Approximate Shifted Laplacian Reconstruction for Multiple Kernel Clustering

Jiali You
Southwest University of Science and Technology
Mianyang, China
yjl1015004@163.com

Zhenwen Ren*
Southwest University of Science and Technology
Mianyang, China
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
Shenzhen, China
Nanjing University
Nanjing, China
rzw@njust.edu.cn

Quansen Sun
Nanjing University of Science and Technology
Nanjing, China
sunquansen@njust.edu.cn

Yuan Sun
Sichuan University
Chengdu, China
sunyuan_work@163.com

Xingfeng Li
Nanjing University of Science and Technology
Nanjing, China
lixingfeng@njust.edu.cn

ABSTRACT

Multiple kernel clustering (MKC) has demonstrated promising performance for handing non-linear data clustering. Positively, it can integrate complementary information of multiple base kernels and avoid kernel function selection. However, negatively, the main challenging is that the kernel matrix with the size $n \times n$ leads to $O(n^2)$ memory complexity and $O(n^3)$ computational complexity. To mitigate such a challenging, taking graph Laplacian as breakthrough, this paper proposes a novel and simple MKC method, dubbed as approximate shifted Laplacian reconstruction (ASLR). For each base kernel, we propose the r-rank shifted Laplacian reconstruction scheme by considering the energy losing of Laplacian reconstruction and the clustering information preserving of Laplacian decompose simultaneously. Then, by analyzing the eigenvectors of the reconstructed Laplacian, we impose some constrains to tame its solution within a Fantope. Accordingly, the byproduct (*i.e.*, the most informative eigenvectors) contains the main clustering information, such that the clustering assignments can be obtained relying on simple k-means algorithm. Owe to the Laplacian reconstruction scheme, the memory and computational complexity can be reduced to $O(n)$ and $O(n^2)$, respectively. As experimentally demonstrated on eight challenging MKC benchmark datasets, the results verify the effectiveness and efficiency of ASLR.

*Corresponding author.

MM '22, October 10–14, 2022, Lisbon, Portugal.

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**;

KEYWORDS

Clustering, multiple kernel learning, Shifted Laplacian

ACM Reference Format:
Jiali You, Zhenwen Ren, Quansen Sun, Yuan Sun, and Xingfeng Li. 2022. Approximate Shifted Laplacian Reconstruction for Multiple Kernel Clustering. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisbon, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Clustering aims at partitioning data points into different clusters according to some similarity measures, such that ones falling in the same cluster are similar to each other and dissimilar to those of other clusters [5]. Although linear clustering methods have achieved impressive performances, the global linearity assumption is somewhat strong, such that they often fail in dealing with nonlinear data. It is widely known that single kernel clustering (SKC) methods are commonly used to explore structure of nonlinear data. Usually, they convert data matrix $\mathbf{X}$ into kernel matrix $\mathbf{K}$, by mapping data points from the original feature space $\mathbb{R}$ to a reproducing Hilbert space $\mathbb{H}$. However, the most suitable kernel function and parameters for a specific dataset are difficult to determine in advance, especially in real-life applications.

Regarding the problem mentioned above, multiple kernel clustering (MKC) has demonstrated its promising performance in recently years. As a convention, it needs to build a kernel pool consisting of multiple base kernels. Without loss of generality, these kernels can be different types of kernels, such as linear kernel, polynomial kernel and Gaussian kernel, *etc*, or different parameter kernels, such as the Gaussian kernels with bandwidth 0.1, 1 and 10, *etc*. Overall, these existing methods can roughly be divided into two categories,

including kernel k-means (KKM) ones and spectral clustering (SC) ones. Along the KKM line, these works [4, 11] usually learn a consensus kernel matrix by integrating these candidate base kernels, and then perform KKM algorithm to obtain the clustering assignments. In addition, along the SC line, these works [16, 19] transform each kernel to an affinity graph (or Laplacian matrix) via many graph learning methods, such as kernelized subspace learning, adaptive local structure learning, and non-negative matrix factorization, *etc.* And then, the SC algorithm is employed to obtain clustering results. However, most of the above methods have to face $O(n^2)$ memory complexity and $O(n^3)$ computational complexity, due to the kernel matrix with the size $n \times n$.

For handling median or large-scale MKC tasks, late-fusion learning is a widely used paradigm [8, 10, 20], which adopts a "two-stage" learning scheme. It firstly performs tiny singular value decomposition to output a lower-dimensional partition matrix for each base kernel, and then fuses these partitions to obtain a resulting partition. Essentially, the first step is a dimensionality reduction preprocessing, which reduces the kernel matrix from $n \times n$ to $m \times n, m \ll n$. Taking the partition matrixes as input, recently, some works [18, 25] introduce a dynamic anchor sampling strategy to reduce the sample size. By observing that the above-mentioned "two-stage" paradigm disconnects the processes of information compression and clustering representational learning. That is, they do not simultaneously consider the energy losing and the clustering information preserving, at the first step. From another point of view, each entry of a kernel matrix (*e.g.*, $\mathbf{k}_{ij}$) stands for the similarity between a pair of points (*i.e.*, $\mathbf{x}_i$ and $\mathbf{x}_j$). However, these late-fusion methods simply treat kernel matrix as plain data, such that the abundant graph information hidden in kernel matrix is ignored.

To address the above issues, in this paper, we propose a novel and simple method for MKC tasks, namely *approximate shifted Laplacian reconstruction (ASLR)*. Specifically, due to the complete theoretical characteristics of spectral graph, we treat each kernel matrix as affinity graph rather than plain data, and then construct its corresponding normalized graph Laplacian and candidate r-rank shifted Laplacian in advance. Accordingly, we reconstruct the approximate shifted Laplacian to exploit the comprehensive information of all the candidate r-rank shifted Laplacians. After that, we project the learned shifted Laplacian onto Fantope to encourage the most informative eigenvalues of the learned shifted Laplacian holding more prominent cluster characteristics. As a result, the most informative vectors are fed into k-means algorithm to obtain the resulting cluster assignments. In summary, the contributions of this paper are:

- Taking graph Laplacian as breakthrough, this paper proposes to transform each base kernel matrix into a r-rank shifted Laplacian matrix, such that the memory and computational complexity of the proposed ASLR can be reduced to $O(n)$ and $O(n^2)$, respectively.
- This paper proposes an approximate shifted Laplacian reconstructing scheme, which can integrate the main energy and clustering information of each candidate r-rank shifted Laplacian simultaneously. Therefore, the clustering performance is largely improved.
- This paper develops Fantope projection to encourage the most informative eigenvalues of learned shifted Laplacian to hold more prominent cluster characteristics.
- Compared with many state-of-the-art MKC methods in term of running time and clustering performance, the superiority of the proposed ASLR is demonstrated by conducting some extensive experiments.

2 RELATED WORKS

2.1 Spectral Graph Theory

For a data matrix $\mathbf{X}$, the k-nearest affinity matrix is denoted as $\mathbf{S} \in \mathbb{R}^{n \times n}$, and the edge between each sample-pair (*e.g.*, $\mathbf{x}_i$ and $\mathbf{x}_j$) is typically defined by a Gaussian kernel, *i.e.*,

$$\mathbf{s}_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right) & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where σ is the kernel bandwidth parameter and k is the number of neighbors for each vertex. Thereupon, the degree matrix is given by $\mathbf{D} = \mathtt{diag}(\tilde{d}_1, \ldots, \tilde{d}_i, \ldots, \tilde{d}_n)$, where $\tilde{d}_i = \sum_{j=1}^{n} \mathbf{s}_{ij}$, the Laplacian matrix of $\mathcal{G}$ is denoted as

$$\mathbf{L} = \mathbf{D} - \mathbf{S} \tag{2}$$

and the normalized Laplacian of $\mathcal{G}$ is denoted as

$$\mathbf{L}_\mathbf{N} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{S})\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2} \tag{3}$$

Accordingly, the eigenvalues of $\mathbf{L}_\mathbf{N}$ is given by $0 = \delta_1 \leq \delta_2 \leq \cdots \leq \delta_n$, and the $\{\delta_i\}_{i=1}^n$ is usually called the spectrum of $\mathbf{L}_\mathbf{N}$. As we known, the problem of spectral clustering (also known as normalized cut) can be formulated as

$$\min_{\mathbf{F} \in \mathbb{R}^{n \times c}} \mathtt{Tr}\left(\mathbf{F}^\top \mathbf{L}_\mathbf{N} \mathbf{F}\right) \quad \mathtt{s.t.}\ \mathbf{F}^\top \mathbf{F} = \mathbf{I} \tag{4}$$

where the optimal clustering indictor matrix $\mathbf{F}$ can be obtained by extracting the c eigenvectors of $\mathbf{L}_\mathbf{N}$ relating to its c smallest eigenvalues (*i.e.*, $\delta_1, \delta_2, \cdots, \delta_c$).

2.2 Laplacian Reconstruction

To fully explore the complementary information of different graphs, linearly combines the candidate Laplacians and learns an optimal one which can best suit for clustering [23]. Zhou *et al.* propose to search the optimal Laplacian from the neighborhood of candidate Laplacians [24]. Moreover, the weight values are guided by a predefined affinity matrix. Mathematically, it is given by

$$\begin{aligned} \min_{\mathbf{F}^\top \mathbf{F} = \mathbf{I}_c, \mathbf{g}, \mathbf{W}, \mathbf{\Lambda}} & \mathtt{Tr}\left(\mathbf{F}^\top \left(\mathbf{I}_n - \mathbf{W}\mathbf{\Lambda}\mathbf{W}^\top\right)\mathbf{F}\right) \\ & + \sum_{o=1}^{O} \left\| \left(\mathbf{I}_n - \mathbf{W}\mathbf{\Lambda}\mathbf{W}^\top\right) - \mathbf{L}_\mathbf{g}^{(o)} \right\|_\mathrm{F}^2 + \alpha \mathbf{g}^\top \mathbf{M}\mathbf{g} \\ \mathtt{s.t.}\ & \mathbf{L}_\mathbf{g}^{(o)} = \sum_{i=1}^{m} \mathbf{g}_i \mathbf{L}_i^{(o)} (o \in [O]), \|\mathbf{g}\|_1 = 1, \\ & \mathbf{g} \geq 0, \mathbf{W} \in \mathbb{R}^{n \times c}, \mathbf{W}^\top \mathbf{W} = \mathbf{I}_c, 0 \leq \Lambda_{ii} \leq 1 \end{aligned} \tag{5}$$

where $\mathbf{W}\mathbf{\Lambda}\mathbf{W}^\top$ can be seen as the graph affinity matrix, $(\mathbf{I}_n - \mathbf{W}\mathbf{\Lambda}\mathbf{W}^\top)$ forces the learned Laplacian matrix $\mathbf{L}_\mathbf{g}$ to be symmetric and positive semi-definite (SPSD), $\mathbf{L}_i^{(o)}$ is the o-th ($o \in [O]$) Laplacian matrix, and

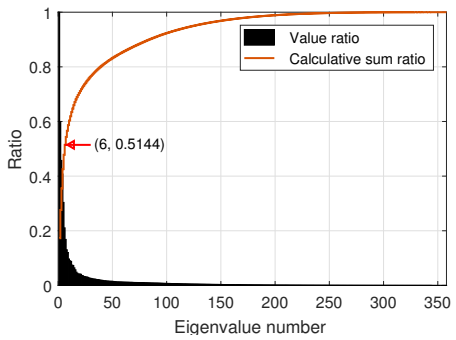


Figure 1: Eigen-value distribution of a symmetric positive semi-definite matrix (SPSD) matrix [9], *e.g.*, kernel matrix K. The eigen-values are sorted from large to small. Note here that the eigen-vector corresponding to a larger eigen-value carries more discriminative information.

$\mathbf{M}$ is the priori-knowledge matrix to guide learning the weighting $\mathbf{g}$.

3 PROPOSED APPROXIMATE SHIFTED LAPLACIAN RECONSTRUCTION (ASLR) METHOD

For MKC, given a set of n samples drawn from c crispy clusters, one can build a base kernel pool by employing multiple kernel functions [16]. From the graph theory, it is well understood that a kernel matrix can be deemed as the affinity graph. Mathematically, the i-th ($1 \le i \le m$) base kernel is defined as $\mathbf{K}^{(i)} \in \mathbb{R}^{n\times n}$ and its corresponding normalized Laplacian matrixes is given by $\mathbf{L}_{\mathrm{N}}^{(i)}$ accordingly.

3.1 Shifted Graph Laplacian

The major advantage of spectral clustering, *i.e.*, Eq. (4), is that it genteelly transforms the data points from the feature space to an indicator space where the cluster properties are more prominent. Note here that, in Laplacian matrix $\mathbf{L}_{\mathrm{N}}$, the necessary cluster information is embedded in its c smallest eigenvectors, whereas the best low-rank approximation of a SPSD matrix can be reconstructed relying on its few largest eigenvectors (see Fig. 1). Thus, the best low-rank approximation of $\mathbf{L}_{\mathrm{N}}$ primarily encodes noise, rather than cluster information. That is, the cluster information and Laplacian reconstructing are a contradiction. In other words, by minimizing $\|\mathbf{L} - \sum_{i=1}^{m} g_i \mathbf{L}_{\mathrm{N}}^{(i)}\|_F^2$, $\mathbf{L}$ can preserve the main energy of all $\{\mathbf{L}_{\mathrm{N}}^{(i)}\}_{i=1}^{m}$, but the cluster information is partly ignored.

Therefore, in order to preserve the cluster information in the low-rank approximation, we can transform the normalized Laplacian to shifted Laplacian [3, 7], formulated as

$$\mathbf{L}_{\mathrm{S}} = 2\mathbf{I} - \mathbf{L}_{\mathrm{N}} = \mathbf{I} + \mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2} \tag{6}$$

The following theorem makes $\mathbf{L}_{\mathrm{S}}$ feasible to reflect the cluster information and perform matrix reconstruction.

Theorem 1: [3] *If (δ, u) is an eigenvalue-eigenvector pair of normalized Laplacian $\mathbf{L}_{\mathrm{N}}$, then $(2-\delta, u)$ is an eigenvalue-eigenvector of shifted Laplacian $\mathbf{L}_{\mathrm{S}}$. And, $\mathbf{L}_{\mathrm{S}}$ is a symmetric and positive semi-definite matrix and its eigenvalues lie in $[0, 2]$.*

3.2 Approximate Shifted Laplacian Reconstruction

Theorem 1 implies that the c smallest eigenvalues of $\mathbf{L}_{\mathrm{N}}$ correspond to the c largest eigenvalues of $\mathbf{L}_{\mathrm{S}}$. Therefore, the solution $\mathbf{F}$ of Eq. (4) is formed by the c largest eigenvectors of $\mathbf{L}_{\mathrm{S}}$. That is, the best r-rank approximation of $\mathbf{L}_{\mathrm{S}}$ can encode its cluster information when performing Laplacian decompose; meanwhile it can hold the main energy when performing Laplacian reconstruction. To facilitate reading, we rewrite $\mathbf{L}_{\mathrm{S}}$ as $\mathbf{L}$ throughout this paper.

Accordingly, we treat each kernel matrix $\mathbf{K}^{(i)}$ as affinity graph $\mathbf{K}$ in Eq. (6), rather than plain data. Here, to remove the redundancy edges, it is customary to employ a parameter k to control the number of neighbors of one vertex (graph sparsity). Then, shifted Laplacian $\mathbf{L}$ can be rewritten as the best r-rank approximation part and the redundancy part, *i.e.*,

$$\begin{aligned}\mathbf{L}^{(i)} =& \mathbf{U}^{(i)}\mathbf{\Lambda}^{(i)}(\mathbf{U}^{(i)})^\top \\ =& \begin{bmatrix} \mathbf{U}_r^{(i)} & \overline{\mathbf{U}_r^{(i)}} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_r^{(i)} & \mathbf{0} \\ \mathbf{0} & \overline{\mathbf{\Lambda}_r^{(i)}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_r^{(i)} & \overline{\mathbf{U}_r^{(i)}} \end{bmatrix}^\top \\ =& \mathbf{U}_r^{(i)}\mathbf{\Lambda}_r^{(i)}(\mathbf{U}_r^{(i)})^\top + \overline{\mathbf{U}_r^{(i)}}\,\overline{\mathbf{\Lambda}_r^{(i)}}\,(\overline{\mathbf{U}_r^{(i)}})^\top = \mathbf{L}_r^{(i)} + \overline{\mathbf{L}_r^{(i)}}\end{aligned} \tag{7}$$

where $\mathbf{0}$ stands for an all-zeros matrix with size $(n-r)\times(n-r)$, $\mathbf{\Lambda}_r^{(i)}$ is the r largest eigenvalues of $\mathbf{L}$ (*i.e.*, $\delta_1, \delta_2, \cdots, \delta_r$), and $\mathbf{U}_r^{(i)}$ is the corresponding r eigenvectors of $\mathbf{\Lambda}_r^{(i)}$. Similarly, $\overline{\mathbf{\Lambda}_r^{(i)}}$ and $\overline{\mathbf{U}_r^{(i)}}$ are the rest $(n-r)$ eigenvalues (*i.e.*, $\delta_{r+1}, \delta_{r+2}, \cdots, \delta_n$) and eigenvectors, respectively. Therefore, for each kernel induced affinity garph $\mathbf{K}^{(i)}$, the r-rank eigenvectors can be obtained, where $c \le r \ll n$, which encode the cluster information and main reconstruction energy of its corresponding shifted Laplacian $\mathbf{L}^{(i)}$. At the initial stage, we let r to be greater than c for extracting extra information from each Laplacian. In theory, we can reconstruct the optimal shifted Laplacian matrix $\mathbf{L}$ via

$$\min_{\mathbf{L}} \|\mathbf{L} - \sum_{i=1}^{m} \mathbf{g}_i \mathbf{L}_r^{(i)}\|_F^2 \quad \text{s.t. } \mathbf{g}^\top \mathbf{1} = 1, \mathbf{g} \ge 0, \mathbf{L} \ge 0 \tag{8}$$

where the SPSD constraint is imposed to guarantee that the learned matrix $\mathbf{L}$ to be an approximate Laplacian matrix.

Denote $\lambda_p(\mathbf{L}), p = 1, \cdots, n$ as the eigenvalues of shifted Laplacian matrix $\mathbf{L}$ in the decreasing order. According to Theorem 1, ideally, these eigenvalues can be segmented into

$$\lambda_p(\mathbf{L}) \begin{cases} = 2, & p = 1, \cdots, c \\ < 2, & p = c+1, \cdots, n. \end{cases} \tag{9}$$

To encourage the c largest eigenvalues of $\mathbf{L}$ holding the main energy and clustering information, we consider the energy losing of Laplacian reconstruction and the clustering information preserving of Laplacian decompose simultaneously. That is, we should maximize the sum of the c largest eigenvalues, *i.e.*,

$$\max \sum_{p=1}^{c} \lambda_p(\mathbf{L}) \tag{10}$$

According to [2], it can be rewritten as

$$\sum_{p=1}^{c}\lambda_p(\mathbf{L}) \Rightarrow \max \mathtt{Tr}(\mathbf{LQ}) \\ \text{s.t. } \mathbf{Q}\in\mathbb{R}^{n\times n}, \mathtt{Tr}(\mathbf{Q})=c, \mathbf{Q}^{\mathsf{T}}=\mathbf{Q}, \mathbf{0}\preceq\mathbf{Q}\preceq\mathbf{1} \tag{11}$$

Such a problem is a Fantope projection problem [2], which projects the reconstructed shifted Laplacian $\mathbf{L}$ onto Fantope *i.e.*, $\mathtt{Tr}(\mathbf{Q}) = c, \mathbf{Q}^{\mathsf{T}} = \mathbf{Q}, \mathbf{0} \preceq \mathbf{Q} \preceq \mathbf{1}$. Note here that since $\max \mathrm{Tr}(\mathbf{LQ})$ can be approximate to $\min \|\mathbf{L}-\mathbf{Q}\|_F^2$. That is, ideally, $\mathbf{L}$ should be the same as $\mathbf{Q}$; therefore, we can transform the constraint of $\mathbf{Q}$ to $\mathbf{L}$, *i.e.*, $\mathrm{Tr}(\mathbf{L}) = c, \mathbf{L}^\top = \mathbf{L}, \mathbf{0} \preceq \mathbf{L} \preceq \mathbf{1}$.

Hereto, the final objective function can be integrated as

$$\min_{\mathbf{L},\mathbf{g}} \|\mathbf{L}-\sum_{i=1}^{m}\mathbf{g}_i\mathbf{L}_r^{(i)}\|_F^2 = \|\mathbf{L}-\sum_{i=1}^{m}\mathbf{g}_i\mathbf{U}_r^{(i)}\mathbf{\Lambda}_r^{(i)}(\mathbf{U}_r^{(i)})^\top\|_F^2 \\ \text{s.t. } \mathbf{g}^\top\mathbf{1}=1, 1\geq\mathbf{g}\geq 0, \mathbf{L}\geq 0, \\ \mathtt{Tr}(\mathbf{L})=c, \mathbf{L}^\top=\mathbf{L}, \mathbf{0}\preceq\mathbf{L}\preceq\mathbf{1} \tag{12}$$

In summary, the advantages of ASLR are three-fold:

- It is a simple and effective MKC method with lower memory and computational costs, compared to the existing MKC ones. And, it can be easily upgraded to a parameter free version.
- It considers the energy losing of Laplacian reconstruction and the clustering information preserving of Laplacian decompose simultaneously for the first time.
- It involves two parameters, *i.e.*, the rank of Laplacian r and the number of neighbors k, which are integer guided by the number of clusters (*i.e.*, c), while the parameters of the existing methods always involve some free decimal parameters. That is, our method is easy to tune. For simplicity, we can tune r's rate from 0.1 to 1 with step 0.1, multiplying the number of samples n; and tune k's scale from 1 to n/c, multiplying the number of clusters c. This property is important since there are few labeled data in clustering tasks.

4 OPTIMIZATION

4.1 Coordinate Descent Solver of Proposed ASLR

The final problem (12) is convex and can be effectively solved by a coordinate descent algorithm. Here, the proposed algorithm consists two steps: $\mathbf{g}$-step and $\mathbf{L}$-step.

► **$\mathbf{g}$-step:** Fixing $\mathbf{L}$, we update the weighting vector (*i.e.*, $\mathbf{g}$) of all candidate r-rank shifted Laplacians $\{\mathbf{L}_r^{(i)}\}_{i=1}^m$ via

$$\min_{\mathbf{g}} \frac{1}{2}\|\mathbf{L}-\sum_{i=1}^{m}\mathbf{g}_i\mathbf{L}_r^{(i)}\|_F^2 \quad \text{s.t. } \mathbf{g}^\top\mathbf{1}=1, 1\geq\mathbf{g}\geq 0 \tag{13}$$

Mathematically, this problem can be rewritten as

$$\min_{\mathbf{g}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{g}_i\mathbf{g}_j\mathtt{Tr}(\mathbf{L}_r^{(i)}\mathbf{L}_r^{(j)}) - \sum_{i=1}^{m}\mathbf{g}_i\mathtt{Tr}(\mathbf{L}\mathbf{L}_r^{(i)}) \\ \text{s.t. } \mathbf{g}^\top\mathbf{1}=1, 1\geq\mathbf{g}\geq 0 \tag{14}$$

which can be further reduced as following:

$$\min_{\mathbf{g}^\top\mathbf{1}=1,\mathbf{g}\geq 0} \frac{1}{2}\mathbf{g}^\top\mathbf{A}\mathbf{g} - \mathbf{g}^\top\mathbf{b} \tag{15}$$

where $\mathbf{a}_{ij} = \mathtt{Tr}(\mathbf{L}_r^{(i)}\mathbf{L}_r^{(j)})$ and $\mathbf{b}_i = \mathtt{Tr}(\mathbf{L}\mathbf{L}_r^{(i)})$ $(i = 1, 2, \cdots, m, j = 1, 2, \cdots, m)$. Such a problem can be effectively solved relying on a standard quadratic programming (QP) problem [14].

► **$\mathbf{L}$-step:** Fixing $\mathbf{g}$, we update the reconstructed shifted Laplacian $\mathbf{L}$ via

$$\min_{\mathbf{L}} \|\mathbf{L}-\mathbf{H}\|_F^2 \text{ s.t. } \mathtt{Tr}(\mathbf{L})=c, \mathbf{L}^\top=\mathbf{L}, \mathbf{0}\preceq\mathbf{L}\preceq\mathbf{1} \tag{16}$$

where $\mathbf{H} = \sum_{i=1}^m \mathbf{g}_i\mathbf{U}_r^{(i)}\mathbf{\Lambda}_r^{(i)}(\mathbf{U}_r^{(i)})^\top$, and $\mathbf{L}$ is the product by projecting $\mathbf{H}$ onto Fantope [2]. Then, such a problem can be effectively solved by Theorem 2.

Theorem 2 *For a symmetric affinity matrix $\mathbf{Q}\in\mathbb{R}^{n\times n}$, the spectral decomposition of $\mathbf{Q}$ is denoted as $\mathbf{M} = \mathbf{U}\mathtt{Diag}(\boldsymbol{\delta})\mathbf{U}^\top$. The following Fantope projection problem*

$$\min_{\mathbf{Q}} \frac{1}{2}\|\mathbf{Q}-\mathbf{M}\|_F^2 \text{ s.t. } \mathtt{Tr}(\mathbf{Q})=c, \mathbf{Q}^\top=\mathbf{Q}, \mathbf{0}\preceq\mathbf{Q}\preceq\mathbf{1} \tag{17}$$

has optimal solution given by $\mathbf{Q}^* = \mathbf{U}\mathtt{Diag}(\boldsymbol{\rho}^*)\mathbf{U}^\top$, where $\boldsymbol{\rho}^*$ is the solution to

$$\min_{\boldsymbol{\rho}} \frac{1}{2}\|\boldsymbol{\rho}-\boldsymbol{\delta}\|_2^2, \text{ s.t. } 0\leq\boldsymbol{\rho}\leq 1, \boldsymbol{\rho}^\top 1 = c \tag{18}$$

Proof. For two symmetric matrices $\mathbf{Q}\in\mathbb{R}^{n\times n}$ and $\mathbf{M}\in\mathbb{R}^{n\times n}$, and let $\rho_1\geq\rho_2\geq\cdots\geq\rho_n$ and $\delta_1\geq\delta_2\geq\cdots\geq\delta_n$ be the ordered eigenvalues of $\mathbf{Q}$ and $\mathbf{M}$, respectively. Due to the fact that $\mathtt{Tr}(\mathbf{Q}^\top\mathbf{M}) \leq \sum_{i=1}^n \rho_i\delta_i$ shown in [13], we can obtain

$$\begin{aligned}\|\mathbf{Q}-\mathbf{M}\|_F^2 &= \mathtt{Tr}(\mathbf{Q}^\top\mathbf{Q}) + \mathtt{Tr}(\mathbf{M}^\top\mathbf{M}) - 2\mathtt{Tr}(\mathbf{Q}^\top\mathbf{M}) \\ &= \sum_{i=1}^{n}\rho_i^2 + \sum_{i=1}^{n}\delta_i^2 - 2\mathtt{Tr}(\mathbf{Q}^\top\mathbf{M}) \\ &\geq \sum_{i=1}^{n}\left(\rho_i^2+\delta_i^2-2\rho_i\delta_i\right) = \|\boldsymbol{\rho}-\boldsymbol{\delta}\|_2^2\end{aligned}$$

Note here that the above equality holds when $\mathbf{Q}$ admits the spectral decomposition $\mathbf{M} = \mathbf{U}\mathtt{Diag}(\boldsymbol{\delta})\mathbf{U}^\top$. Additionally, the constraints $\mathbf{0}\preceq\mathbf{Q}\preceq\mathbf{1}, \mathtt{Tr}(\mathbf{Q}) = c$ are equivalent to $0\leq\boldsymbol{\rho}\leq 1, \boldsymbol{\rho}^\top 1 = c$, respectively. Thus $\mathbf{Q}^* = \mathbf{U}\mathtt{Diag}(\boldsymbol{\delta}^*)\mathbf{U}^\top$ is optimal to problem (17) with $\boldsymbol{\delta}^*$ being optimal to problem (18). After that, we can obtain the final solution, $\mathbf{Q}^* = (\mathbf{Q}^* + (\mathbf{Q}^*)^\top)/2$, to satisfy the balance constraint $\mathbf{Q}^\top = \mathbf{Q}$. ■

Finally, problem (18) can be efficiently solved via the capped simplex projection algorithm [22]. Note here that a by-product $\mathbf{U}\in\mathbb{R}^{n\times c}$ (*i.e.*, $\mathbf{H} = \mathbf{U}\mathtt{Diag}(\boldsymbol{\delta})\mathbf{U}^\top$) can be obtained when solving problem (16), which is fed into k-means algorithm for obtaining the resulting cluster labels. The pseudo-code of ASLR is depicted as in Algorithm 1, and the source code has uploaded as supplementary files for the purpose of reproducibility.

4.2 Computational Complexity and Convergence Study

The proposed Algorithm 1 consists of two simple steps, *i.e.*, updating the reconstructed shifted Laplacian $\mathbf{L}$ and updating the weighting vector $\mathbf{g}$. Their computational complexities are $O(rn^2)$ and $O(m^2)$, respectively. In addition, this algorithm only needs to storage $\mathbf{U}_r^{(i)}$ for the i-th kernel matrix, rather than the whole kernel matrix, such that the memory complexity is $O(mrn)$. Thus, since c, r and m are small numbers, the overall memory and computational complexity can be approximately $O(n)$ and $O(n^2)$, respectively.

Algorithm 1 Algorithm of the proposed ASLR method.

Input: Kernel pool $\{\mathbf{K}^{(i)}\}_{i=1}^{m}$, rank r and neighbors k.
Initialize: $\mathbf{g} = \text{ones}(m, 1)/m$.

```
for i = 1 : m do
    Construct the k-nearest neighbor graph of K^(i).
    Calculate the degree matrix D^(i) and shifted Laplacian matrix
    L^(i) of the base kernel K^(i) using Eq. (6).
    Calculate the approximated r-rank Laplacian matrix L_r^(i) by
    using Eq. (7).
end for
while not converge do
    Update the reconstructed Laplacian L using Eq. (16) and
    obtain the largest k eigenvectors (i.e., V) of L.
    Update the weighting value vector g using Eq. (15).
end while
Perform k-means algorithm on U.
```

Output: The cluster assignments of multiple kernel data.

Table 1: Summaries of the used MKC benchmark datasets.

Dataset	Category	Samples	Clusters	Kernels	Size
		n	c	m	
BBCSport	Sport news	554	5	2	4.40M
ProteinFold	Protein sequence	694	27	12	44.5M
Flower17	Flower	1360	17	3	94.3M
Caltech101	Object	1530	102	25	0.32G
UCI-Digit	Handwritten digit	2000	10	3	83.9M
Mfeat	Handwritten digit	2000	10	12	0.37G
CCV	Video event	6773	20	6	1.04G
Flower102	Flower	8189	102	4	1.98G

Theoretically, the two sub-problems are convex, since one is a Fantope projection problem [2], another is a quadratic programming problem [12]; meanwhile, the whole problem is lower bounded. Therefore, Algorithm 1 will reduce the objective function monotonically until convergence [4].

5 EXPERIMENT

5.1 Datasets and Experimental Setting

To report the performance fairly and roundly, eight widely used MKC benchmark datasets are employed, including BBCSport, Flower17, ProteinFold, Caltech101, UCI-Digit, Mfeat, CCV and Flower102 [16, 25]. These used datasets are widely used to evaluate MKC performance in existing references, and collected from various categories, such as news article, protein sequence, image and video, and the number of kernels, clusters, samples and categories show considerable variations. Their attributes are summarized in Table 1.

For kernel method, adopting different kernel functions of constructing kernels may result in various experimental performances [17]. However, MKL can perform automatic kernel selection and integrate complementary information of multiple base kernels. Usually, we can construct multiple base kernels according to the data attribute in advance (please refer to the second paragraph INTRODUCTION section 1). For example, if we chose a Gaussian kernel, it is a good choice to set its band parameter as the mean of Euclidean distance between all sample pairs. In this work, the used kernel datasets are public benchmark ones, it shall be fair to compare various methods with these datasets [11, 21].

Figure 2: Some images of the used image datasets.

As mentioned in INTRODUCTION section, no matter KKM-based methods or SC-based methods, k-means is an essential step, and its performance is sensitive to the initial cluster centers. Therefore, for each k-means step of all comparison methods, we repeat 50 times (each with a new set of initial cluster) and report the result with the smallest k-means distortion. The clustering metric, accuracy (ACC), normalized mutual information (NMI) and purity (PUR), are adopted to measure the clustering performance.

5.2 Comparison Methods

We compare our ASLR with the following state-of-the-art methods:

- **Average multiple kernel k-means (AMKC)**: Kernel k-means is performed on a uniformly weighting kernel.
- **Best single kernel k-means (BSKC)**: Kernel k-means algorithm is performed on each base kernel separately, and then the best score is reported.
- **Robust multiple kernel k-means (RMKKM)** [4]: RMKKM introduces an ℓ_{21} induced norm to reduce the influencers of large noise or outlier.
- **Optimal neighborhood kernel clustering (ONKC)** [11]: ONKC enhances the representability of the learned neighborhood kernel by construing a guidance matrix in advance.
- **Multi-view clustering via late fusion alignment maximization (MVC-LFA, LFA for short)** [21]: LFA maximally aligns the consensus partition with each weighted base partitions to improve the consistency for clustering.
- **Multi-view learning with adaptive neighbors (MLAN)** [15]: MLAN performs clustering and local adaptive structure learning simultaneously.

Table 2: Clustering results of different methods. The best and second best results are highlighted in red and blue, respectively. The last line shows the average running time. Note that '-' indicates the results are unavailable due to the long execution time.

Dataset	Metrics	AMKC	BSKC	RMKKM	ONKC	MLAN	LFA	LMVSC	ONMSC	MCLES	CoALa	MKC-CSA	ASLR
BBCSport	ACC	66.18	76.65	63.79	68.20	70.58	77.45	66.84	**93.75**	88.24	90.98	88.42	**95.22**
	NMI	53.92	59.38	39.62	54.64	65.34	55.63	50.21	**81.85**	76.77	76.75	72.16	**85.95**
	PUR	77.20	79.59	67.83	77.76	74.44	76.27	85.47	**93.75**	88.24	89.79	88.42	**95.22**
ProteinFold	ACC	30.69	34.58	30.98	37.90	28.38	**40.49**	29.25	38.18	32.71	39.27	35.30	**41.41**
	NMI	40.95	42.33	38.78	46.93	27.86	**48.96**	37.09	47.87	41.96	47.92	43.90	**49.26**
	PUR	37.17	41.21	36.60	45.24	31.84	**46.85**	31.84	46.11	39.19	45.25	40.63	**47.25**
Flower17	ACC	51.02	42.05	48.38	60.88	53.38	61.16	62.28	65.81	62.28	62.43	**66.76**	**68.75**
	NMI	50.18	45.14	50.73	58.58	55.38	60.79	61.71	**64.56**	61.71	61.59	63.37	**65.45**
	PUR	51.98	44.63	51.54	61.69	55.07	62.32	62.72	67.35	62.72	61.11	**68.75**	**71.40**
Caltech101	ACC	35.55	33.13	29.67	37.32	26.33	38.39	24.18	**39.74**	31.50	36.24	35.29	**42.48**
	NMI	59.90	59.06	55.86	61.41	43.25	61.65	52.65	**63.55**	54.42	59.75	56.51	**64.26**
	PUR	37.12	35.09	31.70	39.08	28.56	40.28	28.31	**41.96**	33.27	39.16	37.65	**44.44**
Mfeat	ACC	95.20	86.00	65.30	97.05	96.55	95.15	96.70	96.50	96.70	95.05	**97.30**	**97.52**
	NMI	89.83	75.78	62.67	97.05	92.89	95.00	92.74	92.53	92.74	94.78	**93.63**	**93.71**
	PUR	95.20	86.00	66.25	72.05	96.55	95.05	96.70	96.50	96.70	94.12	**97.30**	**98.10**
UCI-Digit	ACC	88.75	75.40	40.45	91.05	95.45	88.60	75.45	**96.15**	79.15	94.68	81.15	**97.50**
	NMI	80.59	68.38	46.87	83.96	91.38	88.25	69.87	**91.48**	78.67	89.54	83.39	**94.14**
	PUR	88.75	76.10	44.20	91.05	95.45	88.90	78.25	**96.15**	88.30	93.72	84.80	**97.50**
CCV	ACC	19.74	20.08	17.88	22.70	20.09	27.56	26.46	20.68	-	26.16	**32.59**	**33.44**
	NMI	17.16	17.73	15.44	18.70	15.90	20.59	21.25	17.10	-	20.06	**27.66**	**31.43**
	PUR	23.98	23.48	21.57	24.90	21.90	30.71	29.59	24.05	-	29.64	**35.39**	**35.55**
Flower102	ACC	27.29	33.12	28.17	41.56	24.19	42.16	37.06	**43.73**	-	40.06	42.18	**43.00**
	NMI	46.32	48.99	48.17	59.13	34.94	60.48	52.48	**60.52**	-	58.65	60.29	**60.38**
	PUR	32.27	38.78	27.61	47.64	31.15	50.44	42.31	**51.29**	-	44.14	45.88	**49.87**
Average running time		3.87	5.43	62.38	44.62	10.22	11.76	27.47	13.16	-	32.06	15.48	7.15

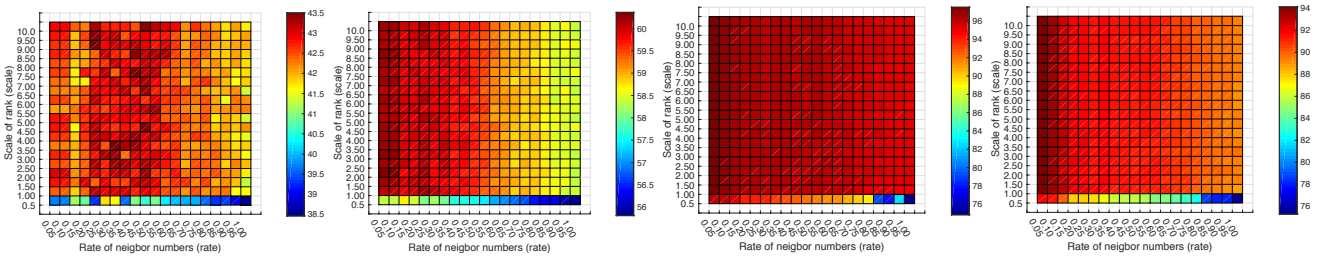
- **Large-scale multi-view subspace clustering (LMVSC)** [6]: LMVSC takes a sampling strategy to select some anchors for handing scalable multi-view subspace clustering.
- **Optimal neighborhood multi-view spectral clustering (ONMSC)** [24]: ONMSC learns a Laplacian matrix by searching a neighborhood of multiple candidate Laplacians.
- **Multi-view clustering in latent embedding space (MCLES)** [1]: MCLES partitions the multi-view data in a latent embedding space by simultaneously learning the cluster indicator matrix and the global structure.
- **Convex-combination of approximate Laplacians (CoALa) [7]**: CoALa integrates noise-free approximations of multiple similarity graphs to construct a low-rank subspace for multimodal data clustering.
- **Multiple kernel clustering with compressed subspace alignment (MKC-CSA)** [25]: MKC-CSA reduces the sample dimensionality by dynamic sampling the kernel matrix.

In summary, all these comparison methods consist of: (1) SKC method, including AMKC; (2) MKC methods, including BSKC, RMKKM, ONKC, LFA, ONMSC and MKC-CSA; and (3) multi-view clustering (MVC) methods, including MLAN, MCLES and LMVSC. For these MVC methods, we treat the kernel matrix as plain data, and fed it into their corresponding algorithms. For fair comparison, the parameters of these comparison methods are carefully tuned by following the suggested experimental settings provided by their original papers. Note here that the extra diverse information term of ONMSC is omitted (*i.e.*, $\alpha = 0$) for fair comparison. The source code of ASLR will be public available upon acceptance.

5.3 Experimental Clustering Results

The experimental results are reported in Table 2, and the following observations can be obtained:

- Our ASLR achieves the best clustering results on most of the used benchmark datasets. Remarkably, it performs better than the recently proposed MKC-CSA, showing its effectiveness and superiority. Therefore, the results clearly show that ASLR is a promising MKC method, which can be used to cluster nonlinear data.
- In general, SKC methods are worse than MKC methods. However, in some cases, MKC method, *e.g.*, BSKC and RMKKM, are even slightly worse than the SKC method *e.g.*, AMKC. This indicates that adequately exploiting multiple kernels still needs good techniques.
- Compared to MLAN, MCLES and LMVSC, they treat the kernel as plain data rather than graph, such that the latent graph information cannot be utilized fully. Furthermore, due to the abuse of linear assumption, they have limitations for clustering nonlinear data. Whereas, ASLR treats kernel as affinity graph.



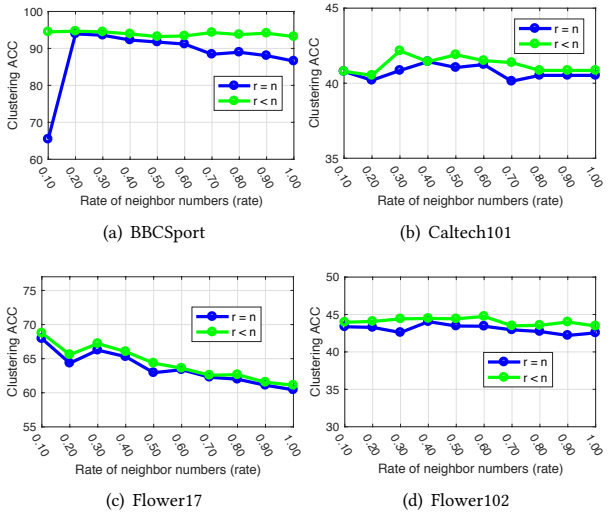
(a) ACC of Flower102 (b) NMI of Flower102 (c) ACC of UCI-Digit (d) NMI of UCI-Digit

Figure 3: The sensitivity study of ASLR with the variations of r and k on Flower102 and UCI-Digit datasets. Note here that `scale` and `rate` stand for the rate of neighbors and the scale of rank, respectively, *i.e.*, $k = \texttt{scale} * c$ and $r = \texttt{rate} * n$.

- Compared to CoALa, it employs shifted Laplacians to learn a low-rank subspace that best preserves the overall cluster information of multiple graphs. However, it does not consider the property of the Laplacian eigenvalues, resulting in suboptimal results. This demonstrates the potency of the proposed Fantope projection (*i.e.*, $\mathrm{Tr}(\mathbf{L}) = c, \mathbf{L}^\top = \mathbf{L}, \mathbf{0} \preceq \mathbf{L} \preceq \mathbf{1}$).
- The performance of our ASLR is superior to that of ONMSC. Although ONMSC also focus on reconstructing a Laplacian matrix to improve clustering performance, it does not consider the energy losing of Laplacian reconstruction and the clustering information preserving of Laplacian decompose simultaneously.
- In term of the average running times, our ASLR has a lower time cost compared to the MKC competitors. Note here that, our ASLR and MKC-CSA have similar computational complexity, but ASLR has a faster convergence speed. The main reasons are that (1) ASLR is a lightweight method, which only involves two optimization sub-problems, while other methods involves four, five or more sub-problems; (2) ASLR reconstructs Laplacian matrix via a r-rank approximate Laplacian rather than a full-rank approximation, such that the memory and computational complexity can be reduced to $O(n)$ and $O(n^2)$, respectively; and (3) ASLR has satisfactory convergence theoretically and experimentally, which usually converges in less than 5 iterations.

5.4 Effectiveness Study of Approximate r-rank Shifted Laplacian Reconstruction

In order to demonstrate the effectiveness of approximate r-rank shifted Laplacian reconstruction, we evaluate the clustering performance when $r = n$ and $r < n$, where $r = n$ indicates the approximate reconstruction is disabled. As shown in Fig. 4, r is fixed to $r = n$ and $r < n$ successively, and the rate of neighbors (*i.e.*, `scale`) is tuned from the ranges $[0.1, 0.2, \cdots, 1]$ with step 0.1, *i.e.*, $k = \texttt{scale} * c$. It can be observed that the clustering ACC when $r < n$ significantly and consistently outperforms that when $r = n$. The reason is that the r top large eigenvalues indicate the most informative parts of the learned shifted Laplacian, and can hold more prominent cluster characteristics. Whereas, the small eigenvalues may correspond to noise hidden in kernel data. Moreover, recall



(a) BBCSport (b) Caltech101 (c) Flower17 (d) Flower102

Figure 4: The clustering ACC in terms of $r = n$ and $r < n$ *w.r.t.* different neighbor rate (*i.e.*, `rate`) on four datasets.

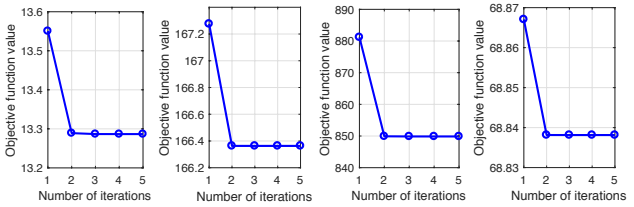
that the experimental results in Table 2, ASLR has a lower running time and higher clustering score, compared to ONMSC, this is mainly due to the proposed r-rank shifted Laplacian reconstruction. Therefore, the approximate r-rank shifted Laplacian reconstruction is a powerful method to filter noise and reduce running cost.

5.5 Parameter Sensitivity and Convergence

In Algorithm 1, two parameters are required to be set properly, *i.e.*, the numbers of neighbors k and the rank of shift Laplacian r. For simplicity tuning k and r, we introduce two auxiliary parameters, `scale` and `rate`, to stand for the rate of neighbors and the scale of rank, respectively. That is, $k = \texttt{scale} * c$ and $r = \texttt{rate} * n$. Subsequently, by using a grid search scheme, we tune `scale` and `rate` from the ranges $[0.5, 1, \cdots, 10]$ and $[0.05, 0.1, \cdots, 1]$ with step size 0.5 and 0.05, respectively. Take the Flower102 and UCI-Digit datasets for example, the clustering performance variations against k and r are illustrated in Fig. 3. We observe that: (1) the proposed method works well within a wide range of k and r values, and can

be easily tuned; (2) a small r can gain better or similar performance relative to a big r, thus the effectiveness of approximate Laplacian is demonstrated; (3) a relatively small k can lead to better performance than a big k, that is, the sparse 'prepruning' of kernel matrix can remove the redundancy edges and improve the representational ability of graph; and (4) the scores are not very smooth, the main result is that k-means is sensitive to the initialized cluster centers, it results in an undulatory standard deviation.

To demonstrate the convergence of algorithm experimentally, we record the objective function value at each iteration on the BBCSport, Caltech101, Flower17 and UCI-Digit datasets in Fig. 5. From the experimental study, the objective function value decreases monotonically as the number of iterations increases, usually converging to less than 5 epochs. Thereby demonstrating that fewer iteration steps can force the algorithm converges faster and stably, which promotes the proposed ASLR more efficient than other competitors.



(a) BBCSport (b) Flower17 (c) Caltech101 (d) UCI-Digit

Figure 5: Convergence curves of ASLR on four datasets.

6 CONCLUSION

While the recently proposed MKC methods are able to handle nonlinear data clustering, we have to face the brutal problem of high memory and computational complexity. In addition, to mitigate the problem, the existing MKC methods usually treat kernel matrix as plain data, and then reduce its dimensionality simply, leading to unsatisfying performance. This paper proposes a novel and simple MKC method *i.e.*, ASLR. It treats each kernel matrix as a affinity graph, and proposes an approximate r-rank shifted Laplacian reconstruction scheme to learn a consensus Laplacian matrix. Meanwhile, it projects the consensus Laplacian onto a Fantope to obtain the optimal eigenvectors for k-means purpose. Experimentally, its effectiveness and efficiency are well demonstrated by conducting convincing experiments on benchmark datasets, in comparison with state-of-the-art methods.

ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China (Grant nos. 62106209), the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) (Grant no. GML-KF-22-04), the Open Project Program of the State Key Lab of CAD & CG (Grant no. A2217), and the Natural Science Foundation of Southwest University of Science and Technology (Grant no. 22zx7101).

REFERENCES

[1] Man-Sheng Chen, Ling Huang, Chang-Dong Wang, and Dong Huang. 2020. Multi-view clustering in latent embedding space. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3513–3520.
[2] Jon Dattorro. 2010. *Convex optimization & Euclidean distance geometry.* Lulu. com.
[3] Charanpal Dhanjal, Romaric Gaudel, and Stéphan Clémençon. 2014. Efficient eigen-updating for spectral graph clustering. *Neurocomputing* 131 (2014), 440–452.
[4] Liang Du, Peng Zhou, Lei Shi, Hanmo Wang, Mingyu Fan, Wenjian Wang, and Yi-Dong Shen. 2015. Robust multiple kernel K-means using L2; 1-norm. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 3476–3482.
[5] Yu Geng, Zongbo Han, Changqing Zhang, and Qinghua Hu. 2021. Uncertainty-Aware Multi-View Representation Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7545–7553.
[6] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. 2020. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4412–4419.
[7] Aparajita Khan and Pradipta Maji. 2021. Approximate Graph Laplacians for Multimodal Data Clustering. *IEEE transactions on pattern analysis and machine intelligence* 43, 3 (2021), 798–813.
[8] Jiyuan Liu, Xinwang Liu, Siwei Wang, Sihang Zhou, and Yuexiang Yang. 2021. Hierarchical multiple kernel clustering. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 35. 8671–8679.
[9] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Xifeng Guo, Marius Kloft, and Liangzhong He. 2021. Multiview Subspace Clustering via Co-Training Robust Data Representation. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–13.
[10] Xinwang Liu, Li Liu, Qing Liao, Siwei Wang, Yi Zhang, Wenxuan Tu, Chang Tang, Jiyuan Liu, and En Zhu. 2021. One Pass Late Fusion Multi-view Clustering. In *International Conference on Machine Learning*, Vol. 139. 6850–6859.
[11] Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, and Jianping Yin. 2017. Optimal neighborhood kernel clustering with multiple kernels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31. 2266–2272.
[12] Istvan Maros and Csaba Mészáros. 1999. A repository of convex quadratic programming problems. *Optimization Methods and Software* 11, 1-4 (1999), 671–681.
[13] Albert W Marshall, Ingram Olkin, and James v Bondar. 1994. Inequalities: Theory of Majorization and Its Applications. *LINEAR ALGEBRA AND ITS APPLICATIONS* 199 (1994), 115–129.
[14] Jorge J Moré and Gerardo Toraldo. 1991. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization* 1, 1 (1991), 93–113.
[15] Feiping Nie, Guohao Cai, and Xuelong Li. 2017. Multi-View Clustering and Semi-Supervised Classification with Adaptive Neighbours. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31. 2408–2414.
[16] Zhenwen Ren, Quansen Sun, and Dong Wei. 2021. Multiple Kernel Clustering with Kernel k-Means Coupled Graph Tensor Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9411–9418.
[17] Zhenwen Ren, Simon X Yang, Quansen Sun, and Tao Wang. 2020. Consensus affinity graph learning for multiple kernel clustering. *IEEE Transactions on Cybernetics* 51, 6 (2020), 3273–3284.
[18] Mengjing Sun, Pei Zhang, Siwei Wang, Sihang Zhou, Wenxuan Tu, Xinwang Liu, En Zhu, and Changjian Wang. 2021. Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3528–3536.
[19] Yongqiang Tang, Yuan Xie, Xuebing Yang, Jinghao Niu, and Wensheng Zhang. 2021. Tensor Multi-Elastic Kernel Self-Paced Learning for Time Series Clustering. *IEEE Transactions on Knowledge and Data Engineering* 33, 03 (2021), 1223–1237.
[20] Rong Wang, Jitao Lu, Yihang Lu, Feiping Nie, and Xuelong Li. 2021. Discrete Multiple Kernel k-means.. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 3111–3117.
[21] Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. 2019. Multi-view clustering via late fusion alignment maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. 3778–3784.
[22] Weiran Wang and Canyi Lu. 2015. Projection onto the capped simplex. *arXiv preprint arXiv:1503.01002* (2015).
[23] Tian Xia, Dacheng Tao, Tao Mei, and Yongdong Zhang. 2010. Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 40, 6 (2010), 1438–1446.
[24] Sihang Zhou, Xinwang Liu, Jiyuan Liu, Xifeng Guo, Yawei Zhao, En Zhu, Yongping Zhai, Jianping Yin, and Wen Gao. 2020. Multi-view spectral clustering with optimal neighborhood Laplacian matrix. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6965–6972.

[25] Sihang Zhou, Qiyuan Ou, Xinwang Liu, Siqi Wang, Luyan Liu, Siwei Wang, En Zhu, Jianping Yin, and Xin Xu. 2021. Multiple Kernel Clustering With Compressed Subspace Alignment. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–12.