UFMKSC: A Uniform Framework for Multiple Kernel Spectral Clustering Using a Noise-Free Laplacian Matrix

Feng Wang[1,2], Shun Mao[1]✉, Yuanfei Deng[3], and Yixiu Qin[1,4]

[1] School of Computer Science, South China Normal University, Guangzhou 510631, People's Republic of China
{fwang,yxqin}@m.scnu.edu.cn, smao@tulip.academy
[2] School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, People's Republic of China
[3] School of Artificial Intelligence, Guangdong Polytechnic Institute, Guangzhou 510091, People's Republic of China
dengyf@gdpi.edu.cn
[4] School of Artificial Intelligence, Zhoukou Normal University, Zhoukou 466001, People's Republic of China

Abstract. Multi-view spectral clustering gains popularity and is successfully applied in various fields due to its superior performance. Existing approaches, however, have three drawbacks when enhancing clustering performance with multi-view data: (1) the representation of nodes in the individual view is insufficient because hidden similarity among high-order neighbors is omitted; (2) noise in each view of the data results in poor performance in constructing a uniform representation; and (3) two separate processes of embedding and discretion lead to suboptimal clustering results. To confront the challenges outlined earlier, we propose UFMKSC, a uniform framework for multiple kernel spectral clustering using a noise-free Laplacian matrix. In detail, we first construct similarity matrices with high-order information using the kernel function. Based on constructed matrices, we then calculate their Laplacian matrices and use the most informative eigenpairs that contain cluster information to reduce noise. Finally, we carefully design an efficient iterative optimization algorithm that includes spectral rotation to avoid suboptimal results. Experiments conducted on six public datasets validate the performance of the proposed method.

Keywords: Multi-view spectral clustering · Spectral rotation · Laplacian matrix · Kernel function.

1 Introduction

In real-world scenarios, there is a growing number of data that can be represented by various views [9], as information often comes from diverse perspectives or attributes. How to analyze data quickly and discover the information effectively

contained in it has become a hot topic and clustering analysis is the main task of exploratory data mining and any value discovery of big data in all industries. As one of the most representative *clustering techniques* [1], *spectral clustering* [20], a graph theory-based clustering method, has been successfully applied in various fields, such as `natural language processing` [8, 19], `computer vision` [14], `learning technology` [11, 16, 17] and `gene expression` [7].

We can organize existing studies of spectral clustering into three categories. The **first** category focused on the *similarity matrix construction*, which enhanced the accuracy of the node representation. On the one hand, a part of the research put it into the overall algorithmic framework. Wang et al. [23] generated a dynamic similarity matrix with adaptive neighbors for each view. On the other hand, a portion of the study treated it as predefined. Khan and Maji [6] utilized cosine similarity to gauge the similarity among data and built a similarity matrix on each view. The **second** category paid attention to *subspace generation*, which could eliminate the noise in the original similarity matrix. Wang et al. [24] learned the global structure based on the learned potential Grassmann manifold. These methods of generating potential subspaces could fully use node neighbor information. Sun et al. [22] combined anchor learning and graph construction which could represent the potential data distribution. The **third** category attempted to get the globally optimal clustering results via two individual embedding and discretization processes. Zhou et al. [31] and Liang et al. [10] constructed higher-order optimal *Laplacian matrices* to obtain corresponding data partition matrices, and then the traditional k-means algorithm was employed to discretize the matrices, resulting in clustering outcomes.

Although the above-mentioned spectral clustering algorithms have achieved promising clustering results, they still exhibit certain weaknesses. **First**, the hidden similarity among nodes is ignored, resulting in lower embedding accuracy. **Second**, spectral clustering typically employs eigendecomposition of the Laplacian matrix to identify the eigenvectors corresponding to the most informative eigenvalues, aiming to eliminate less significant feature information within each individual view. However, inconsistencies or conflicts in the feature information used to describe the data across different views, which can be attributed to data noise, can also affect the clustering results. **Third**, spectral clustering involves a two-phase procedure of spectral embedding and discretization, which involves data transmission and transformation. The original graph data needs to undergo two rounds of processing, which can potentially result in loss or inaccuracy of intermediate results, thereby reducing the precision of the algorithm. Furthermore, it becomes challenging to consider the interdependence between embedding and discretization. There is information loss between independent steps, resulting in inconsistent and poorly coordinated clustering outcomes, which may lead to suboptimal results.

To confront the challenges outlined earlier, we propose **UFMKSC**, a *Uniform Framework for Multiple Kernel Spectral Clustering* using a noise-free *Laplacian matrix*. We combine *kernel matrix* information with diagonalized and higher-order information to create similarity matrices for each view, which capture the

richness and complementarity information in different views. Based on similarity matrices, the maximum eigenpair is then obtained using the eigen-decomposition of their Laplacian matrices to get the wealthiest information. A low-rank subspace is constructed for each view to preserve the overall clustering information while eliminating noise. Following that, a joint optimization framework is designed using convex combination to integrate information from multiple perspectives, and spectral rotation is used to solve the two-step optimization problem of spectral embedding and decomposition. Finally, experiments are performed on several benchmark datasets to validate the effectiveness of the proposed clustering algorithm compared to state-of-the-art methods.

The rest of this article is structured as follows. Section 2 reviews the related work. Section 3 describes the proposed algorithm. Experimental results are reported and analyzed in section 4. Finally, conclusion is provided in section 5.

2 Related Work

Existing studies proposed multi-view clustering approaches, roughly divided into three categories based on different clustering strategies: *kernel clustering*, *subspace clustering* and *spectral clustering* [3].

2.1 Kernel clustering

Kernel clustering [18] typically focuses on the study of kernel functions. In the previous work, Liu et al. [12] constructed an adaptive local kernel by choosing different numbers of neighbors, and the similarity between neighbors were measured by a predefined threshold lower bound. However, such methods were susceptible to redundant features. Wang et al. [25] discretized the kernel matrix to reduce redundancy enhance multi-core learning by measuring the correlation between kernels. In contrast, Zhou et al. [29] found maximal mutual information in multiple mapping kernels, which depended more on the chosen kernel function.

2.2 Subspace clustering

Subspace clustering [21] found the latent subspace for all views and then clustered the data projects correctly [15]. Conventionally, Gao et al. [4] performed subspace clustering for each view simultaneously while ensuring uniformity of the clustering structure among views. But the generalization performance was weak for different datasets. Zhang et al. [28] represented data points using low-dimensional subspaces and represented each data as a linear combination of others. However, effective fusion of multi-view information was not fully addressed.

2.3 Spectral clustering

Spectral clustering [20] was suitable for arbitrarily shaped clusters. In order to capture the wide range of multiple correlation interactions in real data, in graph

construction, Wu et al. [26] explored the higher-order correlations of multi-view representations by tensor kernel parametrization based on tensor singular value decomposition. In graph fusion, Yin et al. [27] divided the data into multiple groups for multi-layer clustering and then fused them together.

3 Methodology

This section introduces our method. The standard notations used in this paper are summarized in Table 1.

Table 1: Summary of Notations

Notation	Description	Notation	Description
X	Dataset of n samples	F_v	Cluster indicating matrix of vth view
X_i	ith sample of X	F^*	Uniform indicating matrix
n	Number of samples in X	R	Rotation matrix
k	Number of clusters	Y	Discrete indicating matrix
r	Number of rank	α	Combination coefficients of multi-view
p	Number of views	β	Regularization parameters
$S^{(o)}$	oth order similarity matrix	γ	Fuzzy coefficient parameters
$D^{(o)}$	Degree matrix of $S^{(o)}$	m	Neighbor number parameters of $S^{(o)}$
$L^{(o)}$	Laplacian matrix of $S^{(o)}$	I_n	n-dimensional identity matrix

3.1 High-Order Similarity Matrix

Most existing clustering methodologies use a non-Euclidean similarity metric kernel [5] to reveal similarities between samples, such as linear combination, nonlinear combination, data-dependent combination, etc. On the one hand, high dimensional data cannot be directly reduced since it may result in useful data loss. On the other hand, there is an urgent demand for higher-order information to express the connections between data.

We introduce a Gaussian kernel to construct the kernel matrix [13] and obtain the kernel matrix K_v for each view. Then we construct the average kernel $\overline{K}$ of all views. Overall, we construct a similarity matrix from the kernel matrix to measure the similarities among nodes, including neighbor searching [30], kernel construction, and normalization. This approach maintains local block diagonalization and encompasses the global data structure. The kernel function is listed below:

$$\begin{aligned} \overline{K}^*(i,j) &= \begin{cases} \overline{K}(i,j), & \text{select top } m\% \text{ neighbors}, \\ 0, & \text{otherwise}, \end{cases} \\ S_v^{(1)}(i,j) &= \begin{cases} K_v(i,j), & \overline{K}^*(i,j) > 0, \\ 0, & \overline{K}^*(i,j) = 0, \end{cases} \end{aligned} \tag{1}$$

where i and j are samples. By selecting the top $m\%$ of neighbor nodes with the highest weight, we can get the sparse kernel matrix $\overline{K}^*$. Finally, the similarity matrix $S_v^{(1)}$ is obtained.

To capture hidden similarity among nodes, we use second-order proximity [31], which shows its superior performance to the first order. According to the definition of the second-order proximity computation in [31], the similarity and degree matrices are listed below:

$$\begin{aligned} S_v^{(2)} &= S_v^{(1)} S_v^{(1)}, \\ D_v^{(2)}(i,j) &= \begin{cases} \sum_{q=1}^{n} S_v^{(2)}(i,q), & i = j, \\ 0. & i \neq j. \end{cases} \end{aligned} \tag{2}$$

3.2 Noise-Free Representation

Immediately following the previous subsection, we take the composed higher-order similarity matrix and use it to construct the corresponding Laplacian matrix. The significant cluster information of the shifted Laplacian matrix is embedded in its largest k eigenvectors [6]. To learn noise-free representation matrices of all views, we obtain the representation matrices from the shifted Laplacian matrices.

Individual views inherently contain noisy information. To eliminate the noise, we use the most informative pair of features:

$$\begin{aligned} L_v^{(2)} &= I_n + (D_v^{(2)})^{-1/2} S_v^{(2)} (D_v^{(2)})^{-1/2}, \\ L_v^{(2)} &= U_v \Sigma_v U_v^\top \\ &= U_v^r \Sigma_v^r (U_v^r)^\top + U_v^{r\perp} \Sigma_v^{r\perp} (U_v^{r\perp})^\top \\ &= L_v^r + L_v^{r\perp}, \end{aligned} \tag{3}$$

where $\Sigma_v^r = diag(\lambda_1^v, \ldots, \lambda_r^v)$ consists of the r largest eigenvalues and U_v^r contains the corresponding r eigenvectors in its columns and where $D_v^{(2)}$ and $S_v^{(2)}$ can be obtained from Equation (2). Similarly, $\Sigma_v^{r\perp}$ and $(U_v^{r\perp})^\top$ are the remaining eigenvalues and eigenvectors. Therefore, the the clustering information is stored by U_v^r. Choosing the rank r to be greater than k allows extra information from each shifted Laplacian at the initial stage. Then the eigenspace will be constructed by storing $r \geq k$ eigenpairs.

Therefore, the eigenspace constructed from the r largest eigenpairs may retain better clustering information. The eigenspace is given by Equation (4), where $\mathcal{C}(B)$ denotes the column space of matrix B. To compute the eigenspace, we first construct a sufficient basis to generate a subspace. Since $\mathcal{J}_p^r$ is the union of p subspaces, its basis is constructed iteratively in p steps. The initial basis U_1 is given by $U_1 = U_1^r$, which spans the subspace $\mathcal{C}(U_v^r))$.

$$\mathcal{J}_p^r = span(\bigcup_{v=1}^{p} \mathcal{C}(U_v^r)). \tag{4}$$

Υ_{v+1} is obtained from U_v^r by calculating the projection transformation, the residual transformation and the Gram-Schmidt orthogonalization.

$$U_{v+1} = [U_v \ \Upsilon_{v+1}]. \tag{5}$$

Let Υ_1=U_1. After p steps, the basis U_p is given by $U_p = [\Upsilon_1 \ \dots \ \Upsilon_p]$. We define a matrix H_v for each view. Combining with Equation (3), Equation (5) and Equation (8), it can be deduced that:

$$H_v = [\Upsilon_1 \ \dots \ \Upsilon_p]^\top U_v^r \Sigma_v^r (U_v^r)^\top [\Upsilon_1 \ \dots \ \Upsilon_p]. \tag{6}$$

R_v and Π_v are the eigenvectors and eigenvalues of H_v, respectively.

$$H_v = R_v \Pi_v R_v^\top. \tag{7}$$

These two bases F_v and U_p can be obtained from each other by a rotation matrix since they come from the same subspace $\mathcal{J}_p^r$.

$$F_v = U_p R_v, \tag{8}$$

where R_v is an orthogonal rotation matrix which can be obtained in Equation (7).

3.3 Uniform Framework

To better formulate the objective function of our method, we design an optimization framework for multi-view joint spectral embedding and spectral rotation [28].

First, a uniform indicating matrix F^* integrates the noise-free information from each individual view of F_v, avoiding the noise and redundancy that occurs when fusing information at the similarity level, as follows:

$$\min_{F^*} \sum_{v=1}^{p} \|F_v - F^*\|_F^2, \text{s.t.} \quad (F^*)^\top F^* = I_k. \tag{9}$$

Since the predefined weights may not be suitable for the new datasets, the weights of the view are to be adapted adaptively to enhance the robustness. To balance the weight relationship between the views, we combine different matrices with weight $\alpha_v (v = 1, 2, ..., p)$ and we introduce a regularization term L_2 with weight coefficients in the target function so that the view weights are continuous, thus improving the generalization ability of the algorithm and allowing it to better adapt to new data. That is, the higher the view importance, the higher the weight. Despite the low influence of views on clustering but still retains all features by reducing their weights in a way which is beneficial to all the features of the views are considered:

$$\begin{aligned} \min_{F^*, \alpha} \sum_{v=1}^{p} \|\alpha_v F_v - F^*\|_F^2 + \frac{\beta}{2} \|\alpha\|_2^2, \text{s.t.} \quad & (F^*)^\top F^* = I_k, \\ & \alpha^\top 1 = 1, \ \alpha_v \geq 0. \end{aligned} \tag{10}$$

Spectral rotation enables direct end-to-end output of clustering results without the need for subsequent discretization of the consensus representation. It can be implemented as follows:

$$\begin{aligned} \min_{F^*,R,Y,\alpha} \quad & \sum_{v=1}^{p} \|\alpha_v F_v - F^*\|_F^2 + \frac{\beta}{2}\|\alpha\|_2^2 \\ & + \sum_{i=1}^{n}\sum_{c=1}^{k} (Y_{i,c})^{\gamma} \|t_c - F^*_{i,:}R\|_2^2, \\ \text{s.t.} \quad & R^\top R = I_k, (F^*)^\top F^* = I_k, \\ & \alpha^\top 1 = 1, \alpha_v \geq 0, \end{aligned} \tag{11}$$

where $Y_{i,c}$ signifies the probability of the ith sample belonging to the cth cluster, R establishes the rational interactions between Y and F^*. $F^*_{i,:}$ is the ith row of F^*. t_c is a $1 * k$ dimensional vector that distinguishes each cluster. Specifically, t_c is a vector where the cth element is 1 and all other elements are 0, with $c \in 1, 2, \ldots, k$. In other words, matrix R captures the unique clustering pattern of F^*. If the representation of the ith sample, $Fi, :^*$, shows a distinct structure at the cth position after rotation, the label matrix Y will have a relatively high probability value at the corresponding position of the ith row and cth column. In other words, the label matrix Y emphasizes the specific clustering arrangement observed in the rotated sample representation F^*. Different clustering structures in multiple views are combined through a convex combination $\alpha = [\alpha_1, \ldots, \alpha_p]$.

3.4 Optimization

Equation (11) contains four variables α, F^*, R and Y to be optimized. Specifically, the updated strategies are presented in the subsections.

Update α and fix others. The objective becomes the following equation:

$$\begin{aligned} \min_{\alpha} \quad & \sum_{v=1}^{p} \|\alpha_v F_v - F^*\|_F^2 + \frac{\beta}{2}\|\alpha\|_2^2, \\ \text{s.t.} \quad & \alpha \geq 0,\ \alpha^\top 1 = 1. \end{aligned} \tag{12}$$

It is not difficult to transform the formula further into:

$$\min_{\alpha} \alpha^\top \alpha - \lambda^\top \alpha + \frac{\beta}{2}\|\alpha\|_2^2, \tag{13}$$

where $\lambda_v = \mathrm{Tr}(F_v^\top F^* + (F^*)^\top F_v)$ and $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_p]$.

Equation (13) can be optimized by the Lagrange Multiplier method [2].

$$L(\alpha, \rho, \sigma) = \alpha^\top \alpha - \lambda^\top \alpha + \frac{\beta}{2}\|\alpha\|_2^2 - \rho^\top \alpha - \sigma(\alpha^\top 1 - 1), \tag{14}$$

where $\rho = [\rho_1, \rho_2, \ldots, \rho_p] \geq 0$ and $\sigma \geq 0$ are the lagrangian multipliers. The optimal solution α^* satisfies the KKT condition:

$$\begin{aligned} &\frac{\partial L}{\partial \alpha}(\alpha^*, \rho, \sigma) = (2+\beta)\alpha^* - \lambda - \rho - \sigma 1 = 0, \\ &\text{s.t.} \quad \alpha^* \geq 0, {\alpha^*}^\top 1 - 1 = 0, \rho \geq 0, \rho_v \alpha_v^* = 0. \end{aligned} \tag{15}$$

Therefore, we can derive the ultimate solution as:

$$\alpha_v = \frac{\rho_v + \sigma + \lambda_v}{2+\beta}, \tag{16}$$

where $v = 1, 2, \ldots, p$.

When $\sigma + \lambda_v > 0$, since $\rho_v \geq 0$, we get $\alpha_v > 0$. From the condition $\rho_v \alpha_v^* = 0$, it can be obtained that $\rho_v = 0$. Then, $\alpha_v = \frac{\sigma+\lambda_v}{\beta}$.

Also, when $\sigma + \lambda_v = 0$, $\alpha_v = \frac{\sigma}{\beta}$, from the condition $\rho_v \alpha_v^* = 0$, we infer that $\rho_v = 0, \alpha_v = 0$.

When $\sigma + \lambda_v \leq 0$, if $\alpha_v \geq 0$, then $\sigma \geq 0$, that is inconsistent when $\rho_v \alpha_v^* = 0$. Thus, we get $\alpha_v = 0$. Therefore, with λ increasing, we can find the positive integer Q = argmax$(\sigma + \lambda_v > 0)$. The optimality conditions are summarized as follows:

$$\alpha_v = \begin{cases} \frac{\sigma+\lambda_v}{2+\beta}, & v \leq Q \\ 0. & v > Q. \end{cases} \tag{17}$$

From ${\alpha^*}^\top 1 = 1$, we can get:

$$\sigma = \frac{\sum_{v=1}^{Q} \alpha_v + 2 + \beta}{p}. \tag{18}$$

Update F^* and fix others. In this situation, the objective becomes:

$$\begin{aligned} &\min_{F^*} \quad \sum_{v=1}^{p} \|\alpha_v F_v - F^*\|_F^2 + \sum_{i=1}^{n} \sum_{c=1}^{k} (Y_{i,c})^\gamma \|t_c - F_{i,:}^* R\|_2^2, \\ &\text{s.t.} \quad (F^*)^\top F^* = I_k. \end{aligned} \tag{19}$$

We first introduce Proposition 1 [28]. According to Proposition 1, Equation (19) can be simplified to Equation (20), as follows:

$$\begin{aligned} &\max \operatorname{Tr}((F^*)^\top N), \\ &\text{s.t.} (F^*)^\top F^* = I_k, \end{aligned} \tag{20}$$

where $N = 2\sum_{v=1}^{p} F^v + GR^\top$ and $G_{i,:} = \sum_{c=1}^{k} (Y_{i,c})^\gamma t_c$. Equation (20) can be solved by proposition 2 [28]. In Proposition 2, $F^* = U_1 V_1^\top$, where U_1 and V_1 are the left and right singular matrices of matrix N, respectively.

Update R and fix others. In this situation, the objective becomes:

$$\begin{aligned} \min_{R} \ & \sum_{i=1}^{n}\sum_{c=1}^{k}(Y_{i,c})^{\gamma}\|t_c - F^*_{i,:}R\|_2^2, \\ \text{s.t.} \ & R^\top R = I_k. \end{aligned} \tag{21}$$

According to Proposition 1 and Proposition 2 [28], the optimal solution $R = U_2V_2^\top$, where U_2 and V_2 are left singular matrix and right singular matrix of $(F^*)^\top G$ respectively.

Update Y and fix others. In this situation, the objective becomes:

$$\begin{aligned} \min_{Y} \ & \sum_{c=1}^{k}(Y_{i,c})^{\gamma}\|t_c - F^*_{i,:}R\|_2^2, \\ \text{s.t.} \ & Y_{i,c} \geq 0, \ Y_{i,:}1_k = 1. \end{aligned} \tag{22}$$

We can get the final result by [28]. The specific formula is as follows:

$$Y_{i,c} = \frac{(P_{i,c})^{\frac{1}{1-\gamma}}}{\sum_{c=1}^{k}(P_{i,c})^{\frac{1}{1-\gamma}}}, \tag{23}$$

where $P_{i,c} = ||\mathrm{t_c} - \mathrm{F^*_{i,:}R}||_2^2$.

Algorithm 1 UFMKSC

Input: The number ofclusters k. Data points in v views $\{X_v\}_{v=1}^{p}$.
Parameter: Hyper-parameters β=1, γ=1, ratio of neighbor m and rank $r \geq k$.
Output: The indicator matrix Y

```
Calculate high-order similarity matrices via Equation (2).
Calculate F_v by using Equation (8).
Initialize F*, R, Y, α.
while convergence criterion is not satisfied do
    Update α by solving Equation (12).
    Update F* by solving Equation (19).
    Update R by solving Equation (21).
    Update Y by solving Equation (22).
end while
return indicator matrix Y.
```

3.5 Clustering Algorithm

Based on the above optimal process, we present the details of Algorithm 1, which consists of an initialization procedure and four-step alternative optimization procedures. The convergence criterion refers to the condition where the algorithm

terminates and reaches a converged state when the change in the result labels Y becomes sufficiently small and stable.

During the initialization procedure, calculating the cluster indicator matrices F_v involves performing an eigen-decomposition of the Laplacian matrices, which has a complexity of $\mathcal{O}(pn^3)$. In the alternative optimization steps, updating α, F^*, R and Y which take $\mathcal{O}(p^2)$, $\mathcal{O}(n^3)$, $\mathcal{O}(pk^3)$ and $\mathcal{O}(pnk^2 + nk)$ time, respectively. Overall, the time complexity of Algorithm 1 is $\mathcal{O}(pn^3 + p^2 + n^3 + pk^3 + pnk^2 + nk)$. Since $p \ll n$ and $k \ll n$, algorithm can be simplified to $\mathcal{O}(n^3)$. In the comparative experimental methods, the time complexity for the majority of the algorithms is $\mathcal{O}(n^3)$.

4 Experiments

4.1 Datasets and Baselines

Table 2 lists six datasets, including `Yale`, `Washington`, `Wisconsin`, `Synthetic3d`, `WebKB` and `Wiki`. The proposed algorithm is compared with several baselines on the standard real-world datasets.

Table 2: Information of datasets

Dataset	Samples	Views	Clusters
Yale	165	5	15
Washington	230	2	5
Wisconsin	265	2	5
Synthetic3d	600	3	3
WebKB	1051	2	2
Wiki	2866	2	10

The following nine state-of-the-art methods are compared with our method:

- `ONMSC` [31]: It searched for the neighborhoods of linear combinations of first-order and higher-order basis Laplacian matrices.
- `DMKKM` [25]: It improved the discrete multi-kernel k-means model of the clustering metrics matrix.
- `COMVSC` [28]: A multi-view subspace clustering algorithm was proposed, integrating similarity learning, clustering partitioning, and classification label unification.
- `COALA` [6]: This method utilized feature vector obtained from the Laplacian matrix decomposition to express the potential subspace.
- `ONMSC-LF` [10]: It learned a high-order neighborhood Laplacian matrix and used a partitioned representation of each view common to fusing multi-view information.

Table 3: Performance comparison of twelve algorithms on six MVC datasets.

Datasets	ONMSC	DMKKM	COMVSC	COALA	ONMSC-LF	ON-ALK	GLSEF	FgMVC	EMVCKM	Proposed
ACC(%)										
Yale	64.85	64.24	60.00	63.64	66.06	63.03	*68.09*	60.61	67.88	**68.49**
Washington	51.30	61.30	65.65	64.78	54.35	60.87	46.96	*70.12*	62.17	**73.48**
Wisconsin	57.36	59.26	53.96	55.09	55.09	*63.77*	48.68	58.87	46.04	**71.32**
Synthetic3d	96.33	91.00	92.83	96.83	96.17	96.33	91.67	*97.53*	95.96	**97.67**
WebKB	80.59	58.90	77.74	*88.20*	76.12	78.21	65.94	85.23	87.56	**88.68**
Wiki	57.50	51.43	53.14	53.84	57.47	*61.86*	52.02	53.77	53.08	**64.13**
NMI(%)										
Yale	64.40	63.50	58.46	*68.19*	64.46	61.99	66.48	59.99	**77.50**	66.83
Washington	33.33	37.99	30.95	24.50	34.10	35.94	27.68	*44.65*	41.62	**45.62**
Wisconsin	32.17	35.65	22.55	12.27	34.00	38.86	24.32	*43.22*	36.96	**44.32**
Synthetic3d	85.39	72.11	75.50	86.53	85.01	85.49	83.23	*89.08*	84.16	**89.52**
WebKB	18.21	20.70	25.36	*37.84*	23.68	13.00	35.87	37.06	35.44	**37.86**
Wiki	51.47	42.65	51.23	52.55	52.13	*54.60*	50.04	52.92	50.07	**55.99**
Purity(%)										
Yale	65.45	64.24	60.61	63.64	66.67	63.64	*68.97*	61.21	67.88	**69.09**
Washington	69.13	73.04	68.70	65.22	69.13	71.30	49.57	70.12	*73.48*	**74.78**
Wisconsin	69.81	*73.21*	64.15	56.98	70.94	71.32	52.83	58.88	46.04	**73.96**
Synthetic3d	96.33	91.00	92.83	96.83	96.17	96.33	96.83	*97.53*	95.96	**97.67**
WebKB	80.59	78.12	78.12	*88.20*	78.12	78.21	71.46	85.23	87.56	**88.68**
Wiki	60.75	52.83	53.24	61.27	61.30	*63.85*	62.14	60.12	58.28	**64.17**

The best is bolded, and the second best is underlined in italics.

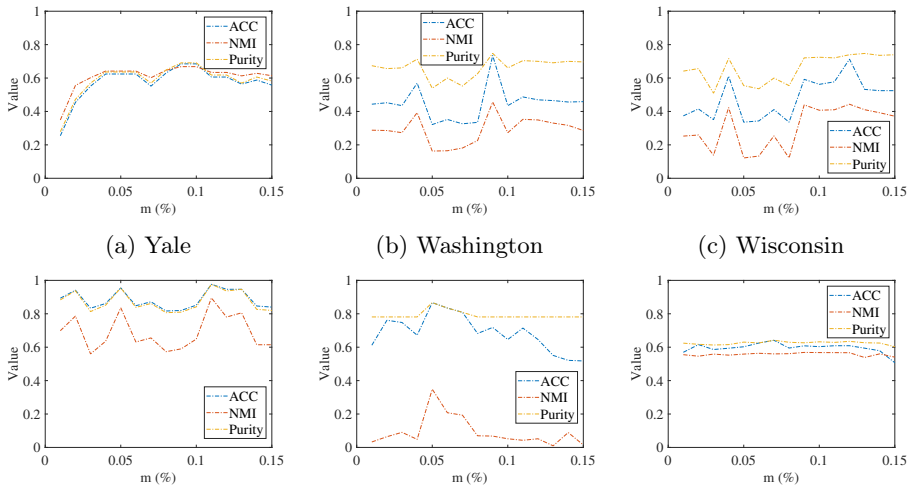
- `ON-ALK` [12]: It enhanced the local density around data samples to strengthen the ability of the kernel matrix to express intra-class information.
- `GLSEF` [24]: The diversity of features and the Grassmann manifold were considered to maintain spatial and topological information.
- `FgMVC` [27]: It utilized a multi-prototype representation to partition the sample space of each view into multiple subclusters.
- `EMVCKM` [14]: This method transformed the adjacency matrix into a distance matrix using a filter function.

In our experiments, we utilize open-source code provided by the authors, conducting 10 trials and reporting the best results obtained. All experiments are performed on a desktop computer equipped with an `Intel Core i7 CPU@3.6GHz` and `16GB RAM`, using `MATLAB` 2022a (64-bit).

4.2 Performance Evaluation

The results of three widely used clustering metrics are presented in Table 3, which include *Accuracy* (ACC), *Normalized Mutual Information* (NMI), and *Purity*. We mark the optimal results in bold and use italic underline for the suboptimal results.

According to the presented results, our proposed UFMKSC achieves the best clustering performance on six datasets, demonstrating its excellent performance. The obvious advantages of ACC on the six data sets are 0.40%, 3.36%, 7.55%, 0.14%, 0.48%, and 2.27%. As for NMI, our method is only below the best method on Yale. Similarly, the metric purity exceeds with other algorithms. The above results proved the effectiveness of our proposed method and demonstrated that our UFMKSC method is a contributing multi-view spectral clustering method.



(a) Yale (b) Washington (c) Wisconsin

(d) Synthetic3d (e) WebKB (f) Wiki

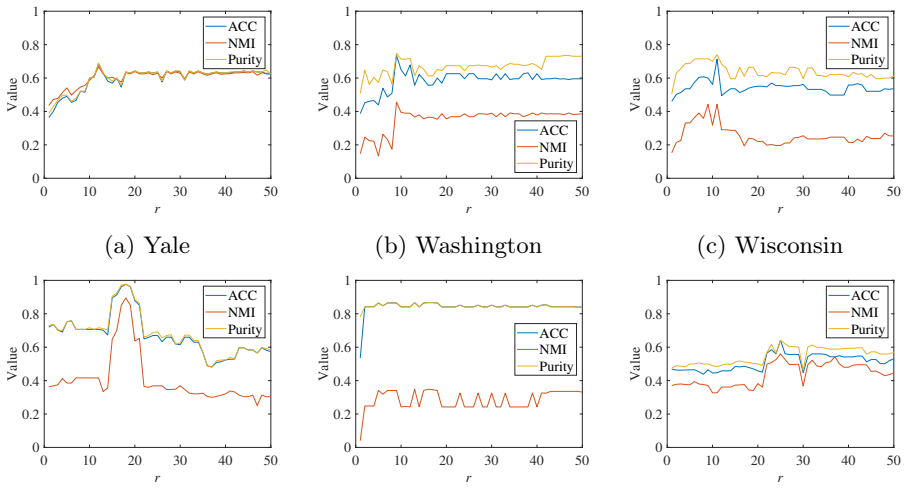
Fig. 1: The influences of ratios of neighbor nodes selected.

4.3 Parameter Sensitivity

We conduct sensitivity analysis on the hyper-parameters. The proposed UFMKSC introduces two hyper-parameters, i.e., the parameter m is to specify the ratios of neighbor nodes, the parameter r is used to store the eigenpairs from the Laplacian matrix.

The sensitivity of the ratios of neighbor nodes selected is reflected in Figure 1. The variation curves of the number of neighbors m versus ACC, NMI and purity are reported, respectively. The proposed method is sensitive to neighbor numbers. It can be noticed that in most of the datasets, the number of neighbors has a better effect in the middle segment of the region, which is caused by two reasons. On the one hand, if the number of neighbors is too small, information is not sufficiently utilized. On the other hand, if the number of neighbors is too large, it will cause more unnecessary information. In this paper, the choice of 0.04 or 0.05 can achieve better results.

The variation of ACC, NMI and purity metrics relative to rank r is shown in Figure 2 for different datasets. It can be observed that the ACC, NMI and purity values vary in similar magnitudes, and the degree of variation gradually becomes smaller or flattens out after reaching the maximum value. Most datasets have better results with intermediate rank values because the larger the rank, the more noise is introduced, and the smaller the rank, the less valid information is needed. The subspace constructed from rank r is distinct from the full-rank subspace, causing the evaluation metrics to fluctuate within a certain range. Therefore, it is more effective to choose a rank of about 10 and 20.



(a) Yale (b) Washington (c) Wisconsin

(d) Synthetic3d (e) WebKB (f) Wiki

Fig. 2: The influences of the rank r.

Table 4: Ablation study.

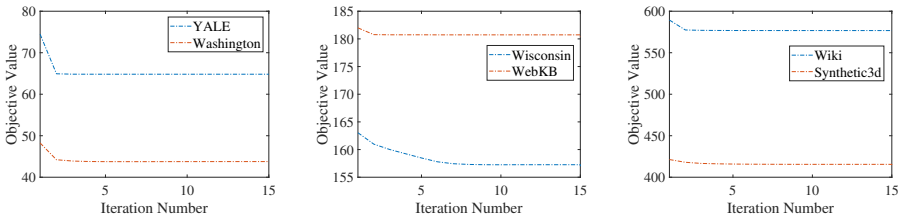
Dataset	w/o order	w/o subspace	w/o rotation	ACC (%)
Yale	64.85	64.24	61.81	**68.49**
Washington	73.04	68.26	63.61	**73.48**
Wisconsin	70.94	62.64	61.32	**71.32**
Synthetic3d	97.33	97.50	95.00	**97.67**
WebKB	81.63	84.97	85.35	**88.68**
Wiki	63.64	58.13	61.97	**64.13**

4.4 Ablation Study

We investigate the effectiveness of each component, i.e., higher-order connectivity information, low-rank subspace and spectral rotation, through an ablation study. The ACC performance of the dataset is presented in Table 4.

When no higher-order neighborhood information is available, first-order information is used to construct the similarity matrix in the experiment. When there is no subspace, the spectral embedding matrix is obtained by direct eigendecomposition of the Laplace matrix in the experiments. When there is no spectral rotation term, the clustering results are obtained by implementing the traditional $k$means method on the spectral embedding matrix in the experiments.

If the quality of the input similarity matrix is low, meaning it fails to accurately capture the true relationships between data points, it will hinder the ability of the subspace method to correctly identify the low-dimensional structure of the data. Furthermore, the spectral rotation operation may not be able



(a) (b) (c)

Fig. 3: Convergence curve of our proposed algorithm on datasets.

to effectively optimize the embedding matrix. The algorithm cannot obtain the global information structure if no higher-order information items are available. Low-rank subspace item prevents data from being mixed with unnecessary information. The discretization step involves transforming continuous embedding matrices into discrete clustering indicators. This transformation can lead to information loss. Additionally, the lack of mutual optimization between the discretization step and the spectral embedding step hinders their full synergy.

4.5 Convergence analysis

Convergence analysis is performed to verify the convergence of the proposed method. The optimization algorithm guarantees the final convergence because the objective function in Equation (11) decreases with iteration. The objective values are plotted as a function of the iteration steps in Figure 3. As shown in Figure 3, we observe that the objective value decreases rapidly during all six iterations of the benchmark data set. It can be observed that convergence can be reached within ten steps of the iteration.

5 Conclusion

This paper proposes a uniform framework for multiple kernel spectral clustering using a noise-free Laplacian matrix. To better explore the multi-view data, we transform the original data to high-order kernel data and introduce subspace learning to minimize noise. The objective function problem is then solved by an efficient iterative optimization algorithm that includes spectral rotation to obtain the global clustering result simultaneously. Experimental results on various datasets demonstrate the effectiveness of the proposed clustering method.

5.1 Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 62407016, and also supported by the Project of Guangdong Provincial Educational Science Planning in China under Grant No. 2024GXJK140. *Shun Mao* is the corresponding author of this paper.

References

1. Chao, G., Sun, S., Bi, J.: A survey on multiview clustering. IEEE Transactions on Artificial Intelligence **2**(2), 146–168 (2021)
2. Cheng, Q., Liu, C., Shen, J.: A new lagrange multiplier approach for gradient flows. Computer Methods in Applied Mechanics and Engineering **367**, 113070 (2020)
3. Fang, U., Li, M., Li, J., Gao, L., Jia, T., Zhang, Y.: A comprehensive survey on multi-view clustering. IEEE Trans. Knowl. Data Eng. **35**(12), 12350–12368 (2023)
4. Gao, H., Nie, F., Li, X., Huang, H.: Multi-view subspace clustering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4238–4246 (2015)
5. Gönen, M., Alpaydın, E.: Localized algorithms for multiple kernel learning. Pattern Recognition **46**(3), 795–807 (2013)
6. Khan, A., Maji, P.: Approximate graph laplacians for multimodal data clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(3), 798–813 (2021)
7. Khan, A., Maji, P.: Multi-manifold optimization for multi-view subspace clustering. IEEE Transactions on Neural Networks and Learning Systems **33**, 3895–3907 (2022)
8. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: state of the art, current trends and challenges. Multimedia tools and applications **82**(3), 3713–3744 (2023)
9. Li, X., Zhang, H., Wang, R., Nie, F.: Multiview clustering: A scalable and parameter-free bipartite graph fusion method. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**, 330–344 (2022)
10. Liang, W., Zhou, S., Xiong, J., Liu, X., Wang, S., Zhu, E., Cai, Z., Xu, X.: Multi-view spectral clustering with high-order optimal neighborhood laplacian matrix. IEEE Transactions on Knowledge and Data Engineering **34**(7), 3418–3430 (2022)
11. Lin, R., Tang, F., He, C., Wu, Z., Yuan, C., Tang, Y.: DIRS-KG: a kg-enhanced interactive recommender system based on deep reinforcement learning. World Wide Web (WWW) **26**(5), 2471–2493 (2023)
12. Liu, J., Liu, X., Xiong, J., Liao, Q., Zhou, S., Wang, S., Yang, Y.: Optimal neighborhood multiple kernel clustering with adaptive local kernels. IEEE Transactions on Knowledge and Data Engineering **34**(6), 2872–2885 (2022)
13. Liu, X., Zhu, X., Li, M., Wang, L., Zhu, E., Liu, T., Kloft, M., Shen, D., Yin, J., Gao, W.: Multiple kernel k k-means with incomplete kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(05), 1191–1204 (2020)
14. Lu, H., Xu, H., Wang, Q., Gao, Q., Yang, M., Gao, X.: Efficient multi-view k-means for image clustering. IEEE Transactions on Image Processing **33**, 273–284 (2024)
15. Ma, J., Wang, R., Ji, W., Zhao, J., Zong, M., Gilman, A.: Robust multi-view continuous subspace clustering. Pattern Recognition Letters **150**, 306–312 (2021)
16. Mao, S., Zhan, J., Li, J., Jiang, Y.: Knowledge structure-aware graph-attention networks for knowledge tracing. In: International Conference on Knowledge Science, Engineering and Management. pp. 309–321. Springer (2022)
17. Mao, S., Zhan, J., Wang, Y., Jiang, Y.: Improving knowledge tracing via considering two types of actual differences from exercises and prior knowledge. IEEE Transactions on Learning Technologies **16**(3), 324–338 (2023)
18. Marin, D., Tang, M., Ayed, I.B., Boykov, Y.: Kernel clustering: Density biases and solutions. IEEE Trans. Pattern Anal. Mach. Intell. **41**(1), 136–147 (2019)
19. Mijangos, V., Sierra, G., Montes, A.: Sentence level matrix representation for document spectral clustering. Pattern Recognition Letters **85**, 29–34 (2017)

20. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems **14** (2001)
21. Peng, X., Feng, J., Zhou, J.T., Lei, Y., Yan, S.: Deep subspace clustering. IEEE Trans. Neural Networks Learn. Syst. **31**(12), 5509–5521 (2020)
22. Sun, M., Zhang, P., Wang, S., Zhou, S., Tu, W., Liu, X., Zhu, E., Wang, C.: Scalable multi-view subspace clustering with unified anchors. In: Proceedings of the ACM International Conference on Multimedia. p. 3528–3536 (2021)
23. Wang, L., Xing, J., Yin, M., Huang, X.: Multi-view spectral clustering with adaptive local neighbors. In: Proceedings of the International Symposium on Parallel Architectures, Algorithms and Programming. pp. 157–161 (2021)
24. Wang, P., Wu, D., Wang, R., Nie, F.: Multi-view graph clustering via efficient global-local spectral embedding fusion. In: Proceedings of the ACM International Conference on Multimedia. pp. 3268–3276 (2023)
25. Wang, R., Lu, J., Lu, Y., Nie, F., Li, X.: Discrete multiple kernel k-means. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 3111–3117 (2021)
26. Wu, J., Lin, Z., Zha, H.: Essential tensor learning for multi-view spectral clustering. IEEE Transactions on Image Processing **28**(12), 5910–5922 (2019)
27. Yin, H., Wang, G., Hu, W., Zhang, Z.: Fine-grained multi-view clustering with robust multi-prototypes representation. Applied Intelligence **53**(7), 8402–8420 (2023)
28. Zhang, P., Liu, X., Xiong, J., Zhou, S., Zhao, W., Zhu, E., Cai, Z.: Consensus one-step multi-view subspace clustering. IEEE Transactions on Knowledge and Data Engineering **34**(10), 4676–4689 (2022)
29. Zhou, B., Liu, W., Shen, M., Lu, Z., Zhang, W., Zhang, L.: Adaptive graph fusion learning for multi-view spectral clustering. Pattern Recognition Letters **176**, 102–108 (2023)
30. Zhou, S., Liu, X., Li, M., Zhu, E., Liu, L., Zhang, C., Yin, J.: Multiple kernel clustering with neighbor-kernel subspace segmentation. IEEE Transactions on Neural Networks and Learning Systems **31**(4), 1351–1362 (2020)
31. Zhou, S., Liu, X., Liu, J., Guo, X., Zhao, Y., Zhu, E., Zhai, Y., Yin, J., Gao, W.: Multi-view spectral clustering with optimal neighborhood laplacian matrix. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 6965–6972. AAAI Press (2020)