \\ &\text{s.t.} \quad F^T F = I_c. \end{aligned} \tag{3.2}$$

Because F is a discrete cluster indicator matrix, it is difficult to solve Eq. (3.2) directly. A widely used method is to relax the constraint of F such that it can take continuous values. The problem (3.2) takes form

$$\begin{aligned} &\min_{F\in\mathbb{R}^{n\times c}} \quad \mathrm{tr}(F^T\tilde{L}F), \\ &\text{s.t.} \quad F^TF = I_c. \end{aligned} \tag{3.3}$$

The solution of Eq. (3.3) is to take the first c eigenvectors of $\tilde{L}$ as the column vectors of spectral embedding matrix F.

3.3. Spectral rotation

The goal of spectral clustering is to divide graph into several subgraphs according to the weight between nodes, which is an NP-hard problem. Therefore, the constraint is often relaxed to obtain a continuous spectral embedding. This continuous result can be considered as a deviation from the real discrete result, and the common discretization method is to apply k-means algorithm to spectral embedding. However, the discrete result obtained by k-means algorithm is not necessarily appropriate, because there are many optimal solutions to the optimization problem of spectral clustering after constraint relaxation according to Theorem 3.1.

Theorem 3.1. *Let $\tilde{H}$ be the optimal solution of the optimization problem* (3.3)*, then $\tilde{H}R$ is also the optimal solution of this problem, where $R\in\mathbb{R}^{c\times c}$ is an orthogonal matrix.*

Proof. For any $H\in\mathbb{R}^{n\times c}$, we obtain

$$\mathrm{tr}(H^T\tilde{L}H) \geq \mathrm{tr}(\tilde{H}^T\tilde{L}\tilde{H}) = \mathrm{tr}(RR^T\tilde{H}^T\tilde{L}\tilde{H}) = \mathrm{tr}\left[(\tilde{H}R)^T\tilde{L}(\tilde{H}R)\right].$$

This shows that H and $\tilde{H}R$ are both optimal solutions. □

Given the existence of multiple optimal solutions, a natural question arises: which solution is most suitable for discretization? Spectral rotation can solve this problem by rotating continuous spectral embedding to search for appropriate discrete indication result. As shown in Fig. 2, even if the initial continuous spectral embedding is not the most appropriate, we can rotate it to be as close to the discrete result as possible. The optimization problem of spectral rotation [11] has the form

$$\begin{aligned} &\min_{H,R,G} \|HR-G\|_F^2 \\ &\text{s.t.} \quad H^TH = I_c, \quad R^TR = I_c, \quad G\in\mathbb{S}, \end{aligned}$$

where

$$\mathbb{S} = \left\{ G \mid G\in\{0,1\}^{n\times c}, \sum_{j=1}^{c} G_{ij} = 1 \right\}.$$

It is equivalent to the following problem:

$$\begin{aligned} &\min_{H,R,G} \|H-GR\|_F^2 \\ &\text{s.t.} \quad H^TH = I_c, \quad R^TR = I_c, \quad G\in\mathbb{S}. \end{aligned} \tag{3.4}$$

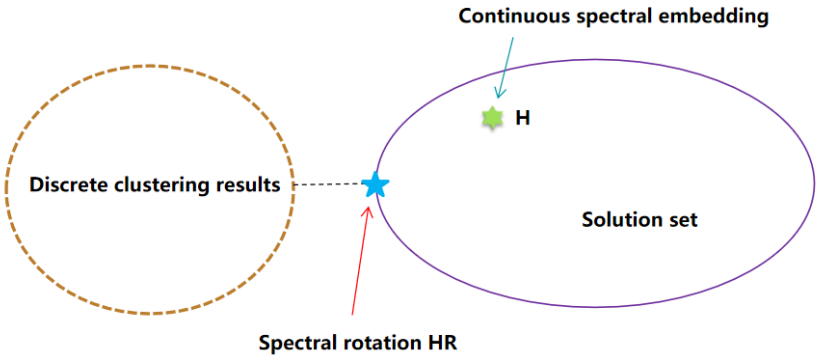


Figure 2: Method illustration of spectral rotation. The dotted brown line on the left represents the discrete clustering result we want to learn, solution set means the set of all optimal solutions. Continuous spectral embedding H finds the closest discrete cluster result through rotation R.

4. Methodology

In this section, we first search the consensus Laplacian matrix such that the learned Laplacian matrix is optimal through some restrictions. Further, we put the learning and discretization of spectral embedding into a unified framework to obtain the clustering result in one step. Last, we use an ingenious optimization framework to update this model.

4.1. Dynamical optimization of Laplacian matrix

For multi-view data of V views: $X = \{X_1, X_2, \cdots, X_V\}$, we could obtain Laplacian matrix of each view $L_1, L_2, \cdots, L_V$ by $L_i = I_n - D_i^{-1/2} W_i D_i^{-1/2}$ after data normalization. Our purpose is to find an optimal Laplacian matrix containing all multi-view information for spectral clustering, instead of linearly combining Laplacian matrices of all views $L = \sum_{i=1}^{V} \mu_i L_i$ [18, 34, 42]. We search this optimal Laplacian matrix L around $L_1, L_2, \cdots, L_V$.

Generally, each view contains noise or only part information of the whole multi-view data. Therefore, they have some deviations from the optimal Laplacian matrix — cf. Fig. 1. In other words, for i-th view, L_i is generated by the perturbation of L, i.e.

$$L_i = L + E_i,$$

where E_i is the error matrix of i-th view. Meanwhile, we also consider that L is near L_i, $1 \le i \le V$. Considering the diverse importance of different views, we apply nonidentical weights to different views and establish the following model:

$$\begin{aligned} \min_{\mu_i, L} \ & \sum_{i=1}^{V} \frac{1}{\mu_i} \|L - L_i\|_F^2 \\ \text{s.t.} \ & \|\mu\|_1 = 1, \quad \mu \ge 0. \end{aligned} \tag{4.1}$$

We use $1/\mu_i$ to model the weight of i-th view instead of μ_i, with the main purpose of avoiding trivial solution to Eq. (4.1) that only single view information would make contributions. According to [14], the learned optimal Laplacian matrix L should satisfy the following conditions:

Condition 4.1. L is a positive semi-definite matrix.

Condition 4.2. The eigenvalues of L are located in $[0, 2]$.

Our method can also be regarded as finding the optimal consensus graph, where the corresponding Laplacian matrix is denoted as L. Ideally, this graph should have exactly c connected components (c is the number of clusters), indicating that the data points in this graph have been divided into c clusters. This is reflected by Theorem 4.1 [31].

Theorem 4.1. *Let G be an undirected graph with nonnegative weights. Then, the multiplicity c of the eigenvalue zero of L is equal to the number of connected components in the graph.*

To obtain the optimal graph with clear clustering result, our model turns into the following form:

$$\begin{aligned} \min_{\mu_i, L} \ & \sum_{i=1}^{V} \frac{1}{\mu_i} \|L - L_i\|_F^2 \\ \text{s.t.} \quad & \|\mu\|_1 = 1, \quad \mu \geq 0, \quad \operatorname{rank}(L) = n - c. \end{aligned} \tag{4.2}$$

This restriction makes it particularly difficult to solve the optimization problem (4.2). To effectively solve the problem, we appropriately adjust Eq. (4.2).

According to Condition 4.1, the eigenvalues of L satisfy $\sigma_n \geq \sigma_{n-1} \geq \cdots \geq \sigma_1 \geq 0$. Therefore, if $\sum_{i=1}^{c} \sigma_i = 0$, the consensus graph is close to the optimal graph with a clear clustering structure. However, making $\sum_{i=1}^{c} \sigma_i = 0$ strictly hold does not reduce the complexity of Eq. (4.2). Thus, we replace $\sum_{i=1}^{c} \sigma_i = 0$ with $\sum_{i=1}^{c} \sigma_i \to 0$ to achieve the same goal and facilitate optimization. Then, according to Conditions 4.1, 4.2 and Theorem 4.1, we adjust Eq. (4.2) to the following form:

$$\begin{aligned} \min_{\mu_i, L} \ & \sum_{i=1}^{c} \sigma_i + \sum_{i=1}^{V} \frac{1}{\mu_i} \|L - L_i\|_F^2, \\ \text{s.t.} \quad & \|\mu\|_1 = 1, \quad \mu \geq 0, \quad L \succeq 0, \quad 0 \leq \sigma_i \leq 2. \end{aligned}$$

Theorem 4.2 (cf. Fan [6])**.**

$$\sum_{i=1}^{c} \sigma_i = \min_{H^T H = I_c} \operatorname{tr}(H^T L H).$$

Theorem 4.2 yields that the proposed model can be written as

$$\begin{aligned} \min_{\mu_i, L, H} \ & \operatorname{tr}(H^T L H) + \sum_{i=1}^{V} \frac{1}{\mu_i} \|L - L_i\|_F^2 \\ \text{s.t.} \quad & \|\mu\|_1 = 1, \quad \mu \geq 0, \quad H^T H = I_c, \quad L \succeq 0, \quad 0 \leq \sigma_i \leq 2. \end{aligned} \tag{4.3}$$

4.2. One-step multi-view spectral clustering

By solving Eq. (4.3), we obtain a continuous spectral embedding matrix H. However, the target of spectral clustering is to obtain a discrete clustering result. Spectral rotation is an efficient method to recover the best discrete approximation of eigenvectors. Therefore, after obtaining an optimal spectral embedding matrix H, our goal is to find a suitable orthogonal matrix R to rotate H to an appropriate discrete indicator matrix G.

To obtain the clustering result in one step, we put search of the optimal Laplacian matrix L and discretization of the spectral embedding H into a unified framework. In particular, combining the Eqs. (4.3) and (3.4), we rewrite the optimization problem as

$$\begin{aligned} \min_{\mu_i,L,H,R,G} \ & \operatorname{tr}(H^T L H) + \sum_{i=1}^{V} \frac{1}{\mu_i} \|L - L_i\|_F^2 + \beta \|H - GR\|_F^2 \\ \text{s.t.} \quad & \|\mu\|_1 = 1, \quad \mu \geq 0, \quad H^T H = I_c, \quad L \succeq 0, \\ & 0 \leq \sigma_i \leq 2, \quad R^T R = I_c, \quad G \in \mathbb{S}, \end{aligned} \tag{4.4}$$

where $\beta > 0$ is a trade-off parameter.

In summary, L_i denotes the Laplacian matrix of i-th view. We find an optimal Laplacian matrix L in the neighborhood of $L_i, i = 1, 2, \ldots, V$ and get the corresponding consensus spectral embedding H, where μ_i indicates the importance of i-th view. Then we obtain the clustering indicator matrix G by the rotation matrix R.

However, it is difficult to update L in Eq. (4.4) directly because we have little information about L. Recalling the definition of the normalized Laplacian matrix, i.e., $L = I_n - D^{-1/2} W D^{-1/2}$, we denote $D^{-1/2} W D^{-1/2} = P \Sigma P^T$ by orthogonal diagonalization. If the value of each element of $D^{-1/2} W D^{-1/2}$ is estimated, the Laplacian matrix L is then obtained. Eq. (4.4) can be modified as follows:

$$\begin{aligned} \min_{H,\mu_i,\Sigma,P,R,G} \ & \operatorname{tr}\left(H^T (I_n - P \Sigma P^T) H\right) + \sum_{i=1}^{V} \frac{1}{\mu_i} \|I_n - P \Sigma P^T - L_i\|_F^2 + \beta \|H - GR\|_F^2 \\ \text{s.t.} \quad & H^T H = I_c, \quad \|\mu\|_1 = 1, \quad \mu \geq 0, \\ & P^T P = I_n, \quad -1 \leq \lambda_i \leq 1, \quad R^T R = I_c, \quad G \in \mathbb{S}, \end{aligned} \tag{4.5}$$

where λ_i is the i-th diagonal element of Σ. In order to ensure that the learned optimal Laplacian matrix L satisfies Condition 4.2, we restrict $\lambda_i \in [-1, 1]$ according to Theorem 4.3 below.

Theorem 4.3. *If the eigenvalues of Σ satisfy $\lambda_i \in [-1, 1]$, then the range of eigenvalue σ_i of L is $[0, 2]$.*

Proof. Let τ_i represent the i-th eigenvalue of $P \Sigma P^T$. Noting that $L = I_n - P \Sigma P^T$, we obtain

$$\sigma_i = 1 - \tau_i.$$

If A and B are square matrices of order n, then AB and BA have the same eigenvalues. Thus, $P\Sigma P^T$ and $P^T P\Sigma$ have exactly the same eigenvalues. Because of $P^T P = I_c$, we have

$$\Sigma = P^T P\Sigma.$$

This implies

$$\tau_i = \lambda_i, \quad \sigma_i = 1-\lambda_i.$$

Thus, if $\lambda_i \in [-1,1]$, then $\sigma_i \in [0,2]$. □

4.3. Optimization

To determine the solution of optimization problem (4.5), we apply an alternative optimization framework to update μ_i,P,Σ, H,R, and G. In each update, only the corresponding variable is updated and other variables remain unchanged. We summarize the steps of our optimization framework in Algorithm 4.2. The whole optimization process can be divided into six steps:

1. Updating μ_i. When P,Σ,H,R,G are fixed, the original optimization problem is simplified as follows:

$$\begin{aligned} \min_{\mu_i} \ & \sum_{i=1}^{V} \frac{1}{\mu_i} \|I_n - P\Sigma P^T - L_i\|_F^2 \\ \text{s.t.} \ & \|\mu\|_1 = 1, \quad \mu \geq 0. \end{aligned}$$

Let $g_i = \|I_n - P\Sigma P^T - L_i\|_F$, we have

$$\begin{aligned} \min_{\mu_i} \ & \sum_{i=1}^{V} \frac{g_i^2}{\mu_i} \\ \text{s.t.} \ & \|\mu\|_1 = 1, \quad \mu \geq 0. \end{aligned}$$

Using the Lagrange multiplier method to optimize μ_i, we equate the derivative of the augmented Lagrange function $\sum_{i=1}^{V} g_i^2/\mu_i + \gamma(\sum_{i=1}^{V} \mu_i)$ to zero — i.e.

$$-\frac{g_i^2}{\mu_i^2} + \gamma = 0, \quad i = 1,\ldots,V.$$

This holds if and only if

$$\frac{g_1}{\mu_1} = \frac{g_2}{\mu_2} = \cdots = \frac{g_V}{\mu_V},$$

hence

$$\mu_i = \frac{g_i}{g_1 + g_2 + \cdots + g_V}, \quad i = 1,\ldots,V. \tag{4.6}$$

2. Updating P. When μ_i, Σ, H, R, G are fixed, the original optimization problem is simplified as follows:

$$\min_{P^TP=I_n} \operatorname{tr}\left(H^T(I_n - P\Sigma P^T)H\right) + \sum_{i=1}^{V} \frac{1}{\mu_i}\|I_n - P\Sigma P^T - L_i\|_F^2.$$

Let $L = I_n - P\Sigma P^T$, the above objective function is equivalent to the following form:

$$\operatorname{tr}(H^T LH) + \sum_{i=1}^{V} \frac{1}{\mu_i}\|L - L_i\|_F^2 = \operatorname{tr}(HH^T L) + \sum_{i=1}^{V} \frac{1}{\mu_i} \operatorname{tr}\left(L^2 - 2L_i L + L_i^2\right).$$

Let

$$A = \sum_{i=1}^{V} \frac{1}{\mu_i} > 0, \quad B = HH^T - 2\sum_{i=1}^{V} \frac{1}{\mu_i} L_i. \tag{4.7}$$

Then

$$\operatorname{tr}(AL^2 + BL) + \sum_{i=1}^{V} \frac{1}{\mu_i} \operatorname{tr}\left(L_i^2\right). \tag{4.8}$$

Further, substituting $I_n - P\Sigma P^T = L$ into (4.8) and ignoring constant term $\sum_{i=1}^{V} \operatorname{tr}(L_i^2)/\mu_i$ yields

$$\begin{aligned} \operatorname{tr}(AL^2 + BL) &= \operatorname{tr}\left[A(I_n - 2P\Sigma P^T + P\Sigma^2 P^T) + B - BP\Sigma P^T\right] \\ &= -\operatorname{tr}(\Sigma P^T BP) + \operatorname{tr}(A\Sigma^2 - 2A\Sigma) + \operatorname{tr}(AI_n + B). \end{aligned} \tag{4.9}$$

When updating P, we only need to solve the following optimization problem:

$$\max_{P^TP=I_n} \operatorname{tr}(\Sigma P^T BP), \tag{4.10}$$

where B is defined in (4.7).

Inspired by [38], we use the Lagrange multiplier method and KKT conditions [1], thus obtaining if $\lambda_i \neq 0$, then P_i is the eigenvector of B, where P_i is the i-th column of P. Using the symmetry of B, we represent B in the form

$$B = \sum_{i=1}^{n} \alpha_i \eta_i \eta_i^T,$$

where α_i is i-th eigenvalue of B and η_i is the eigenvector corresponding to α_i. In this case, we obtain

$$\operatorname{tr}(\Sigma P^T BP) = \sum_{i=1}^{n} \lambda_i P_i^T BP_i = \sum_{i=1}^{n} \lambda_i \tilde{\alpha}_i, \tag{4.11}$$

where $\tilde{\alpha}_i$ is rearrangement of α_i. To achieve the maximum value of Eq. (4.11), the order of λ_i and $\tilde{\alpha}_i$ must be consistent. That is, if $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, then $\tilde{\alpha}_1 \leq \tilde{\alpha}_2 \leq \cdots \leq \tilde{\alpha}_n$.

Therefore, P is a matrix composed of eigenvectors of B and each column is sorted according to the value of λ_i.

3. Updating Σ. When μ_i, P, H, R, G are fixed, according to Eq. (4.9), the original optimization problem is simplified as follows:

$$\begin{aligned} &\min_{\Sigma} \ \mathrm{tr}\left[A\Sigma^2 - (2AI_n + P^T BP)\Sigma\right] \\ &\text{s.t. } -1 \le \lambda_i \le 1, \end{aligned} \tag{4.12}$$

where $A = \sum_{i=1}^{V} 1/\mu_i > 0$. Because the diagonal elements of Σ are uncorrelated, Eq. (4.12) is adjusted as follows:

$$\min_{-1\le\lambda_i\le 1} A\lambda_i^2 - \left(2A + P_i^T BP_i\right)\lambda_i. \tag{4.13}$$

The solution of Eq. (4.13) is given by

$$\lambda_i = \begin{cases} 1, & \text{if} \quad \dfrac{P_i^T BP_i}{2A} > 0, \\ -1, & \text{if} \quad \dfrac{P_i^T BP_i}{2A} < -2, \\ 1 + \dfrac{P_i^T BP_i}{2A}, & \text{otherwise.} \end{cases}$$

4. Updating H. When μ_i, P, Σ, R, G are fixed, the original optimization problem is simplified as follows:

$$\min_{H^T H = I_c} \mathrm{tr}(H^T LH) + \beta \|H - GR\|_F^2.$$

We note that

$$\begin{aligned} &\mathrm{tr}(H^T LH) + \beta \|H - GR\|_F^2 \\ &= \mathrm{tr}(H^T LH) + \beta \,\mathrm{tr}\left[(H^T - R^T G^T)(H - GR)\right] \\ &= \mathrm{tr}(H^T LH) - 2\beta \,\mathrm{tr}(H^T GR) + \beta \,\mathrm{tr}(I_c + G^T G) \\ &= \mathrm{tr}(H^T LH) - \mathrm{tr}(H^T W) + \beta \,\mathrm{tr}(I_c + G^T G), \end{aligned} \tag{4.14}$$

where $W = 2\beta GR$. Consequently, we derive the optimization problem

$$\min_{H^T H = I_c} \mathrm{tr}(H^T LH) - \mathrm{tr}(H^T W),$$

which is equivalent to

$$\max_{H^T H = I_c} \mathrm{tr}(H^T DH) + \mathrm{tr}(H^T W), \tag{4.15}$$

where $D = \gamma I_n - L$ is a positive definite matrix. Let $N = DH + W$, Eq. (4.15) is further changed into

$$\max_{H^T H = I_c} \mathrm{tr}(H^T N). \tag{4.16}$$

Note that N is related to H, we hardly solve this optimization problem. However, if N remains unchanged, we can use Theorem 4.4 below to solve the problem (4.16).

Theorem 4.4. *If $Y = U\Lambda V^T$ is the SVD decomposition of Y, then $H = UV^T$ is the optimal solution of the problem*

$$\max_{H^TH=I_c} \operatorname{tr}(H^TY).$$

Therefore, N is updated when H has been updated. Algorithm 4.1 presents the process of solving this optimization problem.

Algorithm 4.1 Algorithm to Solve the Problem of Eq. (4.16)

Input: L, G, R, β, γ.
Output: H.

1: Compute $D = \gamma I_n - L, W = 2\beta GR$.
2: **repeat**
3: Compute $N = DH + W$.
4: Compute $U_1S_1V_1^T = N$ via compact SVD of N.
5: Compute $H = U_1V_1^T$.
6: **until** converge.

5. Updating R. When μ_i, P, Σ, H, G are fixed, according to Eq. (4.14), we derive the following form:

$$\max_{R^TR=I_c} \operatorname{tr}(R^TG^TH). \tag{4.17}$$

By Theorem 4.4, if USV^T is the SVD decomposition of G^TH, the optimal solution of the Eq. (4.17) is $R = UV^T$.

6. Updating G. When μ_i, P, Σ, H, R are fixed, the original optimization problem is simplified as follows:

$$\min_{G\in\mathbb{S}} \|H - GR\|_F^2.$$

Because $G \in \mathbb{S}$, this optimization problem is difficult to solve. Since each row of G is independent, we update G row by row as follows:

$$G_{ij} = \begin{cases} 1, & j = \arg\min_k \|h_i - r_k\|_F^2, \\ 0, & \text{otherwise}, \end{cases} \tag{4.18}$$

where h_i is the i-th row of H and r_k is the k-th row of R.

4.4. Computational complexity

For the proposed method, the primary computational cost arises from updating P, which requires eigenvalue decomposition and the corresponding calculation complexity is $\mathcal{O}(n^3)$. The complexity of Algorithm 4.1 is $\mathcal{O}(t_1nc^2)$, where t_1 is the number of iterations. The computational cost of the proposed method is $\mathcal{O}(t_2n^3 + t_2t_1nc^2 + t_2n)$, where t_2 is the number of iterations in Algorithm 4.2. If $t_2 \ll n$, the complexity of proposed method is approximated by $\mathcal{O}(t_2n^3)$.

Algorithm 4.2 One-Step Multi-View Spectral Clustering via Dynamical Optimization of Consensus Laplacian Matrix (osDOSC)

Input: Datasets:$\{X_1, X_2, \cdots, X_V\}$, number of clusters c, number of nearest neighbors k, parameter β, γ.
Output: Cluster indicator G.

Compute L_i of each view by $L_i = I_n - D_i^{-1/2} W_i D_i^{-1/2}$.
Initialize μ_i as 1/V, H as $\mathbf{1}_{n\times c}$, P as I_n, Σ as I_n, R as $\mathbf{1}_{c\times c}$, G as $\mathbf{0}_{c\times c}$.
repeat
    Compute μ_i by Eq. (4.6).
    Compute P by solving Eq. (4.10).
    Compute Σ by solving Eq. (4.13).
    Compute H by Algorithm 4.1.
    Compute R by solving Eq. (4.17).
    Compute G by Eq. (4.18).
until converge.

4.5. Convergence analysis

The convergence of Algorithm 4.1 has been proved by [26]. Next, we demonstrate the convergence of Algorithm 4.2. The optimization problem (4.5) is not a convex problem, thus it is difficult to find its global optimal solution. In the proposed alternative optimization framework, only one variable is optimized for each update. As a result, the objective function value is reduced after each iteration, which indicates that the objective function is monotonically decreasing. Considering that the optimal Laplacian matrix is a positive semi-definite matrix, the objective function of the optimization problem takes 0 as a lower bound. Therefore, the value of the objective function converges to a local optimal solution through continuous iteration.

5. Results and Discussions

In this section, we compare the experimental results of the proposed method and other clustering methods on multiple real datasets, and discuss the parameter sensitivity and convergence of our proposed method.

5.1. Materials

A number of widely used real datasets with different sizes and numbers of views are introduced for performance evaluation.

1. **BBC:** `http://mlg.ucd.ie/datasets/segment.html`. This multi-view dataset is composed of BBC news documents. Each documents is divided into four parts to form four views.

2. **BBCSport:** `http://mlg.ucd.ie/datasets/`. This multi-view dataset contains 544 documents in total and is divided into five categories. Each document has two views with 3183 dimensions and 3203 dimensions respectively.

3. **NGs:** `http://lig-membres.imag.fr/grimal/data.html`. This multi-view dataset consists of newsgroup documents. It contains three views, each view containing 500 samples and five labels.

4. **100leaves:** `https://github.com/cshaowang/gmc/blob/master/Dataset`. The samples of this multi-view dataset are collected from 100 different plant species, including three views and 1600 samples.

5. **ORL:** `https://github.com/XIAOCHUN-CAS/Consistent-and-Specific-Multi-View-Subspace-Clustering`. This multi-view dataset is composed of 400 face recognition images. It has 40 categories and three different views.

5.2. Compared methods

In order to prove the effectiveness of our method, we compare our method with 10 state-of-the-art clustering methods including:

1. **Single-view spectral clustering (SCbest) [31]**. This is traditional spectral clustering method. We use single-view spectral clustering to output clustering results on all views, and take the best clustering result as the final result of this method.

2. **FastMICE [9]**. This is a multi-view clustering method based on integration. It proposes an idea of early-late fusion, which forms a consistent bipartite graph through base clustering for final clustering.

3. **TBGL [35]**. This is a multi-view clustering method based on bipartite graph learning, which is constructed by a new anchor selection strategy.

4. **OPMC [22]**. This is a multi-view clustering method based on matrix decomposition. It removes non-negative constraints and learns labels at the same time, which saves time and cost.

5. **MCLES [2]**. This clustering method clusters the data in latent embedding space. It can also learn the structural characteristics of data and the clustering indication result simultaneously.

6. **AWP [25]**. This paper extends the spectral rotation of single view to the case of multiple views.

7. **WMSC [42]**. This paper proposes a new multi-view spectral clustering method based on spectral perturbation theory to update the weights of each view.

8. **AMGL [24]**. This multi-view clustering method can learn the optimal weight without parameter adjustment.

9. **RMSC [33]**. This is a multi-view clustering method based on Markov chain. It tries to learn a suitable transition probability matrix for spectral clustering.

10. **AASC [10]**. This spectral clustering method tries to learn the best combination in affinity matrices.

We also compare our method with DSSC, which is part of our method after removing spectral rotation. It only finds the optimal Laplacian matrix and discretizes spectral embedding through k-means algorithm. The comparison with DSSC is to illustrate the necessity of one-step spectral clustering.

5.3. Experimental settings

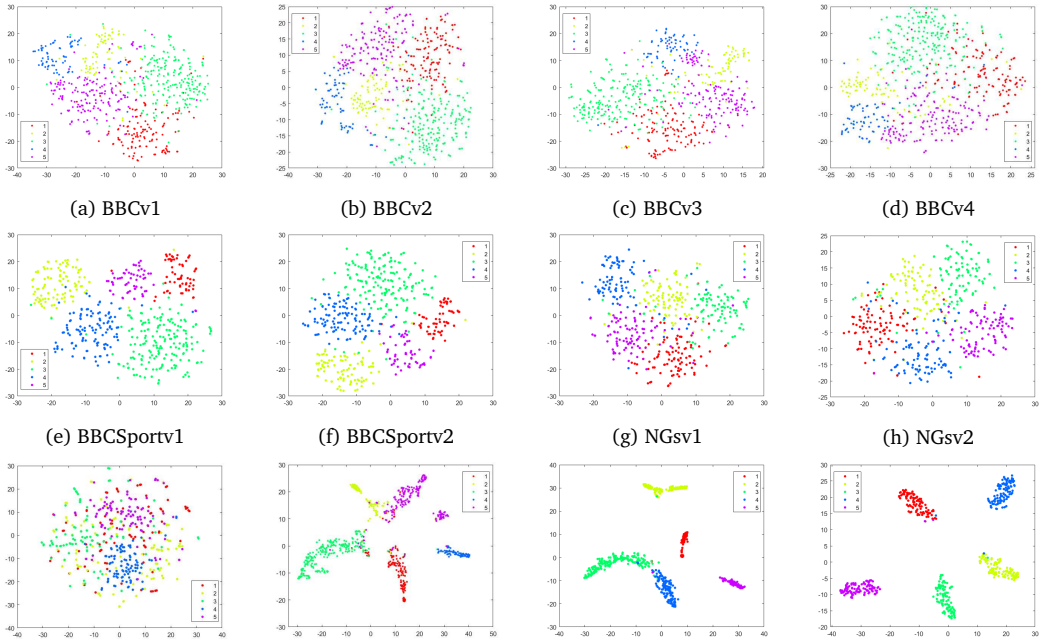
For graph-based clustering methods, we uniformly use k nearest neighbors to construct graphs, where $k = 10$ remains unchanged on all experimental datasets. In this experiment, we use five cluster evaluation metrics: adjusted rand index (ARI), normalized mutual information (NMI), accuracy (ACC), F-score, Purity to evaluate the clustering ability of all methods. The higher their values, the better the clustering effects.

In order to avoid the influence of randomness, all the methods are repeated for 20 times and take the average value as their clustering results. All these experiments were done in MATLAB 2021b environment. Note that most clustering methods contain parameters. In the experiments, we keep the default parameters set by their authors for compared methods. For our method, it is not to take different parameter values on different datasets, but to keep $\beta = 0.001, \gamma = 2.7$ constant.

5.4. Clustering results

We use different clustering evaluation indicators to evaluate the clustering effects of our proposed method and many clustering methods on five experimental datasets, which is shown in Tables 1-5. In Fig. 3, we show the t-SNE visualization results of each view in BBC, BBCSport, and NGs. There are no single-view t-SNE results of 100leaves and ORL because they have too many categories (100 clusters and 40 clusters). At the same time, we also show the t-SNE visualization results of the rotated spectral embedding in our method. Furthermore, we present the visualization results of compared methods in Fig. 4. For illustration purpose, we only present the results on BBCSport. Further, we compare the clustering performance of traditional single-view spectral clustering method and our method in Fig. 5, which shows that our method can still get relatively great clustering results, even when the quality of some views is poor. From Tables 1-5, we can see that:

- The proposed method performs much better than other clustering methods. For example, using ARI, NMI and ACC to evaluate the clustering performance, our method is 6.2%, 6.0% and 5.2% higher than the second clustering method (WMSC) in BBC dataset. This shows the rationality and superiority of putting the search of optimal

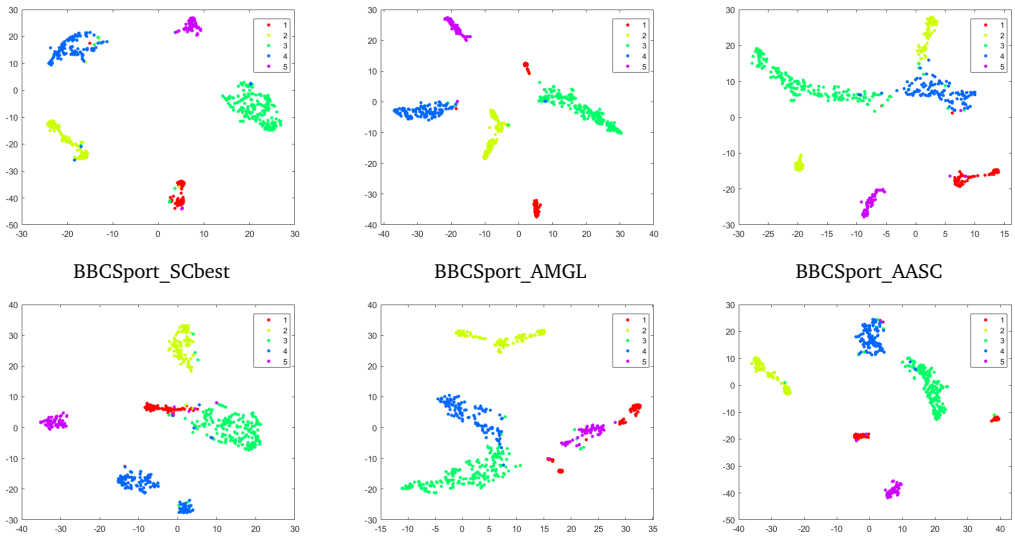


(a) BBCv1 (b) BBCv2 (c) BBCv3 (d) BBCv4

(e) BBCSportv1 (f) BBCSportv2 (g) NGsv1 (h) NGsv2

(i) NGsv3 (j) BBC_Ours (k) BBCSport_Ours (l) NGs_Ours

Figure 3: The tsne visualization for each dataset of all views and the embedding visualization by our method.



BBCSport_SCbest BBCSport_AMGL BBCSport_AASC

BBCSport_RMSC BBCSport_MCLES BBCSport_WMSC

Figure 4: tSNE results for compared methods in BBCSport data.

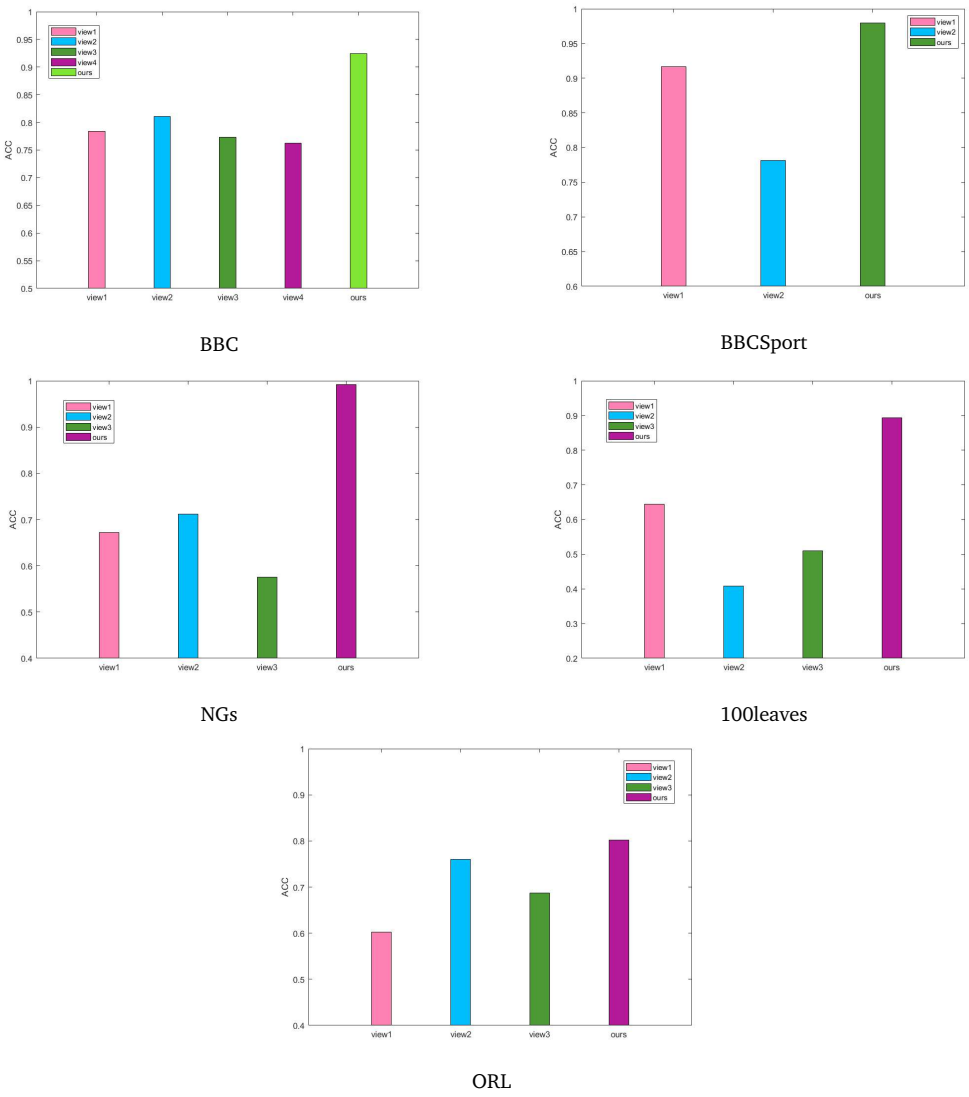


Figure 5: Performance comparison with single-view spectral clustering method.

Laplacian matrix and the learning and discretization of spectral embedding into a unified framework. Our method can make better use of multi-view data information to achieve more accurate clustering.

- Note that although DSSC performs well on most experimental datasets, the experimental effect of our method is still better, which shows that spectral rotation is indispensable. Moreover, there are many spectral embeddings satisfying the first step of spectral clustering, but it is still necessary to find a more suitable spectral embedding for the second step instead of directly using k-means algorithm.

Fig. 3 shows that before clustering, the data from different clusters are very close to each other, with some even overlapping. However, after applying our method, data from the same category become tightly clustered, while data from different categories are separated, resulting in a strong clustering effect. Furthermore, as shown in Fig. 5, even though some views are of poor quality and may not be conducive to information extraction, our method is still able to effectively utilize each view and extract useful information through iterative updates.

Table 1: Performance comparisons of different methods evaluated by ARI.

Dataset	BBC	BBCSport	NGs	100leaves	ORL
SCbest	0.627	0.8623	0.5569	0.5337	0.6793
AMGL	0.5413	0.7539	0.7058	0.6107	0.6412
AASC	0.71	0.5752	0.9445	0.6696	0.6216
RMSC	0.3687	0.7595	0.9119	0.4296	0.4845
OPMC	0.4428	0.6045	0.776	0.5165	0.4155
AWP	0.7556	0.7876	0.6542	0.7205	0.6576
WMSC	0.7629	0.773	0.9491	0.828	0.6708
MCLES	0.3918	0.4508	0.8821	0.5577	0.5068
TBGL	0.4337	0.7607	0.9456	0.1075	0.4969
FastMICE	0.1966	0.1699	0.7167	0.812	0.633
DSSC	0.7459	0.7053	0.9749	0.8326	0.6678
Ours	**0.8253**	**0.9494**	**0.9800**	**0.8651**	**0.7047**

Table 2: Performance comparisons of different methods evaluated by NMI.

Dataset	BBC	BBCSport	NGs	100leaves	ORL
SCbest	0.5912	0.8483	0.6076	0.8226	0.8796
AMGL	0.5957	0.8180	0.8148	0.9273	0.8690
AASC	0.7120	0.7146	0.9358	0.9235	0.8683
RMSC	0.4906	0.7526	0.9025	0.7728	0.7810
OPMC	0.5421	0.6635	0.7886	0.8441	0.7646
AWP	0.7291	0.8024	0.6946	0.9058	0.8691
WMSC	0.7322	0.8005	0.9532	0.9534	0.8748
MCLES	0.5335	0.6116	0.8749	0.8811	0.8129
TBGL	0.5683	0.8299	0.9287	0.8029	**0.8988**
FastMICE	0.3846	0.3662	0.7364	0.9493	0.8721
DSSC	0.7248	0.7628	0.9669	0.9557	0.8721
Ours	**0.7923**	**0.9291**	**0.9739**	**0.9623**	0.8820

Table 3: Performance comparisons of different methods evaluated by ACC.

Dataset	BBC	BBCSport	NGs	100leaves	ORL
SCbest	0.8104	0.9166	0.7117	0.6442	0.7604
AMGL	0.7169	0.8355	0.7714	0.7968	0.7420
AASC	0.8336	0.6979	0.9715	0.7923	0.7399
RMSC	0.6464	0.8459	0.9511	0.5573	0.5913
OPMC	0.6545	0.7384	0.8727	0.6035	0.5524
AWP	0.8730	0.8989	0.8160	0.7875	0.7400
WMSC	0.8722	0.8217	0.9660	0.8513	0.7518
MCLES	0.6772	0.6478	0.9370	0.7116	0.6305
TBGL	0.5766	0.8879	0.9780	0.6719	0.7775
FastMICE	0.5346	0.4976	0.8661	0.8423	0.7238
DSSC	0.8581	0.7973	0.9900	0.8575	0.7556
Ours	**0.9241**	**0.9798**	**0.9920**	**0.8931**	**0.8025**

Table 4: Performance comparisons of different methods evaluated by F-score.

Dataset	BBC	BBCSport	NGs	100leaves	ORL
SCbest	0.7137	0.8952	0.6525	0.5383	0.6869
AMGL	0.6690	0.8231	0.7751	0.6155	0.6501
AASC	0.7853	0.6988	0.9558	0.6733	0.6312
RMSC	0.5236	0.8148	0.9302	0.4353	0.4971
OPMC	0.5710	0.6982	0.8224	0.5221	0.4306
AWP	0.8134	0.8394	0.7250	0.7233	0.6660
WMSC	0.8190	0.8293	0.9600	0.8297	0.6786
MCLES	0.5520	0.6254	0.9062	0.5628	0.5202
TBGL	0.6130	0.8251	0.9564	0.1229	0.5127
FastMICE	0.4178	0.4306	0.7756	0.8139	0.6423
DSSC	0.8059	0.7820	0.9799	0.8342	0.6757
Ours	**0.8671**	**0.9614**	**0.9839**	**0.8665**	**0.7116**

Table 5: Performance comparisons of different methods evaluated by Purity.

Dataset	BBC	BBCSport	NGs	100leaves	ORL
SCbest	0.8153	0.9266	0.7297	0.6680	0.7884
AMGL	0.7520	0.8625	0.8053	0.8373	0.7741
AASC	0.8447	0.7494	0.9722	0.8304	0.7805
RMSC	0.6843	0.8696	0.9541	0.5870	0.6223
OPMC	0.7177	0.7972	0.8851	0.6225	0.5895
AWP	0.8730	0.8989	0.8160	0.8050	0.7525
WMSC	0.8783	0.8704	0.9698	0.8776	0.7779
MCLES	0.6836	0.6987	0.9417	0.7497	0.6606
TBGL	0.5839	0.8879	0.9780	0.6888	0.8050
FastMICE	0.5790	0.5744	0.8721	0.8648	0.7561
DSSC	0.8696	0.8339	0.9900	0.8818	0.7803
Ours	**0.9241**	**0.9798**	**0.9920**	**0.8975**	**0.8050**

5.5. Parameter sensitivity

In this section, we discuss the influence of parameters on our method. The method has two parameters β and γ. We use the grid search method to determine appropriate parameters. We tuned β from $[10^{-4}, 10^{-3}, \cdots, 10]$ and γ from $[2, 3, \cdots, 8]$ to detect the change of clustering results of our proposed method on each experimental datasets (evaluated by ARI). Fig. 6 shows the parameter sensitivity of our proposed method. As shown in Fig. 6, when β takes a relatively smaller value, the clustering results are more stable and accurate, and this proposed method is sensitive to γ, so it is necessary to choose γ carefully. γ is used in the update of H to ensure that D is a positive definite matrix. Generally, $\gamma > 2$ can meet

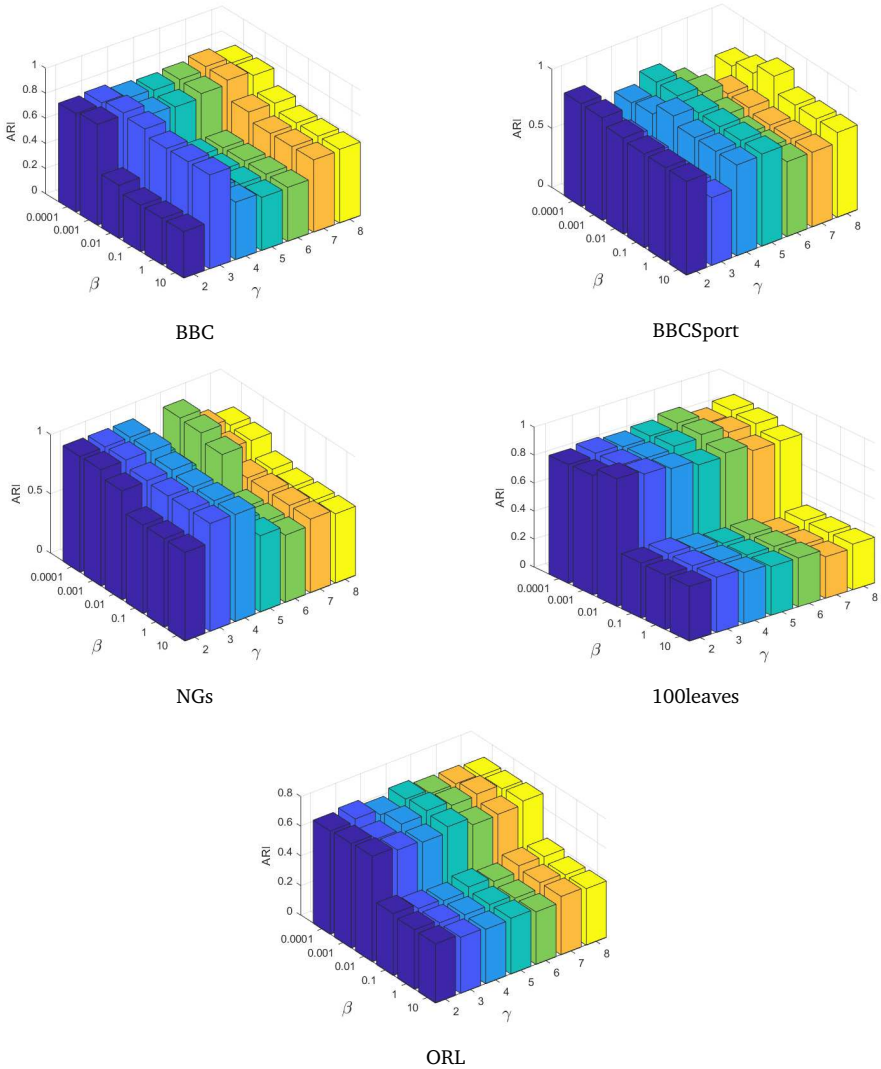


Figure 6: ARI against parameters β and γ on all datasets.

this requirement, and the power method can also be applied to get the value of γ [26]. In addition, when $\beta = 10^{-3}$ and $\gamma = 2.7$, our proposed method can achieve relatively great performance on all the real datasets.

5.6. Convergence

Algorithms 4.1 and 4.2 guarantee theoretical convergence, but the speed of convergence is addressed. Here we experimentally show the convergence of our method for each experimental dataset and visualize the convergence trend. As shown in Fig. 7, our method can converge quickly in a few steps, which shows that our method is very efficient.

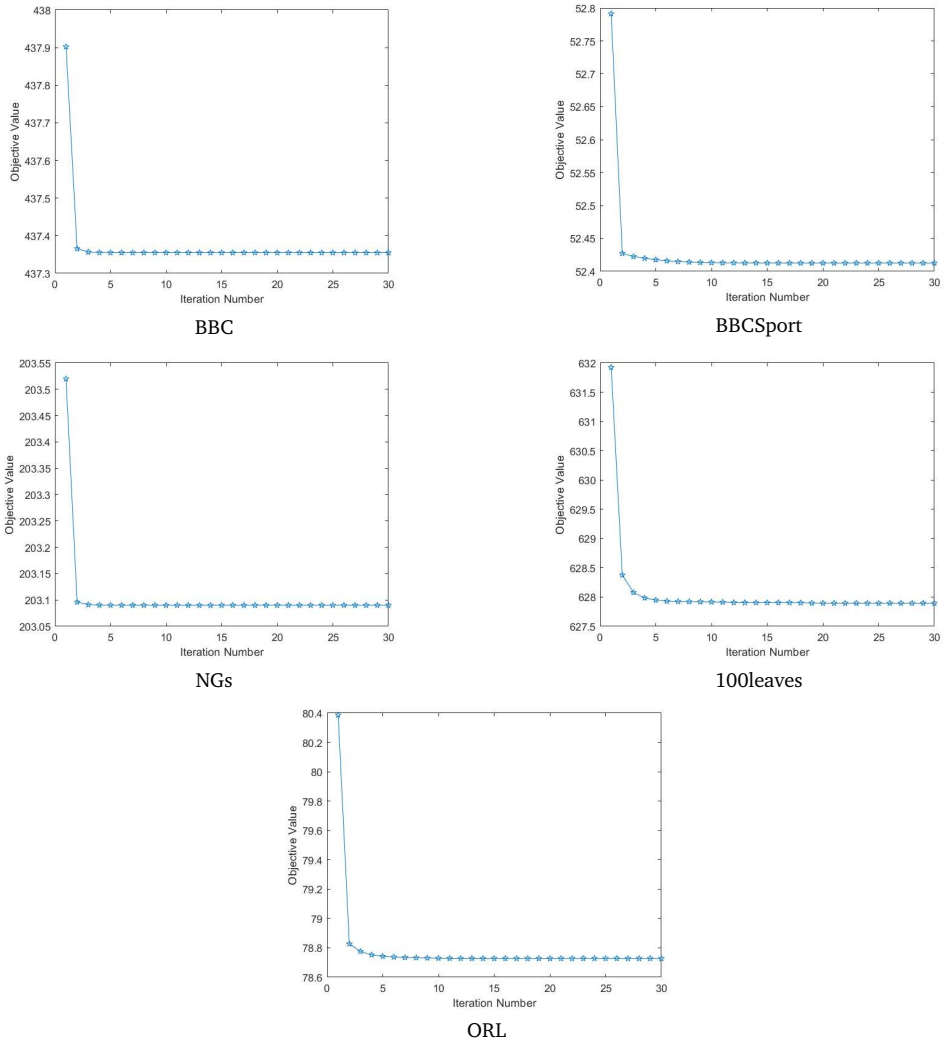


Figure 7: Convergence speed of our method on all datasets.

6. Conclusions

In this paper, we propose a novel multi-view data analysis model — viz. a one-step multi-view spectral clustering method osDOSC, which realizes the dynamic optimization of Laplacian matrix and the discretization of spectral embedding in a unified framework. The one-step strategy enables simultaneous optimization of consensus Laplacian matrix, the learning of the weight of views, and clustering through spectral rotation. From one hand, taking into consideration on the noise and correlation issues in multi-view data, we model the consensus Laplacian matrix as located in the neighborhood of view-specific Laplacian matrices, instead of linear combination. From another perspective, we dynamically optimize the consensus Laplacian matrix through genius design on the optimization framework. Moreover, we adaptively evaluate the importance of views and adjust weights to each view, so as to optimize the sources of information. The accuracy and superiority of our method is demonstrated by using various real-world datasets. Future work will be devoted to integrate high-order graph learning into the one-step spectral clustering framework.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (Grant Nos. 12271522, 62173333) and by the Beijing Natural Science Foundation (Grant No. Z210002).

References

[1] D.P. Bertsekas, *Nonlinear programming*, J Oper Res Soc **48**(3), 334–334 (1997).

[2] M.-S. Chen, L. Huang, C.D. Wang and D. Huang, *Multi-view clustering in latent embedding space*, in: *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(4), 3513–3520 (2020).

[3] Z. Cheng, H. Li, Y. Asano, Y. Zheng and I. Sato, *Multi-view 3d reconstruction of a texture-less smooth surface of unknown generic reflectance*, in: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16221–16230, IEEE (2021).

[4] K. Dong, A. Sun, J.-J. Kim and X. Li, *Syntactic multi-view learning for open information extraction*, in: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva and Y. Zhang (Eds.), pp. 4072–4083, Association for Computational Linguistics (2022).

[5] L. Du, H. Wang, J. Zhang, S. Zhang, L. Guo and J.-F. Han, *Adaptive structured sparse multi-view canonical correlation analysis for multimodal brain imaging association identification*, Sci. China Inf. Sci. **66**, 142106 (2023).

[6] K. Fan, *On a theorem of Weyl concerning eigenvalues of linear transformations I*, Proc. Natl. Acad. Sci. USA **35**(11), 652–655 (1949).

[7] L. Hagen and A.B. Kahng, *New spectral methods for ratio cut partitioning and clustering*, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **11**(9), 1074–1085 (1992).

[8] S. El Hajjar, F. Dornaika and F. Abdallah, *One-step multi-view spectral clustering with cluster label correlation graph*, Inf. Sci. **592**, 97–111 (2022).

[9] D. Huang, C.-D. Wang and J.-H. Lai, *Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity*, IEEE Trans. Knowl. Data Eng. **35**(11), 11388–11402 (2023).
[10] H.-C. Huang, Y.-Y. Chuang and C.-S. Chen, *Affinity aggregation for spectral clustering*, in: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 773–780, IEEE (2012).
[11] J. Huang, F. Nie and H. Huang, *Spectral rotation versus k-means in spectral clustering*, in: *Proceedings of the AAAI Conference on Artificial Intelligence* **27**(1), 431–437 (2013).
[12] Z. Jiang and X. Liu, *Adaptive KNN and graph-based auto-weighted multi-view consensus spectral learning*, Inf. Sci. **609**, 1132–1146 (2022).
[13] P. Jing, Y. Su, Z. Li and L. Nie, *Learning robust affinity graph representation for multi-view clustering*, Inf. Sci. **544**, 155–167 (2021).
[14] A. Khan and P. Maji, *Approximate graph Laplacians for multimodal data clustering*, IEEE Trans. Pattern Anal. Mach. Intell. **43**(4), 798–813 (2021).
[15] A. Kumar and H. Daume, *A co-training approach for multi-view spectral clustering*, in: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 393–400, Omnipress (2011).
[16] J. Li, X. Zhang, J. Wang, X. Wang, Z. Tan and H. Sun, *Projection-based coupled tensor learning for robust multi-view clustering*, Inf. Sci. **632**(C), 664–677 (2023).
[17] L. Li and H. He, *Bipartite graph based multi-view clustering*, IEEE Trans. Knowl. Data Eng. **34**(7), 3111–3125 (2020).
[18] Y. Li, F. Nie, H. Huang, J. Huang, *Large-scale multi-view spectral clustering via bipartite graph*, in: *Proceedings of the AAAI Conference on Artificial Intelligence* **29**(1), 2750–2756 (2015).
[19] J. Liang, X. Liu, L. Bai, F. Cao and D. Wang, *Incomplete multi-view clustering via local and global co-regularization*, Sci. China Inf. Sci. **65**, 152105 (2022).
[20] Y. Liang, D. Huang and C.-D. Wang, *Consistency meets inconsistency: A unified graph learning framework for multi-view clustering*, in: *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1204–1209, IEEE (2019).
[21] J. Liu, J. Liu, S. Yan, R. Jiang, X. Tian, B. Gu, Y. Chen, C. Shen and J. Huang, *Mpc: Multi-view probabilistic clustering*, in: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9499–9508, IEEE (2022).
[22] X. Liu, L. Liu, Q. Liao, S. Wang, Y. Zhang, W. Tu, C. Tang, J. Liu and E. Zhu, *One pass late fusion multi-view clustering*, in: *Proceedings of the 38th International Conference on Machine Learning* **139**, 6850–6859, PMLR (2021).
[23] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang and W. Gao, *Late fusion incomplete multi-view clustering*, IEEE Trans. Pattern Anal. Mach. Intell. **41**(10), 2410–2423 (2019).
[24] F. Nie, J. Li and X. Li, *Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification*, in: *IJCAI'16: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1881–1887, AAAI Press (2016).
[25] F. Nie, L. Tian and X. Li, *Multiview clustering via adaptively weighted procrustes*, in: *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2022–2030, Association for Computing Machinery (2018).
[26] F. Nie, R. Zhang and X. Li, *A generalized power iteration method for solving quadratic problem on the stiefel manifold*, Sci. China Inf. Sci. **60**, 112101 (2017).
[27] Y. Pang, J. Xie, F. Nie and X. Li, *Spectral clustering by joint spectral embedding and spectral rotation*, IEEE Trans. Cybern. **50**(1), 247–258 (2020).
[28] X. Sun, Y. Tian, W. Lu, P. Wang, R. Niu, H. Yu and K. Fu, *From single- to multi-modal remote sensing imagery interpretation: A survey and taxonomy*, Sci. China Inf. Sci. **66**, 140301 (2023).

[29] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang and E. Zhu, *Unified one-step multi-view spectral clustering*, IEEE Trans. Knowl. Data Eng. **35**(6), 6449–6460 (2023).
[30] C. Tang, X. Liu, X. Zhu, E. Zhu, Z. Luo, L. Wang and W. Gao, *CGD: Multi-view clustering via cross-view graph diffusion*, in: *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(04), 5924–5931 (2020).
[31] U. Von Luxburg, *A tutorial on spectral clustering*, Stat. Comput. **17**(4), 395–416 (2007).
[32] H. Wang, Y. Yang and B. Liu, *GMC: Graph-based multi-view clustering*, IEEE Trans. Knowl. Data Eng. **32**(6), 1116–1129 (2020).
[33] R. Xia, Y. Pan, L. Du and J. Yin, *Robust multi-view spectral clustering via low-rank and sparse decomposition*, in: *Proceedings of the AAAI Conference on Artificial Intelligence* **28**(1), 2149–2155 (2014).
[34] T. Xia, D. Tao, T. Mei and Y. Zhang, *Multiview spectral embedding*, IEEE Trans. Syst. Man Cybern. Part B Cybern. **40**(6), 1438–1446 (2010).
[35] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding and D. Tao, *Tensorized bipartite graph learning for multi-view clustering*, IEEE Trans. Pattern Anal. Mach. Intell. **45**(4), 5187–5202 (2023).
[36] B. Yang, Z. Huang, Y. Li, H. Zhou, H. Li, G. Zhang and H. Bao, *Hybrid3D: Learning 3D hybrid features with point clouds and multi-view images for point cloud registration*, Sci. China Inf. Sci. **66**, 172101 (2023).
[37] X. Yu, H. Liu, Y. Wu and C. Zhang, *Fine-grained similarity fusion for multi-view spectral clustering*, Inf. Sci. **568**, 350–368 (2021).
[38] S. Zhou, X. Liu, J. Liu, X. Guo, Y. Zhao, E. Zhu, Y. Zhai, J. Yin and W. Gao, *Multi-view spectral clustering with optimal neighborhood Laplacian matrix*, in: *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(4), 6965–6972 (2020).
[39] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu and C. Wang, *Graph PCA hashing for similarity search*, IEEE Trans. Multim. **19**(9), 2033–2044 (2017).
[40] X. Zhu, H.-I. Suk, L. Wang, S.-W. Lee and D. Shen, *A novel relational regularization feature selection method for joint regression and classification in ad diagnosis*, Med. Image Anal. **38**, 205–214 (2017).
[41] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei and P. Zhu, *One-step multi-view spectral clustering*, IEEE Trans. Knowl. Data Eng. **31**(10), 2022–2034 (2019).
[42] L. Zong, X. Zhang, X. Liu and H. Yu, *Weighted multi-view spectral clustering based on spectral perturbation*, in: *Proceedings of the AAAI Conference on Artificial Intelligence* **32**(1), 4621–4628 (2018).