IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 24, 2019, accepted September 8, 2019, date of publication September 11, 2019, date of current version October 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940896

Consensus Multiple Kernel K-Means Clustering With Late Fusion Alignment and Matrix-Induced Regularization

JINGTAO HU[1], MIAOMIAO LI[2], EN ZHU[1], SIWEI WANG[1], XINWANG LIU[1], (Member, IEEE), AND YONGPING ZHAI[3]

[1]College of Computer, National University of Defense Technology, Changsha 410073, China
[2]Department of Computer, Changsha College, Changsha 410073, China
[3]College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China

Corresponding author: Yongping Zhai (zhaiyongping08@nudt.edu.cn)

This work was supported by the National Key R&D Program of China 2018YFB1003203 and the National Natural Science Foundation of China (Grant NO.61672528, NO. 61773392, NO. 61906020).

ABSTRACT Multiple kernel clustering (MKC) attracts considerable attention due to its competitive performance in unsupervised learning. However, we observe that most of the existing MKC approaches do not sufficiently consider the correlation between different clustering partitions. As a result, the existing methods would cause redundant and low diversity of selected clustering partitions which deteriorate clustering performance. To address these issues, we propose an effective and efficient multiple kernel k-means clustering method termed *Consensus Multiple Kernel Clustering with Late Fusion Alignment and Matrix-Induced Regularization (CMKC-LFA-MR)*. Specifically, the correlations between different clustering partitions are calculated as a matrix-induced regularization to encourage the diversity of clustering results. Moreover, we propose to maximally align the consensus partition with the weighted base partitions. The proposed algorithm jointly optimizes the basic clustering partitions and the optimal consensus clustering result. To solve the resultant optimization problem, a three-step alternate algorithm is proposed with both theoretically and experimentally proved convergence. As demonstrated by the experiments on six benchmark datasets, our algorithm outperforms the existing state-of-the-art multi-kernel methods in clustering performance with less time complexity, which demonstrates the effectiveness and efficiency of our proposed algorithm.

INDEX TERMS Multiple kernel clustering, late fusion, kernel method.

I. INTRODUCTION

Clustering is one of the most fundamental learning tasks in machine learning and data mining fields. Nowadays, many real-world data are represented by various heterogeneous features or views, which are generated from multiple sources of data collection or feature construction ways. For example, images can be represented by its color and shape descriptors, and web-page data can be described by using images, text and hyper links. Normally, these multiple sources of information encode complementary information, which motivates the development of multiple kernel clustering (MKC) whose goal is to explore such information for improvements on clustering performance [1]–[27]. Existing researches on multiple kernel clustering in this field can be summarized into three categories. The first one learns a latent consensus matrix via a low-rank optimization framework from various views [28]–[30]. In [28], it is proposed to learn a latent low-rank transition probability matrix shared from the given kernels as the input to the standard Markov chain for clustering. The work in [29] captures the noises in each kernel and integrate them into a low-rank framework. By following multiple kernel learning framework, the second category optimizes the into a low-rank framework. By following multiple kernel learning framework, the second category optimizes the optimal kernel matrix as a linear combination of known kernel matrices from a given library In [1], a three-step alternate algorithm is proposed to jointly optimize clustering, kernel coefficients and dimension reduction. The work in [8] proposes a multiple kernel k-means clustering algorithm with matrix-induced regularization to reduce the redundancy of the pre-defined k-means clustering algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

with matrix-induced regularization to reduce the redundancy of the the clustering performance in [9]. In [10], a multiple kernel algorithm is proposed to allow the optimal kernel to reside in the neighborhood of the optimal kernels. The last category methods are implemented via a late fusion-based manner, which obtain final clustering by combining the results generated from each view [11], [32], [33]. Similar to ensemble clustering, the work in [32]–[36] propose to fuse multiple clustering results from kernels into a consensus clustering partition instead of fusing kernels in advance. Our method in this paper belongs to the third category.

Although the aforementioned algorithms have been proposed to improve multiple kernel clustering from different aspects, we observe that they suffer from the following drawbacks. i) The intensive computational complexity, i.e., usually $\mathcal{O}(n^3)$ per iteration the number of samples n, prevents the first two kinds' algorithms from being applied into medium or large-scale clustering tasks [11], [37]. ii) The ℓ_1 norm constraint on weights in [32] leads to sparse solution, which reduces the diversity of selected partitions and leads to unsatisfying clustering performance.

To address these issues, we propose a novel algorithm termed *Consensus Multiple Kernel Clustering with Late Fusion Alignment Maximization and Matrix-induced Regularization (CMKC-LFA-MR)* in this paper. Firstly, it maximizes the alignment between the consensus partition and the weighted base partition matrices with orthogonal transformation, where each base partition is generated by performing clustering on each single view. The proposed CMKC-LFA-MR jointly optimizes the rotation matrices, weight coefficients and the optimal consensus partition. Moreover, we develop an efficient algorithm to solve the resultant optimization problem, and theoretically analyze its computational complexity and convergence. Extensive experiments on six multiple-kernel benchmark datasets are conducted to evaluate the effectiveness and efficiency of our proposed method, including the clustering performance, the running time and the objective value with iterations. As demonstrated, the proposed algorithm enjoys superior clustering performance with significant reduction in computational cost, in comparison with several state-of-the-art multi-kernel clustering methods.

The contributions of this paper are summarized as follows,

- The proposed CMKC-LFA-MR integrates multiple kernels via a late fusion manner. It jointly optimizes the consensus partition, rotation matrices and weight coefficients. Moreover, we design a matrix-induced regularization to consider the correlation of different partitions in the framework of multi-kernel clustering with late fusion.
- An alternate optimization algorithm with proved convergence is designed to efficiently tackle the resultant problem. By the virtue of it, CMKC-LFA-MR shows clearly superior clustering performance in comparison with state-of-the-art methods. Moreover, CMKC-LFA-MR requires significantly less computational time, especially essential for large datasets with limited computing sources.

TABLE 1. Main notations used in the paper.

Notation	Meaning
n	The number of samples
m	The number of kernel matrices or views
k	The number of clusters
$\mathbf{K}_i$	The i-th view kernel matrix
$\mathbf{H}_i$	The i-th view's basic partition matrix
$\mathbf{H}^*$	The consensus clustering partition matrix
$\boldsymbol{\mu}$	The weight coefficient of kernels
$\mathbf{W}_i$	The i-th view's rotation matrix
$\mathbf{Z}$	The clustering indicator matrix
$\boldsymbol{\upsilon}_c$	The centroid of c-th cluster

The rest of this paper is organized as follows. Section II introduces some notations in our paper. Section III outlines the related work of multiple kernel clustering. Section IV presents the Multiple-kernel k-means with late fusion Alignment and Matrix-Induced Regularization (CMKC-LFA-MR) convergence and the computational complexity of our proposed algorithm. Section VI shows the experimental results with evaluation. Section VII concludes the paper.

II. PRELIMINARIES

In this section, we introduce some necessary notations and preliminaries in our paper. Throughout this paper, we use boldface uppercase and lowercase letters to denote matrices and vectors respectively. The (i, j)-th elements of a matrix $\mathbf{M}$ is referred as $\mathbf{M}_{ij}$ and the i-th element of a vector $\boldsymbol{\mu}$ is denoted as $\boldsymbol{\mu}_i$. We denote $\mathrm{Tr}(\mathbf{K})$ as the trace norm of a kernel matrix $\mathbf{K}$. The notations are summarized in Table 1.

III. RELATED WORKS

A. KERNEL k-MEANS CLUSTERING(KKM)

Kernel k-means clustering transfers the original data into high-dimensional or infinite space serving for clustering tasks. The optimization goal of kernel k-means clustering algorithm is to minimize the square loss of the within-cluster distance in kernel space. And the kernel function $\phi(x)$ transfers the origin sample x onto a reproducing kernel Hilbert space $\mathcal{H}$ which is a k-means-friendly space and easier to cluster. By supposing the cluster indicator matrix $Z \in \{0,1\}^{n\times k}$, the optimization objective of KKM could be written as follows:

$$\min_{Z\in\{0,1\}^{n\times k}} \mathbf{Z}_{ic}\|\phi(X_i) - \boldsymbol{\upsilon}_c\|^2 \quad s.t. \sum_{c=1}^{k} \mathbf{Z}_{ic} = 1, \tag{1}$$

where $n_c = \sum_{i=1}^{n} \mathbf{Z}_{ic}$ and $\boldsymbol{\upsilon}_c = \frac{1}{n_c}\sum_{i=1}^{n} \mathbf{Z}_{ic}\phi(X_i)$ are the number and centroid of the $c - th(1 \le c \le k)$ cluster respectively.

By equivalently rewritten in matrix-vector form, the function in Eq. (1) is transformed to the following problem,

$$\min_{Z\in\{0,1\}^{n\times k}} \mathrm{Tr}(\mathbf{K}) - \mathrm{Tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{Z}^\top\mathbf{K}\mathbf{Z}\mathbf{L}^{\frac{1}{2}}) \quad s.t.\ \mathbf{Z}1_k = 1_n. \tag{2}$$

Here, we apply the kernel matrix to the Eq. (1), and $\mathbf{K}$ denotes the kernel matrix and $\mathbf{L} = diag([n_1^{-1}, n_2^{-1}, \cdot, \cdot, \cdot, n_k^{-1}])$.

Directly solving the optimization problem in Eq. (2) is difficult for the reason that the elements in matrix $\mathbf{L}$ are discrete. We relax $\mathbf{L}$ to take real values, by letting the new matrix $\mathbf{H}$ follows that $\mathbf{H} = \mathbf{Z}\mathbf{L}^{\frac{1}{2}}$. Then we rewrite the problem in Eq. (2),

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \mathrm{Tr}(\mathbf{K}(\mathbf{I_n} - \mathbf{H}\mathbf{H}^\top)) \quad s.t.\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{3}$$

Compared with the formula set in Eq. (1), the kernel k-means can correctly identify and extract a far more varied collection of cluster structures than the linear k-means clustering algorithm through the non-linear feature mapping. The optimization problem in Eq. (3) could be solved by singular value decomposition (SVD) of the kernel matrix $\mathbf{K}$ mentioned in [8].

However, the clustering performance of kernel k-means mostly depends on the pre-specified kernel matrix. For most applications in real life, it is hard for researchers to set a proper kernel matrix in advance. advance. To address this issue, multiple-kernel k-means clustering is proposed to enhance the representation ability of kernel k-means in a adaptively-weighted manner.

B. MULTI-KERNEL K-MEANS (MKKM)

In multiple kernel setting, we suppose that $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ is a collection of n samples, and $\phi_p(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}_p$ be the p-th feature mapping which transfers $\mathbf{x}$ into a reproducing kernel Hilbert space $\mathcal{H}_p$ $(1 \le p \le m)$. Hence each sample is represented as $\phi_{\boldsymbol{\mu}}(\mathbf{x}) = [\mu_1\phi_1(\mathbf{x})^\top, \cdots, \mu_m\phi_m(\mathbf{x})^\top]^\top$ from m views, where $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_m]^\top$ consists of the coefficients of the m base kernels $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$. The coefficients will be optimized during learning. Based on the definition of $\phi_{\boldsymbol{\mu}}(\mathbf{x})$, a kernel function can be expressed as

$$\kappa_{\boldsymbol{\mu}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\mu}}(\mathbf{x}_i)^\top\phi_{\boldsymbol{\beta}}(\mathbf{x}_j) = \sum_{p=1}^m \boldsymbol{\mu}_p^2\kappa_p(\mathbf{x}_i, \mathbf{x}_j). \tag{4}$$

A kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$ is then calculated by applying the kernel function $\kappa_{\boldsymbol{\mu}}(\cdot,\cdot)$ into $\{\mathbf{x}_i\}_{i=1}^n$. By using the notation that kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$, the optimization goal of MKKM algorithm can be expressed as

$$\begin{aligned}&\min_{\mathbf{H},\boldsymbol{\mu}} \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\mu}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top))\\ &s.t.\ \mathbf{H}\in\mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k,\ \boldsymbol{\mu}^\top\mathbf{1}_m = 1,\ \boldsymbol{\mu}_p \ge 0,\ \forall p.\end{aligned} \tag{5}$$

where $\mathbf{I}_k$ is an identity matrix with size $k \times k$. The optimization problem in Eq. (5) can be solved by alternately updating $\mathbf{H}$ and $\boldsymbol{\beta}$: i) **Optimizing H by fixed $\boldsymbol{\mu}$**. With the kernel coefficients $\boldsymbol{\mu}$ fixed, $\mathbf{H}$ can be obtained by solving a kernel k-means clustering optimization problem shown in Eq. (6);

$$\max_{\boldsymbol{H}} \mathrm{Tr}(\mathbf{H}^\top\mathbf{K}_{\boldsymbol{\mu}}\mathbf{H}) \quad s.t.\ \mathbf{H}\in\mathbb{R}^{n\times k}, \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{6}$$

The optimal $\mathbf{H}$ for Eq. (6) can be obtained by taking the k eigenvectors corresponding to the largest k eigenvalues of $\mathbf{K}_{\boldsymbol{\mu}}$. ii) **Optimizing $\boldsymbol{\mu}$ by fixed H**. With $\mathbf{H}$ fixed, $\boldsymbol{\mu}$ can be optimized via solving the following quadratic programming with linear constraints,

$$\min_{\boldsymbol{\mu}} \sum_{p=1}^m \boldsymbol{\mu}_p^2\mathrm{Tr}(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \quad s.t.\ \boldsymbol{\mu}^\top\mathbf{1}_m = 1,\ \boldsymbol{\mu}_p \ge 0. \tag{7}$$

As noted in [3] and [6], using a convex combination of kernels $\sum_{p=1}^m \boldsymbol{\mu}_p\mathbf{K}_p$ to replace $\mathbf{K}_{\boldsymbol{\mu}}$ in Eq. (5) is not a valid option. With the norm constraint on $\boldsymbol{\mu}$, the solution could be sparse and only one single kernel is activated while others are given with zero weights. Along with this line, many variants of MKKM have been proposed in the literature [8], [9], [18], [38]. Liu *et al.* proposes a multiple kernel k-means clustering algorithm with matrix-induced regularization to reduce the redundancy and redundancy and enhance the diversity of the pre-defined kernels [8]. Furthermore, local kernel alignment criterion has been applied to multiple kernel learning to enhance the clustering performance in [9].

C. MULTIPLE-KERNEL CLUSTERING WITH LOCAL KERNEL ALIGNMENT MAXIMIZATION(MKC-LKA)

In [9], the local kernel alignment criterion has been applied to multiple kernel learning following the motivation that the similar sample pairs shall stay more closer and the similarity evaluations for farther sample pairs are unreliable because of improper metric settings. Considering locally aligning the similarity of each sample to its k-nearest neighbors with corresponding ideal kernel matrix, in specific, the local kernel alignment for the i-th can be calculated as,

$$\max_{\mathbf{H}\in\mathbb{R}^{n\times k}, \boldsymbol{\mu}\in\mathbb{R}_+^m} \frac{\langle\mathbf{K}_{\boldsymbol{\mu}}^{(i)}, \mathbf{H}^{(i)}\mathbf{H}^{(i)\top}\rangle}{\sqrt{\langle\mathbf{K}_{\boldsymbol{\mu}}^{(i)}, \mathbf{K}_{\boldsymbol{\mu}}^{(i)}\rangle}} \quad s.t.\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \boldsymbol{\mu}^\top\mathbf{1}_m = 1. \tag{8}$$

where $\langle\mathbf{K}_{\boldsymbol{\mu}}^{(i)}, \mathbf{H}^{(i)}\mathbf{H}^{(i)\top}\rangle = \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\mu}}^{(i)\top}\mathbf{H}^{(i)}\mathbf{H}^{(i)\top})$, $\mathbf{K}_{\boldsymbol{\mu}}^{(i)}$ and $\mathbf{H}^{(i)}$ are the corresponding sub-matrix of $\mathbf{K}_{\boldsymbol{\mu}}$ and $\mathbf{H}$ whose indexes are specified by the τ-nearest neighbors of the i-th sample. For more details, please refer to the [9].

The Eq. (8) can be conceptually expressed as,

$$\begin{aligned}\min_{\mathbf{H}\in\mathbb{R}^{n\times k}, \boldsymbol{\mu}\in\mathbb{R}_+^m} &\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\mu}}^{(i)}(\mathbf{I}_\tau - \mathbf{H}^{(i)}\mathbf{H}^{(i)\top})) + \frac{\lambda}{2}\boldsymbol{\mu}^\top\mathbf{M}^{(i)}\boldsymbol{\mu}\\ &s.t.\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \boldsymbol{\mu}^\top\mathbf{1}_m = 1.\end{aligned} \tag{9}$$

where $\mathbf{K}_{\boldsymbol{\mu}}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{K}_{\boldsymbol{\mu}}\mathbf{S}^{(i)}$, $\mathbf{H}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{H}$, $\mathbf{S}^{(i)} \in \{0,1\}^{n\times\tau}$ is a matrix indicating the τ-nearest neighbors of the i-th sample and $\mathbf{I}_\tau$ is an identity matrix with size $\tau\times\tau$. $\mathbf{M}^{(i)}$ is a correlation matrix with $\mathbf{M}_{pq}^{(i)} = \mathrm{Tr}(\mathbf{K}_p^{(i)\top}\mathbf{K}_q^{(i)})$

However, we observe that the intensive computational complexity of MKC-LFA is $\mathcal{O}(n^3)$ per iteration, prevents it from being applied into medium or large-scale clustering tasks.

IV. CONSENSUS MULTIPLE-KERNEL K-MEANS WITH LATE FUSION ALIGNMENT AND MATRIX-INDUCED REGULARIZATION(CMKC-LFA-MR)

In this section, we firstly propose a simple but effective multiple kernel clustering algorithm termed *Consensus Multiple-kernel k-means with late fusion Alignment and Matrix-Induced Regularization(CMKC-LFA-MR)*. In specific, CMKC-LFA-MR proposes to maximally align the consensus partition with the weighted base partitions and is regularized to reduce the redundancy and enforce the diversity of the selected base partitions.

As mentioned in [32], we complete our multiple-kernel k-means with late fusion as follows,

$$\begin{aligned}\max_{\mathbf{H}^*,\{\mathbf{W}_p\}_{p=1}^m,\boldsymbol{\mu}} & \operatorname{Tr}(\mathbf{H}^{*\mathrm{T}}\sum\nolimits_{p=1}^m \boldsymbol{\mu}_p\mathbf{H}_p\mathbf{W}_p)+\lambda\operatorname{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{H}_0),\\ s.t.\ & \mathbf{H}^{*\mathrm{T}}\mathbf{H}^*=\mathbf{I}_k,\mathbf{W}_p^{\mathrm{T}}\mathbf{W}_p=\mathbf{I}_k,\boldsymbol{\mu}^{\mathrm{T}}1_m=1,\boldsymbol{\mu}_p\geq 0,\end{aligned} \tag{10}$$

where $\mathbf{X}$ and $\mathbf{H}^*$ are the data and the consensus partition matrix respectively. The latter $\operatorname{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{H}_0)$ is a regularization on the consensus partition to prevent $\mathbf{H}^*$ from being too far way from prior average partition. To be easily extended into multiple kernel clustering, after obtaining the basic partitions $\{\mathbf{H}_p\}_{p=1}^m$ from each single view, we conduct the new consensus partition $\sum_{p=1}^m \boldsymbol{\mu}_p\mathbf{H}_p\mathbf{W}_p$.

To further analysis in depth, we assume that $\mathbf{H}_p\mathbf{W}_p$ is selected and assigned to a large weight.As can be seen from Eq. (10), the ℓ_1 norm constraint on $\boldsymbol{\mu}$ leads to sparse solution, which increases the redundancy and reduce the diversity of selected partitions. According to Eq. (10),the rotated $\mathbf{H}_q\mathbf{W}_q$ with high correlation with $\mathbf{H}_p\mathbf{W}_p$ would be also selected together and assigned to similar important weights. This would result in the high redundancy among the given partitions. On the other hand, the selection of highly correlated partitions could suppress the weights of partitions that are less correlated with $\mathbf{H}_p$ due to the sparsity constraint (ℓ_1 norm) imposed on the weights.This would cause the low diversity among the selected partitions or even prevent complementary partitions from being utilized.

To reduce the redundancy and enforce the diversity of the selected kernels, we need a regularization term that is able to characterize the correlation of each pair of rotated partitions.

Motivated by the work of [8], we firstly define a correlation $\mathcal{M}(\mathbf{H}_p,\mathbf{H}_q)$ between $\mathbf{H}_p\mathbf{W}_p$ and $\mathbf{H}_q\mathbf{W}_q$. Larger value of $\mathcal{M}(\mathbf{H}_p,\mathbf{H}_q)$ represents high correlation between $\mathbf{H}_p$ and $\mathbf{H}_q$. Based on our prior work [8], we introduce the matrix-induced regularization terms as,

$$\min_{u\in\mathbb{R}_+^m}\sum_{p,q=1}\mu_p\mu_q M_{pq}=\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}, \tag{11}$$

where $\mathbf{M}$ is a matrix with $\mathbf{M}_{pq}=\operatorname{Tr}(\mathbf{W}_p^\top\mathbf{H}_p^\top\mathbf{H}_q\mathbf{W}_q)$.

A. PROPOSED FORMULATION

By integrating the matrix-induced regularization into the objective function, we derive a novel optimization formulation for multiple-kernel clustering. In general, after obtaining the basic partitions $\{\mathbf{H}_p\}_{p=1}^m$ from every single view, we conduct the new optimal combinational partition $\sum_{p=1}^m \boldsymbol{\mu}_p\mathbf{H}_p\mathbf{W}_p$ to maximally align with the consensus partition.

As a result, we obtain the objective function of our adaptively-weighted algorithm follows,

$$\begin{aligned}\max_{\mathbf{H}^*,\{\mathbf{W}_p\}_{p=1}^m,\boldsymbol{\mu}} & \operatorname{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X})+\lambda\operatorname{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{H}_0)-\frac{\beta}{2}\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu},\\ s.t.\ & \mathbf{H}^{*\mathrm{T}}\mathbf{H}^*=\mathbf{I}_k,\mathbf{W}_p^{\mathrm{T}}\mathbf{W}_p=\mathbf{I}_k,\mathbf{X}=\sum\nolimits_{p=1}^m\boldsymbol{\mu}_p\mathbf{H}_p\mathbf{W}_p,\end{aligned} \tag{12}$$

where $\{\mathbf{W_p}\}_{p=1}^m$ are a set of rotation matrices, $\mathbf{H}_0$ denotes the average partition region and λ is a trade-off parameter. The latter $\operatorname{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{H}_0)$ is a regularization on the consensus partition to prevent $\mathbf{H}^*$ from being too far way from prior average partition. It is worth noting that we not only set an optimization goal set an optimization goal for the multi-view clustering with late fusion, but also offer a new framework to fuse various clustering methods, which implies that any kind of ensemble clustering results can be applied to our framework. Moreover, as the following optimization process shows, the proposed function could be easily solved by an alternate algorithm with proved convergence.

B. OPTIMIZATION FOR ADAPTIVE ALGORITHM

In order to solve the consultant optimization problem in Eq. (12), we design a three-step alternate alternate optimization algorithm with theoretically-proved convergence, where each step could be easily solved by the existing packages.

1) OPTIMIZATION H^* WITH FIXED $\{\mathrm{W}_p\}_{p=1}^m$ AND μ

With $\{\mathbf{W}_p\}_{p=1}^m$ and $\boldsymbol{\mu}$ being fixed, the optimization Eq. (12) could be rewritten as follows,

$$\begin{aligned}&\max_{\mathbf{H}}\operatorname{Tr}(\mathbf{H}^\top\mathbf{U})\\ &s.t.\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k,\end{aligned} \tag{13}$$

where $\mathbf{U}=\sum_{p=1}^m\boldsymbol{\mu}_p\mathbf{H}_p\mathbf{W}_p+\lambda\mathbf{H}_0$. And this problem in Eq. (13) could be easily solved by taking the singular value decomposition(SVD) of the given matrix $\mathbf{U}$. Here the following Theorem gives a closed-form solution for the problem in Eq. (13).

Theorem 1: Suppose that the matrix $\mathbf{U}$ *in Eq. (13) has the economic rank-k singular value decomposition form as* $\mathbf{U}=\mathbf{S}_k\Sigma_k\mathbf{V}_k^\top$, *where* $\mathbf{S}_k\in\mathbb{R}^{n\times k}$, $\Sigma_k\in\mathbb{R}^{k\times k}$, $\mathbf{V}_k\in\mathbb{R}^{k\times k}$. *The optimization in Eq. (13) has a closed-form solution as follows,*

$$\mathbf{H}^*=\mathbf{S}_k\mathbf{V}_k^{\mathrm{T}}, \tag{14}$$

Proof: By taking the the normal singular value decomposition $\mathbf{U}=\mathbf{S}\Sigma\mathbf{V}^{\mathrm{T}}$, the Eq. (13) could be rewritten as,

$$\operatorname{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{S}\Sigma\mathbf{V}^{\mathrm{T}})=\operatorname{Tr}(\mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}\Sigma). \tag{15}$$

Considering that $\mathbf{Q} = \mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}$, then we have $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}\mathbf{S}^{\mathrm{T}}\mathbf{H}^{*}\mathbf{V} = \mathbf{I}_k$. Therefore we can take $\mathrm{Tr}(\mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}\Sigma) = \mathrm{Tr}(\mathbf{Q}\Sigma) \leq \sum_{i=1}^{k}\sigma_i$. Hence in order to maximize the value of Eq. (13), the solution should be given as Eq. (14). This completes the proof. □

2) OPTIMIZATION $\{\mathrm{W}_p\}_{p=1}^m$ WITH FIXED H* AND μ

With $\mathbf{H}^*$ and $\boldsymbol{\mu}$ being fixed, for each single $\mathbf{W}_p$, the optimization problem in Eq. (12) is rewritten as follows,

$$\max_{\mathbf{W}_p} \mathrm{Tr}(\mathbf{W}_p^{\mathrm{T}}\mathbf{T}) \;\; s.t.\; \mathbf{W}_p^{\mathrm{T}}\mathbf{W}_p = \mathbf{I}_k, \tag{16}$$

where $\mathbf{T} = \boldsymbol{\mu}_p\mathbf{H}_p^{\mathrm{T}}\mathbf{H}^*$. And this problem in Eq.(16) could be easily solved by taking the singular value decomposition (SVD) of the given matrix $\mathbf{V}$. Like the closed-form expressed in Theorem 1, if the matrix $\mathbf{V}$ has the singular value decomposition form as $\mathbf{A} = \mathbf{S}\Sigma\Sigma\mathbf{G}^{\mathrm{T}}$, the optimization in Eq.(16) has a closed-form solution as $\mathbf{W}_p = \mathbf{S}\mathbf{G}^{\mathrm{T}}$. Hence we optimize one $\mathbf{W}_p$ with other $\mathbf{W}_{i\neq p}$ fixed at each iteration. As a result, we can obtain a set of optimized $\{\mathbf{W}_p\}_{p=1}^m$.

3) OPTIMIZATION μ WITH FIXED H* AND $\{\mathrm{W}_p\}_{p=1}^m$

With $\mathbf{H}^*$ and $\{\mathbf{W}_p\}_{p=1}^m$ being fixed, the optimization problem in Eq. (12) is equivalent to the optimization problem as follows,

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \frac{\beta}{2}\boldsymbol{\mu}^{\top}\mathbf{M}\boldsymbol{\mu} - \mathbf{f}^{\top}\boldsymbol{\mu}, \\ &s.t.\; \boldsymbol{\mu}^{\top}1 = 1, \boldsymbol{\mu} \geq 0, \end{aligned} \tag{17}$$

where $f = [f_1, f_2, \ldots, f_m]$ with $f_p = \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{H}_p\mathbf{W}_p)$, $\mathbf{M}_{pq} = \mathrm{Tr}(W_p^{\top}H_p^{\top}H_qW_q)$.

It seems difficult to solve the Eq. (17). However the following proof illustrates the matrix $\mathbf{M}$ is a positive semidefinite (PSD) matrix. Hence, with the simplified problem proposed in Eq. (17), we have observed that this problem is a quadratic programming optimization and could be efficiently solved via the existing convex optimization package.

Lemma 1: for every $x \in \mathbb{R}^m$, we have that

$$\begin{aligned} x^{\top}\mathbf{M}x &= \sum_{p=1}^{m}\sum_{q=1}^{m} x_px_q\mathrm{Tr}(W_p^{\top}H_p^{\top}H_qW_q), \\ &= \mathrm{Tr}(\sum_{p=1}^{m}\sum_{q=1}^{m} x_px_qW_p^{\top}H_p^{\top}H_qW_q), \\ &= \mathrm{Tr}(\sum_{p=1}^{m} x_pW_p^{\top}H_p^{\top}\sum_{q=1}^{m} x_qH_qW_q), \\ &= \left\|\sum_{p=1}^{m} x_pw_p^{\top}H_p^{\top}\right\|_F^2 \geq 0. \end{aligned} \tag{18}$$

Therefore, the matrix $\mathbf{M}$ is a positive semi-definite matrix and the optimization in Eq. (17) could be solved by quadratic programming.

Our algorithm termed CMKC-LFA-MR is outlined in Algorithm 1, where obj$^{(t)}$ denotes the objective value at the t-th iteration.

Algorithm 1 CMKC-LFA-MR

1: **Input**: $\{\mathbf{H}_p\}_{p=1}^m$ and ϵ_0.
2: **Output**: $\mathbf{H}^*$, $\boldsymbol{\mu}$.
3: Initialize $\{\mathbf{W}_p\}_{p=1}^m = \mathbf{I}_k$, $\boldsymbol{\mu} = \frac{1}{m}$ and $t = 1$.
4: **Repeat**
5: Update $\mathbf{H}^*$ by solving Eq.(13) with fixed $\{\mathbf{W}_p\}_{p=1}^m$ and $\boldsymbol{\mu}$.
6: Update $\{\mathbf{W}_p\}_{p=1}^m$ with fixed $\mathbf{H}^*$ and $\boldsymbol{\mu}$ by Eq.(16).
7: Update $\boldsymbol{\mu}$ by solving Eq.(17) with fixed $\mathbf{H}^*$ and $\{\mathbf{W}_p\}_{p=1}^m$.
8: $t = t + 1$.
9: **Until** $\left(\mathbf{obj}^{(t-1)} - \mathbf{obj}^{(t)}\right)/\mathbf{obj}^{(t)} \leq \epsilon_0$

V. ALGORITHM ANALYSIS

In this section, we show our theoretical analysis on the proposed algorithm's convergence and computational complexity to verify the efficiency of proposed algorithm.

A. CONVERGENCE ANALYSIS

The following Theorem 2 shows our algorithm is guaranteed to converge into a local minimum.

Theorem 2: The proposed algorithm 1 is proved to converge to a local optimum.

Proof: Note that for $\forall p, q$, $\mathrm{Tr}[(\boldsymbol{\mu}_p\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\boldsymbol{\mu}_q\mathbf{H}_q\mathbf{W}_q)] \leq \mathrm{Tr}[(\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\mathbf{H}_q\mathbf{W}_q)] \leq \frac{1}{2}(\mathrm{Tr}[(\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\mathbf{H}_p\mathbf{W}_p)] + \mathrm{Tr}[(\mathbf{H}_q\mathbf{W}_q)^{\mathrm{T}}(\mathbf{H}_q\mathbf{W}_q)]) = k$. As a result, we could derive the upper bound of the optimization goal in Eq.(12). We obtain that $\mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}) \leq \frac{1}{2}(\mathrm{Tr}[\mathbf{H}^{*\mathrm{T}}\mathbf{H}^*] + \mathrm{Tr}[\mathbf{X}^{\mathrm{T}}\mathbf{X}]) = \frac{1}{2}(\mathrm{Tr}[\mathbf{H}^{*\mathrm{T}}\mathbf{H}^*] + \mathrm{Tr}(\sum_{p,q=1}^{m}(\boldsymbol{\mu}_p\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\boldsymbol{\mu}_q\mathbf{H}_q\mathbf{W}_q))) \leq \frac{k}{2}(m^2+1)$. Meanwhile, the $(\mathbf{H}^{*\mathrm{T}}\mathbf{H}_0) \leq \frac{1}{2}(\mathrm{Tr}[\mathbf{H}^{*\mathrm{T}}\mathbf{H}^*] + \mathrm{Tr}[\mathbf{H}_0{}^{\mathrm{T}}\mathbf{H}_0]) = k$. As for the last term $-\frac{\beta}{2}\boldsymbol{\mu}^{\top}\mathbf{M}\boldsymbol{\mu}$, with Lemma 1 proved, $-\frac{\beta}{2}\boldsymbol{\mu}^{\top}\mathbf{M}\boldsymbol{\mu} \leq 0$ (since $\beta \geq 0$). Consequently, the whole optimization function is upper bounded by $\frac{k}{2}(m^2+1)+k$. As the three subproblems are strictly convex when optimizing one variable and keeping the others fixed. The objective of Algorithm 1 is monotonically increased when optimizing one variable with the others fixed at each iteration. At the same time, the whole optimization problem is upper-bounded. As a result, the proposed algorithm can be verified to be convergent. This completes the proof. □

B. COMPUTATIONAL COMPLEXITY

Since our algorithm completes multiple kernel clustering via late fusion manner, comparing to the early-fusion method, our algorithm has less time complexity. And in this section, we theoretically analyze the time complexity of the proposed algorithm.

Theoretically, we assume that the number of samples in given datasets is n, the number of clusters k and the number of kernels is m. Going back to the our optimization algorithm in 1, the total time complexity consists of three parts referring to the three alternate steps. With the optimization process

TABLE 2. Multi-kernel datasets used in our experiments.

Dataset	#Samples	#Kernels	#Classes
Flower17	1360	7	17
ProteinFold	694	12	27
Caltech-10	1020	25	102
Caltech-15	1530	25	102
Caltech-20	2040	25	102
CCV	6773	3	20

outlined in Algorithm 1, the computational complexity of CMKC-LFA-MR is $\mathcal{O}(nk^2 + mk^3 + m^3)$ per iteration. This implies that our algorithm has a linearly growing complexity with the number of samples(since $n \gg k, m$), making it efficient to handle large-scale tasks comparing to the state-of-the-art multiple-kernel clustering algorithms.

C. DISCUSSION AND EXTENSIONS

CMKC-LFA-MR can be easily extended with the following considerations. Firstly, CMKC-LFA-MR could be further improved by capturing the noises or bad partition existing in basic partitions. For example, we could integrate the basic partitions $\{\mathbf{H}_p\}_{p=1}^m$ into the optimization procedure to capture more advanced base partitions. By doing so, the high-quality basic partitions are further used to guide the generation of consensus partition. Secondly, we could apply more similarity-based clustering methods to generate basic partitions. Further exploring other generating methods and evaluating their clustering performance will be an interesting future work.

VI. EXPERIMENTS

In this section, we evaluate the effectiveness and efficiency of the proposed CMKC-LFA-MR for six widely used multi-kernel benchmark datasets from the aspects of clustering performance, running time and convergence.

A. DATASETS AND EXPERIMENTAL SETTINGS

The datasets used in our experiments are Oxford Flower17,[1] Protein fold prediction(ProteinFold),[2] Columbia Consumer Video (CCV)[3] and Caltech.[4] The detailed information of the used datasets are listed in Table 2.

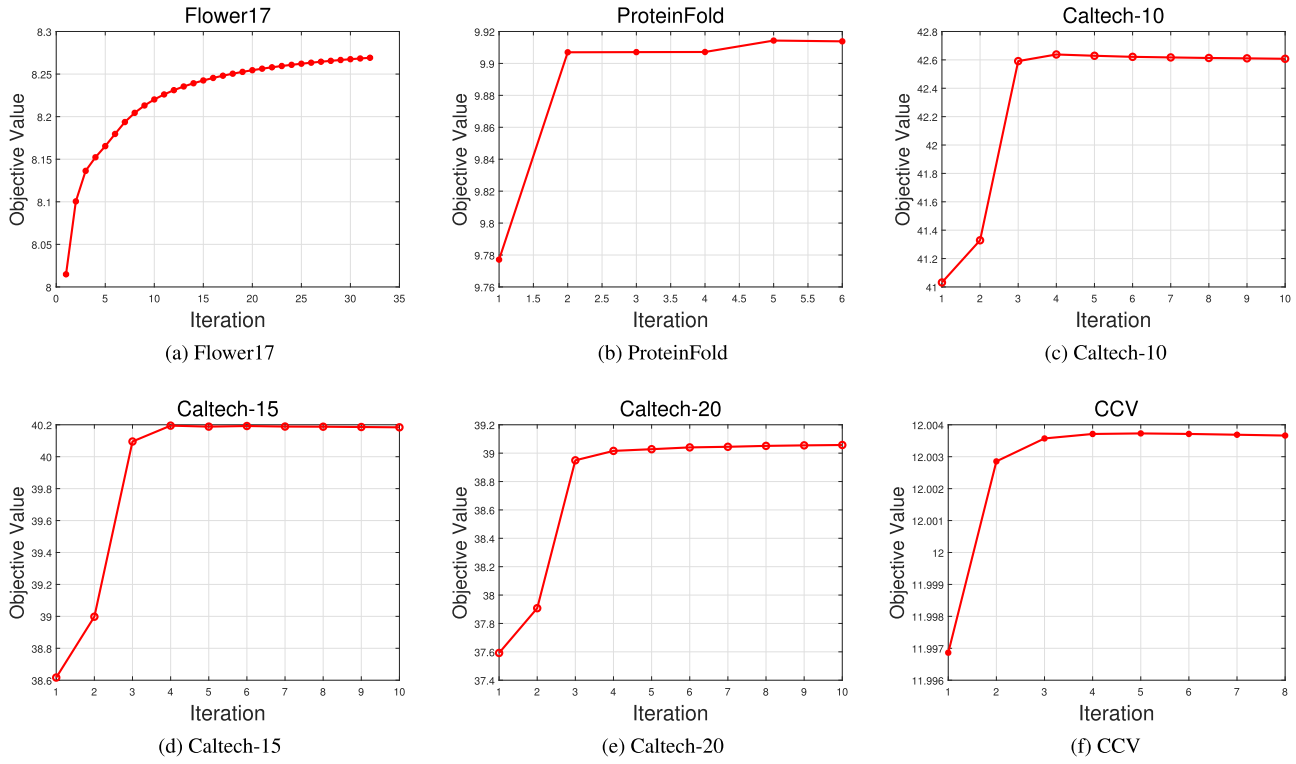
For Caltech datasets, we select 10, 15 and 20 samples randomly from each class. By this way, we generate three datasets on Caltech102, which each dataset has 102 classes and is represented by 48 base kernels. We term the three generated datasets as Caltech102-10, Caltech102-15 and Caltech102-20 respectively. For other benchmark multiple kernel datasets, we use the pre-defined kernel matrices and download them from the official website.

In all our experiments, suggested by [39],all base kernels are first centered and then scaled so that for all sample x_i and p, we have $K_p(x_i, x_i) = 1$. For all data sets, it is assumed that the true number of clusters is known and set as the true number of classes. For the proposed algorithm, the tradeoff

[1] http://www.robots.ox.ac.uk/~vgg/data/flowers/
[2] http://mkl.ucsd.edu/dataset/protein-fold-prediction
[3] http://www.ee.columbia.edu/ln/dvmm/CCV/
[4] http://www.vision.caltech.edu/Image_Datasets/Caltech101

TABLE 3. ACC, NMI and purity comparison of different clustering algorithms on six benchmark data sets.

Datasets	A-MKKM	SB-KKM	MKKM [4]	OKKC [3]	CSRC [40]	MKC-LKA [9]	MKKM-MR [8]	ONKC [10]	Proposed
ACC(%)									
Flower17	51.03	42.06	45.37	44.85	51.76	60.69	59.69	60.88	**61.25**
ProteinFold	30.69	34.58	27.23	37.10	35.59	39.34	36.89	39.90	**40.13**
Caltech-10	30.88	34.71	22.75	26.54	33.24	33.34	33.73	34.42	**37.54**
Caltech-15	31.11	34.44	20.39	25.39	34.38	32.16	32.35	32.25	**35.32**
Caltech-20	30.20	33.77	18.73	22.53	34.71	31.47	33.48	32.46	**35.47**
CCV	19.74	20.08	18.01	20.54	23.06	23.49	22.47	24.18	**26.56**
NMI(%)									
Flower17	50.19	45.14	45.35	45.85	53.19	57.27	57.11	58.58	**60.24**
ProteinFold	40.96	42.33	37.16	40.75	45.66	47.55	45.13	46.93	**49.96**
Caltech-10	61.77	63.15	55.80	52.48	61.89	62.57	62.86	63.16	**65,14**
Caltech-15	57.66	59.62	49.27	51.24	59.46	58.18	58.93	59.02	**60.21**
Caltech-20	54.19	56.79	45.61	48.76	56.77	55.96	56.34	55.78	**57.39**
CCV	17.16	17.73	15.52	16.28	18.89	17.11	18.62	18.24	**21.59**
Purity(%)									
Flower17	51.99	44.63	46.84	45.00	53.68	61.79	60.03	61.64	**62.15**
ProteinFold	37.18	41.21	33.86	39.91	42.07	45.97	43.80	45.24	**48.85**
Caltech-10	33.24	36.47	24.22	30.45	35.78	35.00	36.18	36.32	**38.45**
Caltech-15	32.81	36.54	21.63	26.45	36.73	33.73	34.05	33.96	**37.27**
Caltech-20	31.91	36.23	20.39	25.46	36.23	33.63	34.90	33.23	**37.39**
CCV	23.98	23.48	22.25	24.17	26.80	22.93	25.69	23.34	**27.71**



(a) Flower17 (b) ProteinFold (c) Caltech-10

(d) Caltech-15 (e) Caltech-20 (f) CCV

FIGURE 1. The objective value of our adaptive algorithm at each iteration in Flower17(a), ProteinFold(b), Caltech-10(c), Caltech-15(d), Caltech-20(e) and CCV(f).

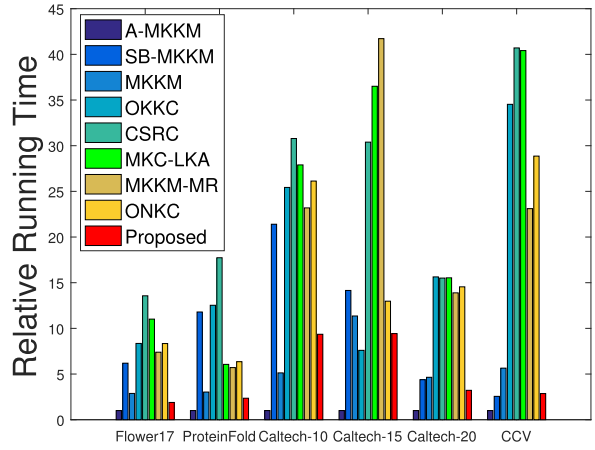


FIGURE 2. The running time comparison of different algorithms on six benchmark datasets.

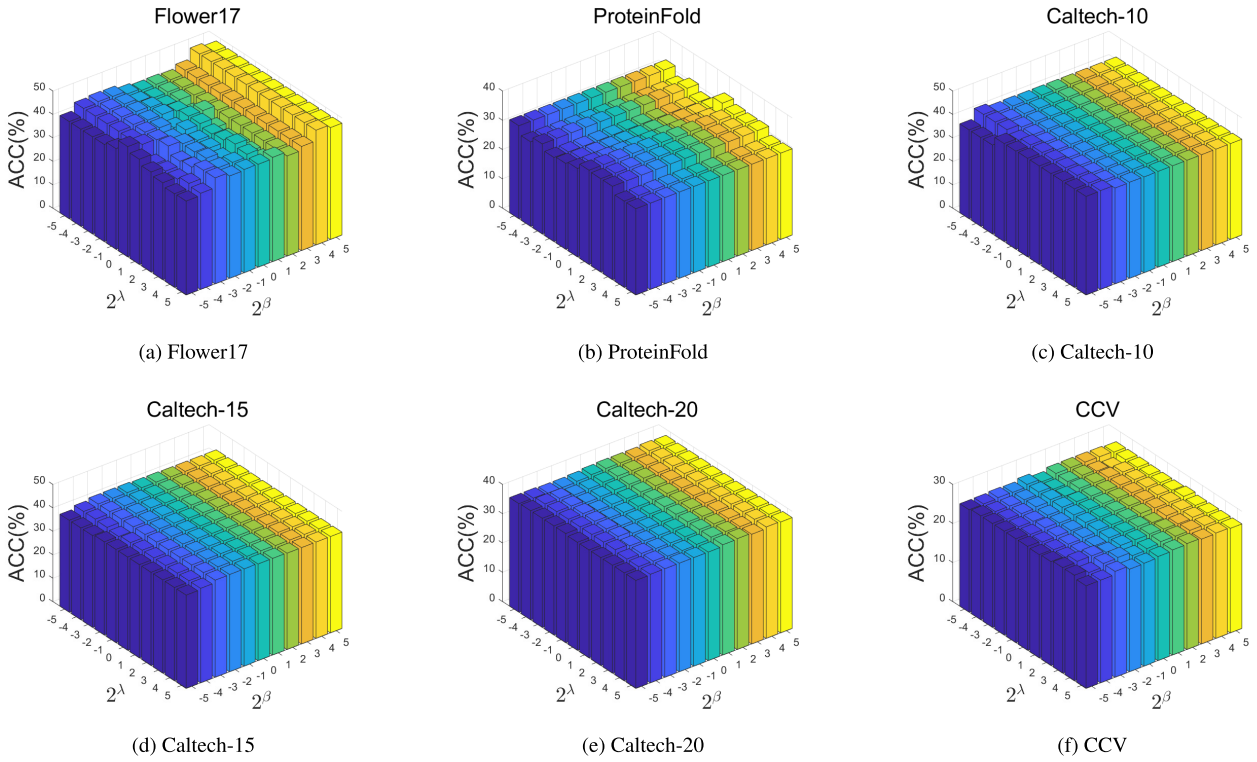
parameter λ and β are chosen from $\left[2^{-5}, 2^{-4}, \cdots, 2^{5}\right]$ by grid search.

The widely used clustering accuracy (ACC), normalized mutual information (NMI) and purity are applied to evaluate the clustering performance. For all algorithms, we repeat each experiment for 50 times with random initialization to reduce the effectiveness of randomness caused by k-means, and report the best result. All the experiments are performed on a desktop with Intel core i7-7820X CPU and 64G RAM.

B. COMPARED ALGORITHM

In this section, we list the compared algorithms as follows,

- Average multiple kernel k-means (A-MKKM): All kernels are averagely weighted to conduct the optimal kernel, which is used as the input of kernel k-means algorithm.
- Single best kernel k-means (SB-KKM): Kernel k-means is performed on each single kernel and the best result is outputted.
- Multiple kernel k-means (MKKM) [4]: The algorithm alternatively performs kernel k-means and updates the kernel coefficients, as introduced in the related work.
- Co-regularized spectral clustering (CRSC) [40]: CRSC provides a co-regularization way to perform spectral clustering on multiple views.
- Multiple kernel k-means with Matrix-induced Regularization (MKKM-MR) [8]: The The algorithm applies the multiple kernel k-means clustering with a matrix-induced regularization to reduce the redundancy and enhance the diversity of the kernels.
- Multiple Kernel Clustering with Local Kernel Alignment Maximization (MKC-LKA) [9]: The algorithm maximizes the local kernel alignment with multiple kernel clustering and focuses on closer sample pairs that they shall stay together.
- Optimal neighborhood kernel clustering with multiple kernels (ONKC) [10]: ONKC allows the optimal kernel



(a) Flower17 (b) ProteinFold (c) Caltech-10

(d) Caltech-15 (e) Caltech-20 (f) CCV

FIGURE 3. The ACC result of our MKC-LFA-MR method with different parameters' settings at benchmark datasets. Other metrics are similar.

to reside in the neighborhood of linear combination of base kernels and effectively enlarges the region from which an optimal kernel can be chosen, and therefore is in a better position than the traditional ones to identify a more suitable kernel for clustering.

The Matlab codes of A-MKKM, SB-KKM and MKKM are publicly available at `http://github.com/ mehmetgonen/lmkkmeans`. For the rest of algorithms, we use their Matlab implementations from authors' websites in our experiments.

C. EXPERIMENTAL RESULTS

The three evaluation metrics (ACC, NMI and Purity) of the compared algorithms on the six benchmark datasets are displayed in Table 3. Concluded from the results, we have the following observations:

- As can be observed from the results, the recently proposed ONKC ([10]) outperforms other early-fusion manner multiple kernel k-means clustering methods (MKKM, OKKC, CRSC, MKC-LKA, MKKM-MR and ONKC) in comparison. For example, it exceeds the second best approach (MKC-LKA) by 0.3%, 1.4%, 0.2%, 12.7%, 3.3% in terms of ACC on Flower17, ProteinFold, Caltech-10, Caltech-15, Caltech-20 and CCV, respectively. These results verify the effectiveness of enlarging the chosen region of optimal kernel matrix in a neighborhood way.
- The proposed algorithm CMKC-LFA-MR in red significantly and consistently outperforms ONNKC by **0.6**%, **0.5**%, **9.1**%, **9.5**%, **9.2**%, **9.8**% in terms of ACC on Flower17, ProteinFold, Caltech-10, Caltech-15, Caltech-20 and CCV, respectively.
- Comparing to the MKKM-MR, CMKC-LFA-MR consistently achieves higher performance among the benchmark datasets. Since both of the two approaches adopt the matrix-induced regularization, the experimental the experimental results illustrate that the late fusion manner benefits from high-level partition kernel level fusion.

Table 3 also reports the comparison of NMI and purity. As can be seen, our proposed algorithm outperforms all other methods in other metrics. The experimental results clearly demonstrate the effectiveness of our proposed algorithm.

Table 3 also reports the comparison of NMI and purity. Again, we observe that the proposed algorithm has promising performance among datasets. In all, these results have well verified the effectiveness of our proposed algorithm.

In summary, the above experimental results have well demonstrated the effectiveness of our proposed CMKC-LFA-MR comparing to other state-of-the-art methods. We attribute the superiority of CMKC-LFA-MR as three aspects:

- CMKC-LFA-MR adopts iterative fusion to update basic partitions, consensus partition and achieves better performance than other approaches. More specifically, when better consensus partition is obtained, we could

further make full use of the high-quality consensus to align with weighted basics partitions serving for clustering.
- Comparing with the existing early-fusion methods, the proposed CMKC-LFA-MR fuses multiple kernel information in the partition level, which demonstrates the benefits of fusing high-level information.
- Our proposed algorithm introduces a matrix-induced regularization term to enhance the diversity and reduce the redundancy of the selected basic partitions. Comparing to the sparse solution, CMKC-LFA-MR could utilize more high-quality partitions.

These three factors contribute to the significant improvements on clustering performance and update the consensus optimal partition better serving for clustering.

D. RUNNING TIME

To evaluate the computational efficiency of our proposed algorithm, we record the running time of these algorithms on the six benchmark datasets and report them in Figure 2. As can be seen, CMKC-LFA-MR has the shortest running time on all datasets comparing to the-state-of-art multiple kernel methods (MKKM, OKKC, CRSC, MKC-LKA, MKKM-MR and ONKC), demonstrating the computational efficiency of the proposed approach. As theoretically demonstrated, CMKC-LFA-MR reduces the time complexity from $\mathcal{O}(n)^3$ to $\mathcal{O}(n)$ per iteration and avoid complicated optimization procedure.

In sum, both the theory and the experimental results in Figure 2 have well demonstrated the computational advantage of CMKC-LFA-MR, making it efficient to handle with multiple kernel clustering.

E. CONVERGENCE AND PARAMETER SENSITIVITY

Our algorithm is theoretically guaranteed to converge according to Theorem 2. For the experimental study, we conduct experiments on the benchmark datasets. Furthermore, as shown in Figure 1, the objective value of CMKC-LFA-MR does monotonically increase at each iteration among six datasets and it usually converges in less than 10 iterations in practical.

We also conduct the parameter sensitivity study on CCV dataset and report the clustering performance by ranging λ and β within the set of $\left[2^{-5}, 2^{-4}, \cdots, 2^{5}\right]$ shown in Figure 3. From the observation, increasing λ will improve the performance, and vice versa, that is, increasing β will improve the performance. Moreover, the clustering metric ACC increases when both λ and β increase.The curves with other datasets are similar and committed.

VII. CONCLUSION

This work has proposed a multiple kernel clustering framework with late fusion alignment to jointly utilize the various views of clustering partitions. The algorithm aligns the consensus partition with weighted base partitions. Moreover, a matrix-induced regularization is introduced to enhance the diversity and reduce the redundancy of selected clustering results. The proposed approach jointly optimizes the rotation matrices, weight coefficients and the optimal consensus partition. Our proposed algorithm, termed **CMKC-LFA-MR**, is proved to be an effectiveness and efficient algorithm comparable with state-of-the-art multiple kernel methods. In the future, we are supposed to analysis the influence of different initialization of basic partitions and more deep considerations about the connection between multi-view learning and ensemble learning needs to be explored.

REFERENCES

[1] J. Chen, Z. Zhao, J. Ye, and H. Liu, "Nonlinear adaptive distance metric learning for clustering," in *Proc. ACM 13th SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 123–132.
[2] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proc. ACM 25th Int. Conf. Mach. Learn.*, 2008, pp. 352–359.
[3] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. K. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for kernel K-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.
[4] 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.
[5] X. Liu, L. Wang, J. Yin, E. Zhu, and J. Zhang, "An efficient approach to integrating radius information into multiple kernel learning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 557–569, Apr. 2013.
[6] M. Gönen and A. A. Margolin, "Localized data fusion for kernel K-means clustering with application to cancer biology," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1305–1313.
[7] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel K-means using L21-norm," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3476–3482.
[8] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1888–1894.
[9] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1704–1710.
[10] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *Proc. AAAI 31st Conf. Artif. Intell.*, 2017, pp. 2266–2272.
[11] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.
[12] M. Ji, H. Rao, Z. Li, J. Zhu, and N. Wang, "Partial multi-view clustering based on sparse embedding framework," *IEEE Access*, vol. 7, pp. 29332–29343, 2019.
[13] P. Zhou, F. Ye, and L. Du, "Unsupervised robust multiple kernel learning via extracting local and global noises," *IEEE Access*, vol. 7, pp. 34451–34461, 2019.
[14] C. Wang, E. Zhu, X. Liu, L. Gao, J. Yin, and N. Hu, "Multiple kernel clustering with global and local structure alignment," *IEEE Access*, vol. 6, pp. 77911–77920, 2018.
[15] M. A. Rahman, K. L.-M. Ang, and K. P. Seng, "Unique neighborhood set parameter independent density-based clustering with outlier detection," *IEEE Access*, vol. 6, pp. 44707–44717, 2018.
[16] Z. Liao, L. Gao, T. Zhou, X. Fan, Y. Zhang, and J. Wu, "An oil painters recognition method based on cluster multiple kernel learning algorithm," *IEEE Access*, vol. 7, pp. 26842–26854, 2019.
[17] C. Liu, J. Liu, D. Peng, and C. Wu, "A general multiobjective clustering approach based on multiple distance measures," *IEEE Access*, vol. 6, pp. 41706–41719, 2018.
[18] X. Liu, L. Wang, G.-B. Huang, J. Zhang, and J. Yin, "Multiple kernel extreme learning machine," *Neurocomputing*, vol. 149, pp. 253–264, Feb. 2015.

[19] L. Sun, R. Liu, J. Xu, S. Zhang, and Y. Tian, "An affinity propagation clustering method using hybrid kernel function with LLE," *IEEE Access*, vol. 6, pp. 68892–68909, 2018.
[20] S. Rui, S. Liu, and P. Fan, "Amplifying inter-message distance: On information divergence measures in big data," *IEEE Access*, vol. 5, pp. 24105–24119, 2017.
[21] T. Lee, T. Moon, S. J. Kim, and S. Yoon, "Regularization and kernelization of the maximin correlation approach," *IEEE Access*, vol. 4, pp. 1385–1392, 2016.
[22] T. Li, L. Zhang, W. Lu, H. Hou, X. Liu, W. Pedrycz, and C. Zhong, "Interval kernel fuzzy C-means clustering of incomplete data," *Neurocomputing*, vol. 237, pp. 316–331, May 2017.
[23] D. Marin, M. Tang, I. B. Ayed, and Y. Boykov, "Kernel clustering: Density biases and solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 136–147, Jan. 2019.
[24] Q. Wang, Y. Dou, X. Liu, F. Xia, Q. Lv, and K. Yang, "Local kernel alignment based multi-view clustering using extreme learning machine," *Neurocomputing*, vol. 275, pp. 1099–1111, Jan. 2018.
[25] J. Liu, Z. Wu, Z. Xiao, and J. Yang, "Region-based relaxed multiple kernel collaborative representation for hyperspectral image classification," *IEEE Access*, vol. 5, pp. 20921–20933, 2017.
[26] Z.-Y. Li, R. Fang, and Y. Wang, "Stability and robust stability of integral delay systems with multiple exponential kernels," *IEEE Access*, vol. 5, pp. 21650–21659, 2017.
[27] A. Koppel, K. Zhang, H. Zhu, and T. Başar, "Projected stochastic primal-dual method for constrained online learning with kernels," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2528–2542, May 2019.
[28] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI 28th Conf. Artif. Intell.*, 2014, pp. 2149–2155.
[29] P. Zhou, L. Du, L. Shi, H. Wang, and Y.-D. Shen, "Recovery of corrupted multiple kernels for clustering," in *Proc. 34th Int. Joint Conf. Artif. Intell.*, 2015, pp. 4105–4111.
[30] L. Du, Y. Pan, and X. Luo, "Robust spectral clustering via matrix aggregation," *IEEE Access*, vol. 6, pp. 53661–53670, 2018.
[31] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 638–649.
[32] S. Wang, E. Zhu, J. Hu, M. Li, K. Zhao, N. Hu, and X. Liu, "Efficient multiple kernel K-means clustering with late fusion," *IEEE Access*, vol. 7, pp. 61109–61120, 2019.
[33] X. Liu, X. Zhu, M. Li, C. Tang, E. Zhu, J. Yin, and W. Gao, "Efficient and effective incomplete multi-view clustering," in *Proc. AAAI*, 2019, pp. 1–8.
[34] Z.-Z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann, "Double fusion for multimedia event detection," in *Proc. Int. Conf. Multimedia Modeling*. Springer-Verlag, 2012, pp. 173–185.
[35] L. Houthuys, R. Langone, and J. A. K. Suykens, "Multi-view kernel spectral clustering," *Inf. Fusion*, vol. 44, pp. 46–56, Nov. 2018.
[36] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *Proc. IJCAI*, 2017, pp. 2843–2849.
[37] Y. Lu, L. Wang, J. Lu, J. Yang, and C. Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognit.*, vol. 47, no. 11, pp. 3656–3664, 2014.
[38] T. Wang, J. Lu, and G. Zhang, "Two-stage fuzzy multiple kernel learning based on Hilbert–Schmidt independence criterion," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3703–3714, Dec. 2018.
[39] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 795–828, Jan. 2012.
[40] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.

JINGTAO HU is currently a graduate student with the National University of Defense Technology (NUDT), China. Her current research interests include kernel learning, unsupervised anomaly detection, and neutral networks.

MIAOMIAO LI is currently pursuing the Ph.D. degree with the National University of Defense Technology, China. She is currently a Lecturer with the Changsha College, Changsha, China. She has published several peer-reviewed articles such as AAAI, IJCAI, and *Neurocomputing*. Her current research interests include kernel learning and multi-view clustering. She served on the Technical Program Committees of IJCAI.

EN ZHU received the Ph.D. degree from the National University of Defense Technology (NUDT), China, where he is currently a Professor with the School of Computer Science. He has published more than 60 peer-reviewed articles, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, and IJCAI. His current research interests include pattern recognition, image processing, machine vision, and machine learning. He received the China National Excellence Doctoral Dissertation.

SIWEI WANG is currently a graduate student with the National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning, scalable kernel k-means, and deep neutral networks.

XINWANG LIU (M'13) received the Ph.D. degree from the National University of Defense Technology (NUDT), China, where he is currently an Assistant Researcher with the School of Computer Science. He has published more than 40 peer-reviewed articles, including those in highly regarded journals and conferences, such as IEEE T-IP, IEEE T-NNLS, ICCV, AAAI, and IJCAI. His current research interests include kernel learning and unsupervised feature learning. He served on the Technical Program Committees of IJCAI 2016/2017/2018 and AAAI 2016/2017/2018.

YONGPING ZHAI received the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), China, where he is currently a Research Associate with the College of Advanced Interdisciplinary Studies. His current research interests include computer vision, deep learning, and medical robotics.