Multiple Kernel $k$-Means With Low-Rank Neighborhood Kernel

QIYUAN OU, LONG GAO, AND EN ZHU, (Member, IEEE)
School of Computer, National University of Defense Technology, Changsha 410073, China
Corresponding author: En Zhu (enzhu@nudt.edu.cn)
This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1003203; and in part by the National Natural Science Foundation of China under Grant 61672528, Grant 61773392, Grant 61872377, and Grant 62006237.

ABSTRACT Multiple kernel clustering algorithms achieve promising performances by exploring the complementary information from kernel matrices corresponding to each data view. Most of the existing methods aim to construct a consensus kernel for the afterward clustering. However, they ignore that the desired kernel is supposed to reveal the cluster structure among samples and thus to be low rank. As a consequence, the corresponding clustering performance could decrease. To address this issue, we propose a low-rank kernel learning approach for multiple kernel clustering. Specifically, instead of regularizing the consensus kernel with low-rank constraints, we use a re-parameterize scheme for the kernel matrix. Meanwhile, the consensus kernel is located in the neighborhood area of the linear combination of base kernels. An alternate optimization strategy is designed to solve the resulting optimization problem. We evaluate the proposed method on 13 benchmark datasets with 9 state-of-the-art algorithms. As is demonstrated by experimental results, our proposed algorithm achieves superior clustering scores against the compared algorithms on the reported popular multiple kernel datasets.

INDEX TERMS Clustering algorithms, multiple kernel clustering, multiple view clustering.

I. INTRODUCTION

Clustering is one of the major research topics regarding semi-supervised and unsupervised learning tasks, which allows the learning models to automatically uncover the patterns and structures of data and categorize these items into different classes accordingly. Owing to the variety of multimedia data sources, multi-view clustering has attracted increasing attention during the past decade. Although single-view algorithms such as [1], [2] can independently obtain several clustering results using distinct features provided by each view, the clustering accuracy is far from satisfactory. Therefore, researchers have made great efforts to design multi-view clustering algorithms to explore feature patterns provided by different views and fuse them to obtain optimal clustering results.

Researches on multi-view clustering methods can be classified into four categories [3]. The first category is co-training approach. It is based on the assumptions that 1) the features provided by each view are self-sufficient for the clustering task and 2) similar feature patterns lead to the same cluster prediction probabilities. Co-training method is a collaborative learning approach which usually involves two successive stages by firstly applying different algorithms to each view and secondly joining the separate results together [4]–[8]. The second category is multiple kernel clustering algorithm, which employs different kernels to introduce non-linearity to the clustering process [9], [10]. The difficulty of this method is to combine different kernels, with which [11]–[13] and [14] deal by matrix-induced regularization and late-fusion alignment, respectively. These solutions fuse information either in different ways or in different stages of the training process.

The associate editor coordinating the review of this manuscript and approving it for publication was Tomasz Trzcinski.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
high-dimensional data lies aside low-dimensional subspaces and can be represented linearly from the subspaces. Representative researches fall into two general types, namely, low-rank representation (LRR) [20] and sparse subspace clustering (SSC) [21]. Both of these approaches follow a similar algorithmic routine: first by aligning the representation with the original data, then by applying the reconstruction matrix to spectral clustering to acquire the final clustering result. The conformity of the alignment term makes it straightforward to extend to multi-view learning tasks, where alignment operations are optimized synchronically among multiple views [22]–[24].

Multiple kernel algorithms are developed to enlarge the searching space of feasible kernel mappings. One representative work in multiple kernel clustering is demonstrated by optimal neighborhood kernel clustering (ONKC) [25], [26], which extends the multiple kernel $k$-means clustering (MKKM) method [27] to a general formulation by mediating between the optimization procedure and the clustering process. Specifically, ONKC has delicately designed an approach to go beyond MKKM which merely allocates the optimal kernel using the linear combination of base kernels from different views. In additional to the simple assumption that the ideal kernel can be learned with linearity, ONKC imposes an alternative kernel which lies in the neighborhood of the linearly combined kernel, thus acquiring the property of nonlinearity. This method expands the searching scope of the possible optimal kernel for clustering. In addition, the injection of the alternative kernel builds a bridge between cluster indicating matrix and the coefficient vector of base kernels, because the optimization of the alternative kernel resorts to both the combined kernel and the indicating matrix. Accordingly, the clustering results using the neighborhood kernel as the input representation matrix outperforms the traditional MKKM methods.

The clustering results with neighborhood kernel outperform the traditional MKKM method. However, the only constraint of the neighborhood kernel matrix in ONKC is positive semi-definite (PSD) as a guarantee for its validity as a representative kernel matrix. In [25], The neighborhood kernel matrix combines the two processes (the kernel refinement process and the clustering process) into a unified framework. It can depict the non-linear structures and capture non-uniformly shaped clusters. Nevertheless, it takes no account of the rank constraint, making the clustering structure described by the kernel is indistinct and bear some noise which will inevitably bring distractions during the clustering process.

To address this issue, we present an algorithm named multiple kernel $k$-means with low-rank neighborhood kernel (MKKM-LR) in this paper. We propose an assumption that the kernel matrix with low-rank property and its rank in the vicinity of the number of clusters can depict the clustering structure, thus performing better clustering tasks. Specifically, we first acquire a linear combination of base kernels by equally assigning weights to each kernel. In the vicinity of this combined kernel, we further implement a neighborhood kernel to approximate the linear kernel, thus imposing property of non-linearity to the similarity matrix. We then re-parameterize the neighborhood matrix using the equivalent factorized formulation to refine the kernel matrix with an auxiliary low-rank constraint. Subsequently, the neighborhood kernel matrix is then applied to traditional MKKM to acquire the cluster representatives, through which we obtain the final clustering results by $k$-means clustering. We extensively evaluate our method on 13 multiple kernel learning (MKL) benchmark datasets and compare with nine state-of-the-art algorithms for demonstration of its feasibility.

Our contribution in this work can be summarized as follows:
• We propose a multiple kernel clustering algorithm by searching the neighborhood of the linear combination of base kernels and re-parameterizing the kernel matrix. Hence, the searching scope of the optimal kernel is enlarged and the kernel matrix naturally acquires the low-rank property.
• A formulation is proposed under the assumption that a kernel with low-rank property can represent the clustering structure thus better serve the clustering task. We design a four-step alternate optimization algorithm with proved convergence to learn the optimal kernel matrix. Theoretical analysis guarantees the low-rank property and reveals a relationship between the optimization process and the clustering process.
• We construct an experimental study on 13 benchmark datasets, covering a variety of applications and compare clustering performance with nine state-of-the-art algorithms. Our method demonstrates superior performance in most datasets with fast convergence.

This paper is organized as follows. Section II briefly outlines the kernel $k$-means task and the basis of our work. Section III formulates the optimization target and designs the four-step optimization algorithm. In Section IV, we analyze the clustering performance, convergence and parameter sensitivity of our method. Section V concludes our work.

II. RELATED WORK

A. KERNEL $k$-MEANS CLUSTERING (KKM)
$K$-means [28] is one of the most popular clustering methods which can cluster data into disjoint partitions through an iteratively convergent procedure. It has been widely applied in a variety of applications. However, this algorithm is restricted to linearly separable feature space because it is based on the Euclidean distance. To tackle this restriction, some methods are proposed to generalize $k$-means algorithm by integrating kernel functions [29], which can extend the original input space $X$ to a higher-dimensional linear-separable feature space $H$. The kernel mapping can not be directly calculated as it is implicit. Thus an indirect way is proposed by calculating the inner product $K(x, x')$, as explained below.

Let us denote the collection of $n$ samples as $\{x_i\}_{i=1}^{n}$ and denote the feature mapping as $\phi(x)$. The partitioning of data
points are denoted as \( \{C_i\}_{i=1}^k \), with each cluster signified as \( C_i \), and the centroids of clusters are denoted as \( \{c_i\}_{i=1}^k \). We can then formulate the kernel \( k \)-means objective as:

\[
F \left( \{C_i\}_{i=1}^k \right) = \sum_{i=1}^k \frac{1}{n_i} \| \phi(x) - c_i \|^2 ,
\]

where \( c_i = \frac{1}{n_i} \sum_{x \in C_i} \phi(a) \) is the \( i \)-th cluster centroid because it satisfies:

\[
c_i = \text{argmin}_y \sum_{x \in C_i} \| \phi(x) - y \|^2 ,
\]

and the distance from each data point to its clustering center is measured by the Euclidean distance:

\[
\left\| \phi(x) - \frac{1}{|C_i|} \sum_{a \in C_i} \phi(a) \right\|^2 = \phi(x)' \cdot \phi(x) - \sum_{a \in C_i} \frac{2}{|C_i|} \phi(a)' \cdot \phi(x) + \frac{1}{|C_i|^2} \sum_{a,b \in C_i} \phi(a)' \cdot \phi(b).
\]

Note that the distance measure reduces the prerequisite knowledge of feature mapping \( \phi(a) \) to the knowledge of dot product \( \phi(a)' \cdot \phi(b) \), thus the distance can be measured by the kernel function \( \mathcal{K}(\cdot, \cdot) \). Table 1 shows the popularly adopted kernel functions in practice.

The optimization problem in (1) can be reformulated using the matrix form:

\[
\min_S \text{Tr}(K) - \text{Tr} \left( L^{\frac{1}{2}} S^T K S L^{\frac{1}{2}} \right)
\]

subject to \( S I_k = I_n \),

where \( S \subseteq \{0, 1\}^{n \times k} \) is the cluster indicating matrix which satisfies \( \sum_{i=1}^n S_{ic} = |C_i| \) and \( \sum_{c=1}^k S_{ic} = 1 \). Here, \( L = \text{diag} \left[ \left| C_1 \right|^{-1} , \left| C_2 \right|^{-1} , \ldots , \left| C_k \right|^{-1} \right] \), \( I_k \) and \( I_n \) are \( k \times k \) and \( n \times n \) identity matrices respectively.

Owing to the fact that the discrete \( S \) is difficult to solve, a typical approach is to use a relaxed version of the discrete matrix and let it take real values. In specific, by defining \( U = SL^{\frac{1}{2}} \) and relaxing it to real values, the formula (4) can be expressed as:

\[
\min_U \text{Tr} \left( K(I_n - UU^T) \right)
\]

subject to \( U \in \mathbb{R}^{n \times k}, U^T U = I_k \).

The optimal \( U \) can be obtained by concatenating the \( k \) eigenvectors corresponding to the largest \( k \) eigenvalues of the kernel matrix \( K \).

### B. MULTIPLE KERNEL \( k \)-MEANS CLUSTERING (MKKM)

To benefit from different kernels that are suitable for different types of features, the multiple kernel \( k \)-means clustering (MKKM) task is proposed by combining different kernels with coefficients. The kernel combination coefficient can be constrained in linear [30], quadratic form [31] or \( L_2 \)-norm [32]. This paper is based on the linear form:

\[
\mathcal{K}_\lambda \left( x_i, x_j \right) = \sum_{p=1}^m \lambda_p K_p \left( x_i, x_j \right),
\]

where \( \mathcal{K}_p(\cdot, \cdot) \) denote kernel functions and \( \lambda = [\lambda_1, \ldots, \lambda_p, \ldots, \lambda_m]^T \) denotes the coefficient of each base kernel that forms the hyperplane of the combined kernel matrix, which can be optimized through the learning process [30].

Accordingly, by substituting the simple kernel matrix \( K \) in (5) with the combined matrix \( \mathcal{K}_\lambda \), the optimization target [30] for MKKM is:

\[
\min_{U \in \mathbb{R}^{n \times k}, \lambda \in \mathbb{R}^m} \text{Tr} \left( K \left( I_n - UU^T \right) \right)
\]

subject to \( U^T U = I_k \), \( \lambda^T I_m = 1 \).

The optimization problem of (7) can be divided into two steps: i) Optimizing \( U \) with \( \lambda \) fixed. With \( \lambda \) given, the combined kernel \( K \) can be regarded as a predefined kernel and the overall optimization problem is equivalent to:

\[
\min_{U \in \mathbb{R}^{n \times k}} \text{Tr} \left( K \left( I_n - UU^T \right) \right)
\]

subject to \( U^T U = I_k \),

which can be easily solved by the standard KKM optimization.

ii) Optimizing \( \lambda \) with \( U \) fixed. With \( U \) given, the coefficient vector \( \lambda \) can be optimized by solving:

\[
\min_{\lambda \in \mathbb{R}^m} \sum_{p=1}^m \lambda_p \text{Tr} \left( K_p \left( I_n - UU^T \right) \right)
\]

subject to \( \lambda^T I_m = 1 \),

this leads to a sparse solution of \( \lambda \).

### C. OPTIMAL NEIGHBOURHOOD MULTIPLE KERNEL CLUSTERING (ONKC)

By taking into account the nonlinearity of the optimal kernel matrix, [25] proposes the concept of the neighborhood kernel which resides alongside the linear combination of base kernels. The incorporation of the neighborhood kernel \( B \) can be fulfilled using the alignment with the linear kernel \( K \):

\[
\min_{U, \lambda, B} \text{Tr} \left( B \left( I_n - UU^T \right) \right) + \frac{\rho}{2} \| B - K \|_F^2 + \frac{\delta}{2} \lambda^T M \lambda
\]

subject to \( U, \lambda \in \mathbb{R}^{n \times k}, U^T U = I_k \),

\[
\lambda^T I_m = 1, \lambda_p \geq 0, \forall p,
\]

\[
B \succeq 0,
\]

where \( U \) is jointly optimized with the neighborhood kernel matrix \( B \) and the coefficient vector \( \lambda \).
It is worth mentioning that the neighborhood kernel \( \mathbf{B} \) is utilized to acquire the optimal kernel that better serves the clustering task. However, the above learning task in (10) neglects the constraints of the neighborhood kernel which could possibly refine the clustering structure. We observe that the only restriction of the optimal kernel \( \mathbf{B} \) is PSD, which is a very general requirement for the kernel matrix. This would result in abundant noise in the acquired kernel, leading to unsatisfactory clustering results. To address this issue, we propose a low-rank optimal neighborhood kernel clustering for multiple kernel learning to refine the optimal kernel matrix and improve the representative capability upon multiple base kernels.

### III. METHOD

In this section, we reformulate the clustering structure adopted in [25] and make some adjustments to the neighborhood kernel using the re-parameterization technique for further refinement III-A. Based on our formulation, a four-step alternate optimization algorithm in Section III-B is conducted efficiently with proved convergence.

#### A. THE PROPOSED FORMULATION

1) **RE-PARAMETERIZATION**

In order to tackle the problems mentioned in Section I, our motivation is to refine the optimal kernel matrix by re-parameterizing the optimal kernel while maintaining it in the neighborhood of the linearly combined kernel. Meanwhile, the linear kernel is one of the optimization variables, varying in coordination with the neighborhood kernel matrix. In addition to the PSD constraint of the optimal kernel, which is acquired in the neighborhood linear kernel, the re-parameterization strategy further imposes a low-rank characteristic on the kernel matrix, making the optimal kernel naturally possess a low-rank clustering structure.

Since the kernel matrix \( \mathbf{B} \) is rigorously restricted to be PSD, it can be further re-parameterized as \( \mathbf{B} = \mathbf{W} \Sigma \mathbf{W}^T \), where \( \mathbf{W} \) is an orthogonal matrix and \( \Sigma \) is a diagonal matrix with \( \text{diag}(\Sigma) = [\sigma_1, \sigma_2, \ldots, \sigma_c] \), \( \{\sigma_i\}_{i=1}^c \) are the top \( c \) eigenvalues of \( \mathbf{B} \). As \( \text{rank}(\mathbf{B}) = \text{rank}(\mathbf{W} \Sigma \mathbf{W}^T) \leq \text{rank}(\Sigma) = c \), the rank of kernel \( \mathbf{B} \) is constrained to be less than \( c \). By assigning a \( c < n \) to the re-parametric expression, we constrain the kernel matrix to be low-rank with the regularized re-parameterization matrix \( \mathbf{W} \) and the eigenvalue matrix \( \Sigma \). We further enforce our re-parameterization integrity to be in the locality of the linear kernel matrix \( \mathbf{K}_\lambda \), which lies in the hyperspace of the multiple base kernels. This can be formulated as:

\[
\min_{\Sigma, \lambda} \| \mathbf{W} \Sigma \mathbf{W}^T - \mathbf{K}_\lambda \|_F^2
\]

\[\text{s.t. } \lambda^T \mathbf{l}_m = 1, \lambda \in \mathbb{R}_+^m, \]

\[\mathbf{W} \in \mathbb{R}^{n \times c}, \mathbf{W}^T \mathbf{W} = \mathbf{I}_c, \]

\[\text{diag}(\Sigma) = [\sigma_1, \sigma_2, \ldots, \sigma_c]. \tag{11}\]

2) **MATRIX-INDUCED REGULARIZATION**

The minimization of matrix-induced regularization term increases the diversity and diminishes the redundancy of the multiple kernels [11]. Specifically, to utilize the information provided by multiple kernels, the similarity measure of different kernels is applied with a pre-defined criterion \( \mathcal{M}(\mathbf{K}_p, \mathbf{K}_q) \), and it is a consensus that higher value represents higher similarity. By minimizing the term \( \mu_p \mu_q \mathcal{M}(\mathbf{K}_p, \mathbf{K}_q) \), where \( \mu_p \) and \( \mu_q \) are factors of weights for \( \mathbf{K}_p \) and \( \mathbf{K}_q \), we can radically and effectively avoid duplicate large weights being assigned to homogeneous kernels. Based on this, the regularization term can be formulated as:

\[
\min_{\mu} \sum_{p,q=1}^m \mu_p \mu_q \mathcal{M}(\mathbf{K}_p, \mathbf{K}_q) = \mu^T \mathbf{M} \mu
\]

\[\text{s.t. } \mu^T \mathbf{l}_m = 1, \mu \in \mathbb{R}_+^m. \tag{12}\]

To merge the clustering process into the optimal kernel learning task, we consolidate the neighborhood alignment structure and the kernel regularization term with the MKKM framework. Thus we can obtain the following formulation:

\[
\min_{\mathbf{w}, \Sigma, \mathbf{U}, \mathbf{I}, \lambda} \text{Tr} \left[ \mathbf{W} \Sigma \mathbf{W}^T \left( \mathbf{I}_n - \mathbf{U} \mathbf{U}^T \right) \right]
\]

\[+ \frac{\rho}{2} \| \mathbf{W} \Sigma \mathbf{W}^T - \mathbf{K}_\lambda \|_2^2 + \frac{\delta}{2} \lambda^T \mathbf{M} \lambda
\]

\[\text{s.t. } \mathbf{U} \in \mathbb{R}^{n \times k}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_k,
\]

\[\lambda^T \mathbf{l}_m = 1, \lambda \in \mathbb{R}_+^m,
\]

\[\mathbf{W} \in \mathbb{R}^{n \times c}, \mathbf{W}^T \mathbf{W} = \mathbf{I}_c,
\]

\[\text{diag}(\Sigma) = [\sigma_1, \sigma_2, \ldots, \sigma_c]. \tag{13}\]

where \( \mathbf{M} \in \mathbb{R}^{m \times m} \) is the similarity criterion matrix with assignment function \( \mathcal{M}(\mathbf{K}_p, \mathbf{K}_q) = \text{Tr}(\mathbf{K}_p \mathbf{K}_q) \). \( \rho \) and \( \delta \) are trade-off factors for neighborhood kernel alignment and kernel information enhancement term respectively.

#### B. OPTIMIZATION ALGORITHM

We provide a four-step iterative optimization approach to approximate the optimal target in (13) with three variables fixed while the remaining one is optimized. The detailed procedure is derived below.

1) **OPTIMIZING \( \Sigma \) WITH \( \lambda \), \( \mathbf{U} \) AND \( \mathbf{W} \) FIXED**

Given \( \{\lambda_p\}_{p=1}^m \), \( \mathbf{U} \) and \( \mathbf{W} \), the linear kernel \( \mathbf{K}_\lambda \) in (13) can be calculated by \( \mathbf{K}_\lambda = \sum_{p=1}^m \lambda_p \mathbf{K}_p \), and the optimization target can be equivalently expressed by minimizing the following formula w.r.t. \( \Sigma \):

\[
\min_{\Sigma} \text{Tr} \left[ \Sigma^2 - 2 \Sigma \mathbf{W}^T \left[ \mathbf{K}_\lambda - \frac{1}{\rho} (\mathbf{I}_n - \mathbf{U} \mathbf{U}^T) \right] \right] \mathbf{W}
\]

\[\text{s.t. } \text{diag}(\Sigma) = [\sigma_1, \sigma_2, \ldots, \sigma_c]. \tag{14}\]

Denoting \( \mathbf{P} = \mathbf{W}^T \left[ \mathbf{K}_\lambda - \frac{1}{\rho} (\mathbf{I}_n - \mathbf{U} \mathbf{U}^T) \right] \), (14) can be further converted to:

\[
\min_{\Sigma} \text{Tr} \left[ \Sigma^2 - 2 \Sigma \mathbf{P} \right] \mathbf{W}
\]

\[\text{s.t. } \text{diag}(\Sigma) = [\sigma_1, \sigma_2, \ldots, \sigma_c]. \tag{15}\]

Note that for any quadratic optimization problem in the following form:

\[
\min_{\sigma} \sigma^T \sigma - 2 b^T \sigma,
\]

\[s.t. \quad \sigma^T \mathbf{I} \sigma = 1, \quad \mathbf{I} \in \mathbb{R}^{n \times n}, \quad b \in \mathbb{R}^n. \tag{16}\]
the optimal solution is:

$$\sigma_i = b_i,$$

for $i$ varying from 1 to $c$.

2) **OPTIMIZING $\mathbf{U}$ WITH $\Sigma$, $\lambda$ AND $\mathbf{W}$ FIXED**

With $\Sigma$, $(\lambda_p)_{p=1}^m$ and $\mathbf{W}$ given, the optimization problem of (13) w.r.t. $\mathbf{U}$ becomes:

$$\min_{\mathbf{U} \in \mathbb{R}^{n \times c}} \operatorname{Tr}(K_\lambda (I_n - \mathbf{UU}^\top))$$

subject to:

$$\mathbf{U}^\top \mathbf{U} = I_c,$$

(18)

which matches the form of the typical KKM problem (shown in (5)), and the optimal $\mathbf{U}$ can be comprised by taking the top $k$ eigenvectors.

3) **OPTIMIZING $\mathbf{W}$ WITH $\Sigma$, $\lambda$ AND $\mathbf{U}$ FIXED**

With $\Sigma$, $(\lambda_p)_{p=1}^m$ and $\mathbf{U}$ fixed, optimizing (13) w.r.t. $\mathbf{W}$ can be reformulated as follows:

$$\max_{\mathbf{W}} \operatorname{Tr}(W^\top (\Sigma W - \frac{1}{\rho} \lambda (I_n - \mathbf{UU}^\top)) \mathbf{W})$$

subject to:

$$\mathbf{W} \in \mathbb{R}^{n \times c}, \quad \mathbf{W}^\top \mathbf{W} = I_c.$$

Denoting $\mathbf{Q} = \mathbf{K}_\lambda - \frac{1}{\rho} \lambda (I_n - \mathbf{UU}^\top)$, the optimization problem in (19) can be simplified as:

$$\max_{\mathbf{W}} \operatorname{Tr}(\Sigma W^\top \mathbf{Q} W)$$

subject to:

$$\mathbf{W} \in \mathbb{R}^{n \times c}, \quad \mathbf{W}^\top \mathbf{W} = I_c.$$

Likewise, the maximization of $\operatorname{Tr}(\Sigma W^\top \mathbf{Q} W)$ w.r.t $\mathbf{W}$ is similar to the kernel $k$-means task and satisfies the following equation:

$$\max_{\mathbf{W}} \operatorname{Tr}(\Sigma W^\top \mathbf{Q} W) = \sum_{i=1}^c \sigma_i \beta_i$$

subject to $\text{diag}(\Sigma) = [\sigma_1, \sigma_2, \ldots, \sigma_c],$

(20)

where $\beta_i$ is the largest $c$ arbitrary eigenvalues of $\mathbf{Q}$.

4) **OPTIMIZING $\lambda$ WITH $\Sigma$, $\mathbf{W}$ AND $\mathbf{U}$ FIXED**

Given $\Sigma$, $\mathbf{W}$ and $\mathbf{U}$, the optimization problem in (13) w.r.t $(\lambda_p)_{p=1}^m$ can be reformulated as,

$$\min_{\lambda} \frac{\rho + \delta}{2} \lambda^\top \mathbf{M} \lambda - z^\top \lambda$$

subject to:

$$\lambda^\top \mathbf{1}_m = 1, \quad \lambda \in \mathbb{R}_+^m,$$

(22)

where $z = [z_1, \ldots, z_m]^\top$ with $z_p = \rho \operatorname{Tr}(W \Sigma W^\top K_p)$. This is a standard quadratic programming problem with equality and inequality linear constraints.

The overall optimization procedure is demonstrated in Algorithm 1. It is worth noticing that as the four iterative optimization processes perform with three other variables fixed, each subproblem is strictly convex w.r.t. the target variable. As a result, the optimization target will decrease monotonously for each iteration. Moreover, the whole optimization problem is lower-bounded. The convergence of the algorithm is accordingly verified.

### Algorithm 1 Multiple Kernel $k$-Means With Low-Rank Neighborhood Kernel (MKKM-LR)

**Input:**

- Base kernels $(\mathbf{K}_1, \ldots, \mathbf{K}_m)$, number of clusters $k$, parameters $\rho$ and $\delta$, rank $c$ of $\Sigma$ and convergence boundary $\epsilon = 10^{-3}$.

**Output:**

- The cluster representation matrix $\mathbf{U}$.

1. **Initialization** Initialize $\lambda^{(0)}$ by assigning $\lambda^{(0)} = \mathbf{1}_m/m$, and $\mathbf{K}^{(0)} = \sum_{p=1}^m \lambda^{(0)}_p \mathbf{K}_p$. Initialize $\Sigma^{(0)} = \mathbf{I}_k$. Initialize $\mathbf{U}$ by a $c \times c$ identity matrix. Initialize $\mathbf{U}$ by calculating (18). Assign $t = 1$.

2. **repeat**

3. Calculate $\Sigma^{(t)}$ by optimizing (14).

4. Update $\mathbf{U}^{(t)}$ by solving (18) with $\Sigma$, $\lambda$ and $\mathbf{W}$ given.

5. Update $\mathbf{W}^{(t)}$ with fixed $\Sigma$, $\lambda$ and $\mathbf{U}$ by optimizing (19).

6. Update $\lambda^{(t)}$ by optimizing (22) with $\mathbf{Sigma}$, $\mathbf{U}$ and $\mathbf{W}$ fixed.

7. $t = t + 1$.

8. until $\frac{\text{obj}^{(t)} - \text{obj}^{(t-1)}}{\text{obj}^{(t-1)}} < \epsilon$.

Perform $k$-means clustering on $\mathbf{U}$.

To summarize, as the optimization of the neighborhood kernel matrix depends on both the linear kernel and the cluster representation matrix, the clustering structure and the refinement of the kernel matrix are bonded together. With the help of the iterative learning process, this joint effectuation of mutual optimization spreads across all learnable variables, making it a unified framework for the multiple kernel clustering task.

### C. DISCUSSION AND EXTENSION

In this section, we mainly discuss the relationship and difference between our proposed method and other analogical algorithms. We then explore some potential extensions of the proposed method. In the end, the computational cost is analyzed.

Our algorithm adopts a re-parametric scheme by separately and iteratively optimizing two instead of one single variable to delicately learn the optimal kernel matrix. With the orthogonal constraint of one of the multiplier component $\mathbf{W}$, the kernel matrix is representative and distinctive. Additionally, re-parameterization naturally acquires the low-rank property, which will benefit the clustering task. In this case, the representation matrix is expected to be of rank $k$ (number of the cluster), and distinctive in each row vector. Similar to [11], [12], the matrix-induced term regulates the weights of kernels and utilizes mutual information of multiple base kernels, thus reducing redundancy while boosting the diversity of the representation matrix. One of the differences between our algorithm and the most existing MKKM methods is that we use a neighborhood matrix (as was adopted in [25]) to expand the searching space of the optimal kernel. The re-parametric strategy further provides guidance to the rank of the optimal
kernel, which in turn assists the optimization of the linear kernel.

In our algorithm, the rank $c$ of the optimal kernel is a hyper-parameter with $c \in \{ \frac{n}{10}, \frac{2n}{10}, \ldots, n \}$, where $n$ is the data count. The best choice of the rank of the optimal kernel is an open question, and it is readily extensible to an adaptive parameter version. We also observe that by setting the rank to multiples of the number of clusters, the clustering results reach a local maximum compared to the rank numbers nearby.

MKKM-LR’s effectiveness benefits from the low-rank regulation of the optimal kernel matrix, which is ensured by the re-parametric scheme. In addition to the computational cost of the standard MKKM algorithm, the optimization of our formulation further requires the computation of the orthogonal multiplier matrix $W$. While updating the diagonal matrix $\Sigma$, the computational cost is $O(n^3(k + c) + nc^2)$, where $k$ is the number of clusters and $c$ is the rank of $\Sigma$. Optimizing $U$ requires $O(n^3)$ to take eigenvalue decomposition. Optimizing $W$ requires $O(n^3 + n^2k)$. And the quadratic programming costs $O(2nc^2 + nc^2 + m^3)$ computational time while optimizing $\lambda$. Overall, the computational complexity is comparable with the existing MKKM, which is $O(n^3)$. In the next section, experimental results demonstrate superior performance in numerous benchmark datasets.

IV. EXPERIMENTS AND ANALYSIS

A. EXPERIMENTAL SETTINGS

1) DATASETS

The benchmark datasets used in our experiments are ProteinFold, Digital, Flower17, psortPos, Plant, warpiAR10P, Cornell, Heart, warpiPIE10P, Washington, YALE, Orl and Texas. These datasets are in different varieties and cover a wide range of real-world applications, such as images and protein sequences. The detailed information of these benchmark datasets is listed in Table 2. The number of classes in each dataset is given in advance and the base kernels are normalized beforehand.

2) COMPARED ALGORITHMS

In our experiment, MKKM-LR is compared with several state-of-the-art multi-view clustering algorithms listed below.

A) Average Multiple Kernel $k$-means (A-MKKM): By uniformly weighting all base kernels, a new kernel is generated and taken as the input of the kernel $k$-means.

B) Single Best Kernel $k$-means (SB-KKM): Each single kernel is separately tested on the kernel $k$-means and the best result is reported.

C) Multiple Kernel $k$-means (MKKM): As shown in (7), MKKM performs kernel $k$-means and updates kernel coefficients alternately to obtain the best combinations.

D) Robust multiple kernel $k$-means (RMKKM) [33]: RMKKM proposes to replace the sum-of-squared loss with an $\mathcal{L}_{2,1}$-norm to improve the robustness of MKKM.

E) Co-regularized Spectral Clustering (CRSC) [34]: CRSC provides a co-regularization way to perform spectral clustering.

F) Robust multiview spectral clustering (RMSC) [35]: RMSC constructs a transition probability matrix from each single view, and uses them to recover a shared low-rank transition matrix for clustering.

G) Multiple kernel $k$-means with matrix-induced regularization (MKKM-MR) [12]: MKKM-MR learns the optimal combination weights by introducing a matrix-induced regularization to reduce the redundancy and enhance the diversity among the selected kernels.

H) Optimal neighborhood kernel clustering with multiple kernels (ONKC) [25]: While sufficiently considering the correlation between base kernels, ONKC also allows the optimal kernel to reside in the neighborhood of linear combination of base kernels and effectively enlarges the region from which an optimal kernel can be chosen.

I) Robust Multiple Kernel Clustering (RMKC) [36]: RMKC learns a robust yet low-rank kernel for clustering by capturing the structure of noises in multiple kernels.

B. RESULT ANALYSIS

In our experiments, the clustering accuracy (ACC), normalized mutual information (NMI), and purity are used to evaluate the clustering performance of these ten algorithms. According to the clustering results and convergence analysis listed in Table 3 and Figure 1, we can draw the following conclusions:

- The proposed algorithm outperforms on 11 out of 13 benchmark datasets (as shown in red) with dominant superiority, substantially exceeding the second best methods. In particular, our algorithm’s accuracy (55.38%) exceeds the second-best algorithm ONKC (47.69%) on the dataset warpiAR10P by 16.1% in proportion. Besides, there are more than 5% increment w.r.t. the best performance achieved by previous methods on datasets such as ProteinFold, Orl and Cornell.

- In datasets Washington and Texas, where the proposed method does not fulfill the best performance in accuracy, those that achieve the best ACC score fail to sustain comparable performances in the other two criteria, namely, NMI and purity. However, our method demonstrates...
TABLE 3. Clustering metrics on benchmark datasets. The best results are highlighted in red and the second best results are highlighted in blue.

| Datasets     | MKKM | A-MKKM | SB-KKM | RMKKM | CRSSC | RMSSC | RMKC | MKKM-MR | ONKC | Proposed |
|--------------|------|--------|--------|-------|-------|-------|------|--------|------|----------|
| ProteinFold  | 37.16| 38.53  | 42.03  | 38.78 | 43.34 | 43.91 | 39.46| 43.52  | 46.93| 47.67    |
| Digital      | 48.16| 80.59  | 68.38  | 46.87 | 73.51 | 81.80 | 80.88| 83.22  | 83.96| 84.53    |
| Flower17     | 45.35| 50.19  | 45.14  | 50.73 | 52.13 | 53.89 | 50.42| 57.11  | 58.28| 60.01    |
| psoptPos     | 35.01| 28.86  | 42.54  | 36.54 | 39.98 | 30.69 | 28.86| 26.74  | 44.33| 44.76    |
| Plant        | 20.02| 26.57  | 17.40  | 19.39 | 22.71 | 23.18 | 26.57| 21.56  | 30.94| 31.81    |
| warpAR10P    | 40.07| 36.07  | 42.61  | 26.60 | 34.41 | 26.14 | 36.07| 41.81  | 50.35| 54.94    |
| Cornell      | 4.90 | 35.33  | 35.86  | 9.57  | 19.13 | 39.22 | 36.60| 43.11  | 38.70| 43.31    |
| Heart        | 0.02 | 32.40  | 20.49  | 20.49 | 28.53 | 34.51 | 32.40| 34.24  | 39.18| 39.18    |
| warpPIE10P   | 76.23| 49.20  | 78.67  | 39.95 | 60.64 | 33.28 | 49.20| 59.18  | 82.01| 82.01    |
| Washington   | 26.80| 32.57  | 32.68  | 33.06 | 21.14 | 33.62 | 33.43| 34.86  | 35.47| 35.29    |
| Yale         | 54.16| 57.72  | 58.42  | 59.70 | 56.89 | 59.11 | 57.69| 61.29  | 62.27| 63.13    |
| Orl          | 74.05| 60.51  | 79.97  | 35.68 | 54.06 | 25.50 | 77.51| 66.22  | 80.32| 85.61    |
| Texas        | 12.04| 29.60  | 30.52  | 33.26 | 20.40 | 25.40 | 30.19| 33.22  | 32.42| 34.90    |

a stable and well-balanced performance throughout all these three criteria, with NMI the best in Texas, the second in Washington and purity the best both in Texas and Washington.

- Without manifest deficit in any criterion, the proposed algorithm can improve ACC, NMI and purity simultaneously, validating the robustness of our method.
- The proposed algorithm has a fast convergence speed (see Figure 1). It generally can reach its predefined convergence condition within 3 iterations, which brings down the time cost by a large margin and improves the efficiency of the algorithm.

In summary, the magnificent improvement of the proposed method verifies the effectiveness of multiple kernel k-means clustering with low-rank neighborhood kernel, which employs the re-parametric scheme to subtly and skillfully optimize the optimal kernel matrix, naturally acquiring the low-rank property while maintaining its clustering structure.

C. PARAMETER SENSITIVITY

The proposed algorithm has three hyper-parameters $\rho$, $\delta$ and c, as discussed in Section III-C. $\rho$ and $\delta$ are trade-off parameters for neighborhood matrix alignment term and kernel information diversity enhancement term, c is set with different percentages of the sample number. In our experimental settings, we investigate the influence of two hyper-parameters upon the clustering result at each time, as demonstrated in Figures 2, 3, 4. The vertical axis shows the accuracy influenced by every two parameters pairwise. From those figures, we can conclude that our algorithm exhibits competitive while stable clustering performance across a wide range of hyper-parameters. In addition, we observe...
that for most benchmark datasets, our algorithm achieves the best clustering performance when the parameter \( c \) is small, which means the kernel matrix is low-rank. This proves the effectiveness of our proposed method.
In this paper, we propose a novel multiple kernel k-means clustering with low-rank neighborhood kernel. This algorithm learns a low-rank optimal kernel matrix with a distinct clustering structure while effectively improves the clustering presentation matrix with the learned neighborhood matrix. Extensive experiments verify the effectiveness of the proposed method with notably rapid convergence speed. In the future, we will explore a heuristic or adaptive method to automatically choose the best rank $c$ of the optimal kernel matrix, further releasing a hyper-parameter in the algorithm.

**REFERENCES**

[1] F. Pourkamali-Anaraki, J. Folberth, and S. Becker, “Efficient solvers for sparse subspace clustering,” *Signal Process.*, vol. 172, Jul. 2020, Art. no. 107548.

[2] C. You, D. P. Robinson, and R. Vidal, “Scalable sparse subspace clustering by orthogonal matching pursuit,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA: IEEE Computer Society, Jun. 2016, pp. 3918–3927.

[3] Y. Yang and H. Wang, “Multi-view clustering: A survey,” *BID Data Mining Anal.*, vol. 1, no. 2, pp. 83–107, Jun. 2016.

[4] A. Kumar and H. Daume, “A co-training approach for multi-view spectral clustering,” in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, L. Getoor and T. Scheffer, Eds. Bellevue, WA, USA: Omnipress, Jun. 2011, pp. 393–400.

[5] Y. Jiang, J. Liu, Z. Li, and H. Lu, “Collaborative PLSA for multi-view clustering,” in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Tsukuba, Japan: IEEE Computer Society, Nov. 2012, pp. 2997–3000.

[6] M. Ghassany, N. Grozavu, and Y. Bennani, “Collaborative multi-view clustering,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Dallas, TX, USA, Aug. 2013, pp. 1–8.

[7] Y. Jiang, F.-L. Chung, S. Wang, Z. Deng, J. Wang, and P. Qian, “Collaborative fuzzy clustering from multiple weighted views,” *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 688–701, Apr. 2015.

[8] F. Ma, D. Meng, X. Dong, and Y. Yang, “Self-paced multi-view co-training,” *J. Mach. Learn. Res.*, vol. 21, no. 57, pp. 57:1–57:38, 2020.

[9] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, and J. Yin, “Multiple kernel clustering with neighbor-kernel subspace segmentation,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1351–1362, Apr. 2020.

[10] S. Zhou, E. Zhu, X. Liu, T. Zheng, Q. Liu, J. Xia, and J. Yin, “Subspace segmentation based robust multiple kernel clustering,” *Inf. Fusion*, vol. 53, pp. 145–154, Jan. 2020.

[11] C. Hinrichs, V. Singh, J. Peng, and S. C. Johnson, “Q-MKL: Matrix-induced regularization in multi-kernel learning with applications to neuromaging,” in *Advances in Neural Information Processing Systems 25*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Lake Tahoe, NV, USA, 2012, pp. 1430–1438.

[12] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, “Multiple kernel K-means clustering with matrix-induced regularization,” in *Proc. 30th AAAI Conf. Artif. Intell.*, D. Schuurmans and M. P. Wellman, Eds. Phoenix, AZ, USA: AAAI Press, 2016, pp. 1888–1894.

[13] Y. Zhao, Y. Dou, X. Liu, and T. Li, “A novel multi-view clustering method via low-rank and matrix-induced regularization,” *Neurocomputing*, vol. 216, pp. 342–350, Dec. 2016.

[14] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, “Multi-view clustering via late fusion alignment maximization,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, J. Kraus, Ed. Macao, China, Aug. 2019, pp. 3778–3784.

[15] W. Tang, Z. Lu, and I. S. Dhillon, “Clustering with multiple graphs,” in *Proc. 9th IEEE Int. Conf. Data Mining*, W. Wang, H. Kargupta, S. Rana, P. S. Yu, and X. Wu, Eds. Miami, FL, USA: IEEE Computer Society, Dec. 2009, pp. 1016–1021.

[16] 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 *Proc. 34th AAAI Conf. Artif. Intell.*, 32nd Innov. Appl. Artif. Intell. Conf., AAAI, 10th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI), New York, NY, USA: AAAI Press, 2020, pp. 6965–6972.

[17] A. Serra, D. Greco, and R. Tagliaferri, “Impact of different metrics on multi-view clustering,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[18] Z. Xue, G. Li, S. Wang, C. Zhang, W. Zhang, and Q. Huang, “GOMES: A group-aware multi-view fusion approach towards real-world image clustering,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Turin, Italy: IEEE Computer Society, Jun. 2015, pp. 1–6.

[19] F. Nie, J. Li, and X. Li, “Parameter-free auto-weighted multiple graph learning: A framework for multi-view clustering and semi-supervised classification,” in *Proc. 35th Int. Joint Conf. Artif. Intell. (IJCAI)*, S. Kamhkampati, Ed. New York, NY, USA: IJCAI/AAAI Press, 2016, pp. 1881–1887.

[20] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[21] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[22] M. White, Y. Yu, X. Zhang, and D. Schuurmans, “Convex multi-view subspace learning,” in *Advances in Neural Information Processing Systems 25*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Lake Tahoe, NV, USA, 2012, pp. 1682–1690.

[23] Y. Guo, “Convex subspace representation learning from multi-view data,” in *Proc. 27th AAAI Conf. Artif. Intell.*, M. des Jardins and M. L. Littman, Eds. Bellevue, WA, USA: AAAI Press, 2013, p. 2.

[24] H. Guo, F. Nie, X. Li, and H. Huang, “Multi-view subspace clustering,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile: IEEE Computer Society, Dec. 2015, pp. 4238–4246.

[25] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, “Optimal neighborhood kernel clustering with multiple kernels,” in *Proc. 31st AAAI Conf. Artif. Intell.*, S. P. Singh and S. Markovitch, Eds. San Francisco, CA, USA: AAAI Press, 2017, pp. 2266–2272.
[26] J. Liu, X. Liu, J. Xiong, Q. Liao, S. Zhou, S. Wang, and Y. Yang, “Optimal neighborhood multiple kernel clustering with adaptive local kernels,” IEEE Trans. Knowl. Data Eng., early access, Aug. 4, 2020, doi: 10.1109/TKDE.2020.3014104.

[27] B. Zhao, J. T. Kwok, and C. Zhang, “Multiple kernel clustering,” in Proc. SIAM Int. Conf. Data Mining (SDM), Sparks, NV, USA: SIAM, 2009, pp. 638–649.

[28] S. Lloyd, “Least squares quantization in PCM,” IEEE Trans. Inf. Theory, vol. IT-28, no. 2, pp. 129–136, Mar. 1982.

[29] I. S. Dhillon, Y. Guan, and B. Kulis, “Kernel K-means: Spectral clustering and normalized cuts,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel,Eds. Washington, DC, USA, 2004, pp. 551–556.

[30] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, “Multiple kernel fuzzy clustering,” IEEE Trans. Fuzzy Syst., vol. 20, no. 1, pp. 120–134, Feb. 2012.

[31] E. Z. Zhu, “L2 regularization for learning kernels,” in Proc. 25th Conf. Uncertainty Artif. Intell., J. A. Bilmes and A. Y. Ng. Eds. Montreal, QC, Canada: AUIA Press, Jun. 2009, pp. 109–116.

[32] M. Kloft and G. Blanchard, “The local rademacher complexity of lp-norm multiple kernel learning,” in Advances in Neural Information Processing Systems 24, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds. Granada, Spain, 2011, pp. 2438–2446.

[33] P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y. Shen, “Robust multiple kernel k-means using l21-norm,” in Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI), Q. Yang and M. J. Wooldridge, Eds. Buenos Aires, Argentina: AAAI Press, 2015, pp. 3476–3482.

[34] A. Kumar, P. Rai, and H. Daume, “Co-regularized multi-view spectral clustering,” in Proc. Adv. Neural Inf. Process. Syst., 2011, pp. 1413–1421.

[35] R. Xia, Y. Pan, L. Du, and J. Yin, “Robust multi-view spectral clustering via low-rank and sparse decomposition,” in Proc. 28th AAAI Conf. Artif. Intell., C. E. Brodley and P. Stone, Eds. Québec City, QC, Canada: AAAI Press, 2014, pp. 2149–2155.

[36] P. Zhou, L. Du, L. Shi, H. Wang, and Y. Shen, “Recovery of corrupted multiple kernels for clustering,” in Proc. 24th Int. Joint Conf. Artif. Intell. IJCAI, Q. Yang and M. J. Wooldridge, Eds. Buenos Aires, Argentina: AAAI Press, 2015, pp. 4105–4111.