Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 6654304, 11 pages
https://doi.org/10.1155/2023/6654304

Research Article

Localized Simple Multiple Kernel K-Means Clustering with Matrix-Induced Regularization

Jiaji Qiu,[1] **Huiying Xu**,[1] **Xinzhong Zhu**,[1,2] **and Michael Adjeisah**[1]

[1]*School of Computer Science and Technology, Zhejiang Normal University, Jinhua 321004, China*
[2]*AI Research Institute of Beijing Geekplus Technology Co. Ltd., Beijing 100101, China*

Correspondence should be addressed to Huiying Xu; xhy@zjnu.edu.cn

Received 1 December 2022; Revised 9 February 2023; Accepted 4 March 2023; Published 17 March 2023

Academic Editor: Radu-Emil Precup

Copyright © 2023 Jiaji Qiu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multikernel clustering achieves clustering of linearly inseparable data by applying a kernel method to samples in multiple views. A localized SimpleMKKM (LI-SimpleMKKM) algorithm has recently been proposed to perform min-max optimization in multikernel clustering where each instance is only required to be aligned with a certain proportion of the relatively close samples. The method has improved the reliability of clustering by focusing on the more closely paired samples and dropping the more distant ones. Although LI-SimpleMKKM achieves remarkable success in a wide range of applications, the method keeps the sum of the kernel weights unchanged. Thus, it restricts kernel weights and does not consider the correlation between the kernel matrices, especially between paired instances. To overcome such limitations, we propose adding a matrix-induced regularization to localized SimpleMKKM (LI-SimpleMKKM-MR). Our approach addresses the kernel weight restrictions with the regularization term and enhances the complementarity between base kernels. Thus, it does not limit kernel weights and fully considers the correlation between paired instances. Extensive experiments on several publicly available multikernel datasets show that our method performs better than its counterparts.

1. Introduction

Clustering is a widely used machine learning algorithm [1–4]. Multikernel clustering is one of the clustering methods which is based on multiview clustering and performs clustering by implicitly mapping sample points of different views to high dimensions. Many studies have been carried out in recent years [5–9]. For example, early work [10] shows that kernel matrices could encode different views or sources of the data, and MKKM [11] extends the kernel combination by adapting the weights of kernel matrices. Gönen and Margolin [12] improve the performance of MKKM by focusing on sample-specific weights on the correlations between neighbors to obtain a better clustering, called localized MKKM. Du et al. [13] engaged the $\ell_{2,1}$ norm to reduce the uncertainty of algorithm results due to unexpected factors such as outliers. To enhance the complementary nature of base kernels and reduce redundancy, Liu et al. [14] employed a regularization term containing a matrix that measures the correlation between base kernels to facilitate alignment. Other works [15–19]are different from the original MKKM method [11] that prefused multiple view kernels. These methods first obtain the clustering results of each kernel matrix, then fuse each clustering result in a later stage to obtain a unified result.

More recently, a newly proposed optimization strategy, simple multiple kernel k-means (SimpleMKKM) [20] has emerged as a representative of multikernel clustering (MKC). Different from the normal MKKM algorithm, SimpleMKKM assumes minimization of kernel weights and maximization of cluster partition, which leads to min-max optimization that is somewhat difficult to unravel. It converts the optimization to a minimization problem and cleverly solves it with a specially designed gradient descent method rather than a coordinate descent method. However, it is established that the strict alignment of the combined kernel matrix can force the combination globally. Therefore, Liu et al. proposed [21] localized SimpleMKKM, which reduces the negative impact of distant samples on clustering by restricting the kernel alignment to the k-nearest neighbors of the sample rather than the global alignment. In this way, LI-SimpleMKKM can sufficiently account for the variation between samples, improving clustering performance.

Although localized SimpleMKKM shows excellent performance on MKC problems, we find that the correlation between the given kernels is not sufficiently considered providing an opportunity for improvement based on the listed problem statement.

(i) The original method [21] makes the data stable by setting a larger weight η_u in the gradient descent step and maintaining the summation and nonnegativity of the weights through the association with other weights. However, this idea only enhances the correlation between different view weights and η_u and does not consider the relationship between view kernel matrices, especially between pairs.

(ii) The original method is possible to select multikernel kernels with high correlation for clustering simultaneously. Repeated selection of similar information sources makes the algorithm redundant and has low diversity, leading to the low ratio of different kernel matrices' effectiveness, ultimately affecting the accuracy of the clustering results.

Motivated by these, we propose a localized SimpleMKKM with matrix-induced regularization (LI-SimpleMKKM-MR) to improve upon the LI-SMKKM algorithm by adding an entry containing a matrix to measure the correlation between each two basis kernel matrices. LI-SimpleMKKM-MR algorithm can reduce the probability and simultaneously select high-correlation kernels, thereby enhancing the diversity of synthetic kernels and enhancing the complementarity of low-correlation kernels. Moreover, it adopts the advantage of localized SimpleMKKM, which has a better optimization effect that can be achieved by clustering the neighbor index matrix formed by the sample and the nearest k neighbors, and uses the optimization strategy $\min_\eta - \max_H$ instead of $\min_\eta - \min_H$.

Compared with the original multiple kernel clustering, the proposed method optimizes kernel matrix weights by using gradient descent rather than coordinate descent, combined with localized sample alignment and kernel matrix induced regularization. This reduces the negative effects of forced alignment of long-distance samples and high redundancy and low complementarity of multiple kernel matrices.

We experimented with the algorithm on 6 benchmark datasets and compared it with the other nine baseline algorithms that solve similar problems through four indicators: clustering accuracy (ACC), normalized mutual information (NMI), purity, and rand index. We find that LI-SimpleMKKM-MR outperforms other methods. This is the first work to fully consider and solve the correlation problem between the base kernels to the best of our knowledge.

The contributions of this method are summarized as follows:

(1) Proposed algorithm LI-SimpleMKKM-MR can productively deal with the alignment problem between kernel matrices using a regularization term, in order to reduce the redundancy, enhance the complementarity, and correlation between kernel matrices.

(2) The novelty is that our proposed method can be transformed into SimpleMKKM or LI-SimpleMKKM by adjusting the hyperparameters, making LI-SimpleMKKM-MR an extension of the previous two methods.

(3) We conducted extensive experiments on 6 public multiple kernel datasets using 4 metrics. The results show that our method achieves state-of-the-art performance compared to 9 existing baseline algorithms. The experiments essentially validate our understanding of the previous problems and the effectiveness of the proposed solution.

2. Related Works

2.1. Multiple Kernel K-Means. Let $\{\mathbf{x}_i\}_{i=1}^n \in \chi$ be a set of n samples, and $\phi_p(\cdot)$: $\mathbf{x} \in \chi \Rightarrow \mathscr{H}_p$ means mapping the features of the sample $\mathbf{x}$ of the pth view into a high-dimensional Hilbert space $\mathscr{H}_p\,(1 \leq p \leq m)$. According to this theory, each sample can be represented by $\phi_\eta(\mathrm{x}) = [\eta_1\phi_1^\top(\mathrm{x}), \ldots, \eta_m\phi_m^\top(\mathrm{x})]$, where $\eta = [\eta_1, \cdots \eta_m]^\top$ means the weights of m prespecified base kernels $\{\mathbf{K}_p(\cdot,\cdot)\}_{p=1}^m$. The kernel weights will be changed according to the algorithm optimizing in the kernel learning step. According to the definition of $\phi_\eta(\mathrm{x})$ and the definition of kernel function, the kernel function can be defined as follows:

$$\mathbf{K}_{\boldsymbol{\eta}}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \phi_\eta{}^\top(\mathbf{x}_i)\phi_\eta\left(\mathbf{x}_j\right) = \sum_{p=1}^{m} \eta_p^2 \mathbf{K}_p\left(\mathbf{x}_i, \mathbf{x}_j\right). \tag{1}$$

We can use training samples $\{\mathbf{x}_i\}_{i=1}^n$ by (1) to calculate a kernel matrix $\mathbf{K}_\eta$. Based on the calculation of $\mathbf{K}_\eta$, the objective function of MKKM with $\mathbf{K}_\eta$ can be expressed as follows:

$$\begin{aligned} &\min_{\mathbf{H},\boldsymbol{\eta}} \quad \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\eta}}\left(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top\right)\right) \text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ &\boldsymbol{\eta}^\top \mathbf{1}_m = 1, \eta_p \geq 0, \forall p. \end{aligned} \tag{2}$$

Here, $\mathbf{H} \in \mathbb{R}^{n\times k}$ means one soft label matrix, which is used to solve NP-hard problems caused by the direct use of hard allocation, which is also called the partition matrix. Moreover, $\mathbf{I}_k$ means an identity matrix which is $k \times k$ in size.

Optimization of (2) can be divided into 2 steps: optimizing $\mathbf{H}$ or η and fixing the other one.

(i) Optimizing $\mathbf{H}$ with η is fixed, the problem of optimizing $\mathbf{H}$ in (2) can be represented as follows:

$$\begin{aligned} &\max_{\mathbf{H}} \quad \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{K}_{\boldsymbol{\eta}}\mathbf{H}\right) \text{s.t. } \mathbf{H} \in \mathbb{R}^{n\times k}, \\ &\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k. \end{aligned} \tag{3}$$

The optimization of $\mathbf{H}$ of (3) can be easily solved by taking the first k eigenvalues of the matrix $\mathbf{K}_\eta$.

(ii) Optimizing η with $\mathbf{H}$ is fixed, with the soft label matrix $\mathbf{H}$ is fixed, the problem of optimizing η in (2) can be represented as follows:

$$\min_{\boldsymbol{\eta}} \quad \sum_{p=1}^{m} \eta_p^2 \mathrm{Tr}\left(\mathbf{K}_p \left(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top\right)\right) \text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \quad (4)$$
$$\boldsymbol{\eta}^\top \mathbf{1}_m = 1, \eta_p \geq 0, \forall p.$$

According to the constraints, it can be easily solved by the Lagrange multiplier method [10].

2.2. MKKM with Matrix-Induced Regularization. As (2) shows that η_p only depends on $\mathbf{K}_p$ and $\mathbf{H}$. However, the interactions between different kernel matrices are not considered. Liu et al. [14] defined a criterion $\mathcal{M}(\mathbf{K}_p, \mathbf{K}_q)$ to measure the correlation between $\mathbf{K}_p$ and $\mathbf{K}_q$. A larger $\mathcal{M}(\mathbf{K}_p, \mathbf{K}_q)$ means high correlation between $\mathbf{K}_p$ and $\mathbf{K}_q$, and a smaller one implies that their correlation is low. By introducing the criterion term in (2), we can obtain the following objective function:

$$\underset{\mathbf{H},\boldsymbol{\eta}}{\mathrm{Min}} \quad \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\eta}} \left(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top\right)\right) + \frac{\lambda}{2}\boldsymbol{\eta}^\top \mathbf{M}\boldsymbol{\eta} \text{ s.t. } \mathbf{H} \in \mathbb{R}^{n\times k},$$
$$\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \quad (5)$$
$$\boldsymbol{\eta}^\top \mathbf{1}_m = 1, \eta_p \geq 0, \forall p,$$

where λ is a hyperparameter to balance clustering loss and regularization term.

2.3. Localized SimpleMKKM. Unlike the existing $\min_\eta - \min_H$ paradigm, SimpleMKKM adopts $\min_\eta - \max_H$ optimization [20]. However, it is extended to make full use of the information between local sample neighbors and $\min_\eta - \max_H$ optimization to enhance the clustering effect with a fusion algorithm called localized SimpleMKKM. The objective value of LI-SimpleMKKM can be represented as follows:

$$\min_{\boldsymbol{\eta}} \max_{\mathbf{H}} \mathrm{Tr}\left(\mathbf{H}^\top \sum_{i=1}^{n}\left(\mathbf{B}^{(i)}\mathbf{K}_{\boldsymbol{\eta}}\mathbf{B}^{(i)}\right)\mathbf{H}\right) \text{s.t. } \mathbf{H} \in \mathbb{R}^{n\times k},$$
$$\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \quad (6)$$
$$\sum_{p=1}^{m} \eta_p = 1, \eta_p \geq 0, \forall p,$$

where $\sum_{p=1}^{m}\eta_p^2\mathbf{K}_p$ and $\mathbf{B}^{(i)} = \mathbf{N}^{(i)}\mathbf{N}^{(i)\top}$ with $\mathbf{N}^{(i)} \in 0, 1^{n\times \mathrm{round}(\tau\times n)}$ are the ith sample's neighborhood mask matrices; that is, only the samples closest to the target sample will be aligned. This new method is hard to solve with a simple two-step alternating optimization convergence method. To solve this problem, LI-SimpleMKKM first optimizes $\mathbf{H}$ by a method similar to MKKM and then converts the problem into a problem of finding the minimum with respect to η. With proving the differentiability of the minimized formula, the gradient descent method can be used to optimize η [21].

3. Localized Simple Multiple Kernel K-Means with Matrix-Induced Regularization

According to Liu et al. [21], the relative value of η_p is only dependent on $\mathbf{K}_p$, $\mathbf{H}$, and $\mathbf{K}_u$, where u is the largest component of η. Only the weights of different kernels are linked, indicating that the LI-SimpleMKKM algorithm is not fully considered the interaction of the kernels when optimizing the kernel weights. This motivates us to derive a regularization term which can measure the correlation between the base kernels to improve this shortcoming.

3.1. Formulation. Although the performance of clustering can be improved to some extent by aligning samples with closer samples, there is still room for further improvement of that algorithm.

To address this issue, we define a criterion $\mathcal{M}(\mathbf{K}_p, \mathbf{K}_q)$ to measure the correlation between $\mathbf{K}_p$ and $\mathbf{K}_q$. A larger $\mathcal{M}(\mathbf{K}_p, \mathbf{K}_q)$ means high correlation between $\mathbf{K}_p$ and $\mathbf{K}_q$, and a smaller one implies that their correlation is low. We propose to add a matrix-induced regularization $\eta^T\mathbf{M}\eta$ based on LI-SimpleMKKM to improve the shortcomings, enhancing the kernel alignment between multiple kernels and reducing the redundancy of kernels with higher correlation. By fusing the regular term with (6), we can get the objective function as follows:

$$\min_{\boldsymbol{\eta}\in\Delta} \quad \max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \mathrm{Tr}\left(\mathbf{H}^\top \sum_{i=1}^{n}\left(\mathbf{B}^{(i)}\mathbf{K}_{\boldsymbol{\eta}}\mathbf{B}^{(i)}\right)\mathbf{H}\right) + \frac{\lambda}{2}\boldsymbol{\eta}^\top\mathbf{M}\boldsymbol{\eta},$$
$$\text{s.t. } \mathbf{H} \in \mathbb{R}^{n\times k}, \quad (7)$$
$$\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k,$$
$$\sum_{p=1}^{m} \eta_p = 1, \eta_p \geq 0, \forall p,$$

where λ is a trade-off parameter to balance the loss of clustering problem and the regularization term on kernel weights. The regularization term has many types, such as KL divergence and Hilbert–Schmidt independent criterion.

In our proposed algorithm, we set $\mathbf{M}_{pq} = \mathrm{Tr}(\mathbf{K}_p\mathbf{K}_q)$ for each element in $\mathbf{M}$ to measure the correlation between $\mathbf{K}_p$ and $\mathbf{K}_q$. Choosing this method makes the calculation not too complicated and adopts the Hilbert–Schmidt independent criterion in disguise, which can reflect the correlation between different base kernels to a certain extent.

The incorporation of $\eta^T\mathbf{M}\eta$ use of the basic kernel better, thus improving clustering performance. Moreover, we can clearly see that if we set $\lambda = 0$, equation (7) is a special case of LI-SimpleMKKM.

Li et al. [22] use $\eta^\top\mathbf{M}^{(i)}\eta$ instead of $\eta^\top\mathbf{M}\eta$ as a regular term, where $\mathbf{M}^{(i)}$ means a matrix with $\mathbf{M}_{pq}^{(i)} = Tr(\mathbf{K}_p^{(i)}\mathbf{K}_q^{(i)})$, $\mathbf{K}_\eta^{(i)} = \mathbf{N}^{(i)\top}\mathbf{K}_\eta\mathbf{N}^{(i)}$. Although this method shows excellent performance, we find that the induced regularization of matrices should be global rather than local because the kernel alignment should be for the global kernel matrix. It can also be found from the experimental results in Table 1 that the global kernel-induced regularization has a better effect.

Table 1: Comparation of localized **M** and global **M** in ACC.

Datasets	Localized **M**	Global **M**
Flower17	51.8 ± 1.0	61.3 ± 1.5
Flower102	37.4 ± 0.8	44.0 ± 1.0
ProteinFold	36.4 ± 1.5	39.0 ± 1.4
Digital	83.4 ± 1.2	94.9 ± 3.0
Cal-25views	36.6 ± 0.7	37.3 ± 1.1
Cal-7classes	47.8 ± 1.0	49.8 ± 0.8

3.2. Alternate Optimization. We design a two-step alternating optimization to solve the formula in (7).

(i) Optimizing $\mathbf{H}$ by η is fixed: fixed η, the optimization value with respect to $\mathbf{H}$ in (7) is represented as follows:

$$\max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \quad \mathrm{Tr}\left(\mathbf{H}^\top \sum_{i=1}^{n}\left(\mathbf{B}^{(i)}\mathbf{K}_{\boldsymbol{\eta}}\mathbf{B}^{(i)}\right)\mathbf{H}\right) \text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k. \tag{8}$$

Treating the summation $(\mathbf{B}^{(i)}\mathbf{K}_\eta\mathbf{B}^{(i)})$ as a whole, (8) can be solved by solving for the eigenvalues of the matrix.

(ii) Optimizing η by $\mathbf{H}$ is fixed: fixed $\mathbf{H}$, the optimization value with respect to η in (7) can be represented as follows:

$$\mathcal{J}(\boldsymbol{\eta}) = \max_{\boldsymbol{\eta}} \mathrm{Tr}\left(\mathbf{H}^\top \sum_{i=1}^{n}\left(\mathbf{B}^{(i)}\mathbf{K}_{\boldsymbol{\eta}}\mathbf{B}^{(i)}\right)\mathbf{H}\right). \tag{9}$$

$$+ \frac{\lambda}{2}\boldsymbol{\eta}^T\mathbf{M}\boldsymbol{\eta} \text{ s.t. } \mathbf{H}^\top\mathbf{H} = .$$

We first prove the differentiability of (9), then calculate the gradient, and optimize η by the gradient descent method. The first part of the objective function in (9) is as follows:

$$\mathrm{Tr}\left(\mathbf{H}^\top \sum_{i=1}^{n}\left(\mathbf{B}^{(i)}\mathbf{K}_{\boldsymbol{\eta}}\mathbf{B}^{(i)}\right)\mathbf{H}\right). \tag{10}$$

With the hyperparameter τ defined, we can regard $\mathbf{B}^{(i)}\mathbf{K}_\eta\mathbf{B}^{(i)}$ as a whole, which is global kernel alignment and PSD [21]. For convenience, we let $\mathbf{B}^{(i)}\mathbf{K}_\eta\mathbf{B}^{(i)} = \mathbf{T}_\eta^{(i)}$.

Thus, the function in (9) can be represented as follows:

$$\min_{\boldsymbol{\eta}\in\Delta} \mathcal{J}(\boldsymbol{\eta}), \tag{11}$$

with

$$\mathcal{J}(\boldsymbol{\eta}) = \max_{\mathbf{H}} \quad \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{T}_{\boldsymbol{\eta}}^{(i)}\mathbf{H}\right) + \frac{\lambda}{2}\boldsymbol{\eta}^\top\mathbf{M}\boldsymbol{\eta} \text{ s.t. } \mathbf{H}\in\mathbb{R}^{n\times k},$$
$$\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k. \tag{12}$$

Theorem 1. *$\mathcal{J}(\eta)$ in (12) is differentiable.* $(\partial\mathcal{J}(\eta)/\partial\eta_p) = 2\eta_p Tr(\mathbf{H}^{*\top}\mathbf{T}_p^{(i)}\mathbf{H}^*) + \lambda\sum_{q=1}^{m}\eta_q M_{pq}$, *where* $\mathbf{H}^* = \left\{\underset{\mathbf{H}}{\mathrm{argmax}}\ Tr(\mathbf{H}^\top\mathbf{T}_\eta^{(i)}\mathbf{H}) s.t. \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\right\}$.

Proof. For any given $\eta \in \Delta$, the maximum of optimization problem $Tr(\mathbf{H}^\top\mathbf{K}_\eta\mathbf{H}) s.t. \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k$ is unique [21], with $\widetilde{\mathbf{H}^*} \in \left\{\widetilde{\mathbf{H}^*} \mid \widetilde{\mathbf{H}^*} = \mathbf{H}^*\mathbf{U}, \mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I}_k\right\}$ the corresponding maximizer. According to theorem in [23], the former part of $\mathcal{J}(\eta)$ is differentiable. By defining other elements in η except for p as s and the latter part of the $\mathcal{J}(\eta)$ as $\mathcal{J}_2(\eta)$, the differential of $\mathcal{J}_2(\eta) = (\lambda/2)\eta^\top\mathbf{M}\eta$ can be expressed as follows:

$$\begin{aligned}\mathcal{J}_2(\boldsymbol{\eta}) &= \frac{\lambda}{2}\left[\eta_p \ \ \eta_s\right]\begin{bmatrix} M_{pp} & M_{ps} \\ M_{ps} & M_{ss}\end{bmatrix}\begin{bmatrix}\eta_p \\ \eta_s\end{bmatrix} \\ &= \frac{\lambda}{2}\left(\eta_p^2 M_{pp} + 2\eta_p\eta_s M_{ps} + \eta_s^2 M_{ss}\right), \\ \frac{\partial\mathcal{J}_2(\boldsymbol{\eta})}{\partial\eta_p} &= \lambda\eta_p M_{pp} + \lambda\eta_s M_{ps},\end{aligned} \tag{13}$$

where p means one of the components of η and s means all of the other components so that $(\partial\mathcal{J}_2(\eta)/\partial\eta_p) = \lambda\sum_{q=1}^{m}\eta_q M_{pq}$, and the whole $\mathcal{J}(\eta)$ in (12) is differentiable.

We can solve this problem by designing a gradient descent method. After obtaining the gradient of $\mathcal{J}(\eta)$ under the premise of satisfying the equality constraints $\sum_{p=1}^{m}\eta_p = 1$ and nonnegativity constraints $\eta_p \geq 0$ of η, we update η by gradient descent [23]. To implement this method, we let η_u become a nonzero unit in η and $\nabla\mathcal{J}(\eta)$ indicates the reduced gradient of $\mathcal{J}(\eta)$. The pth $(1 \leq p \leq m)$ element of $\nabla\mathcal{J}(\eta)$ is presented as follows:

$$[\nabla\mathcal{J}(\boldsymbol{\eta})]_p = \frac{\partial\mathcal{J}(\boldsymbol{\eta})}{\partial\eta_p} - \frac{\partial\mathcal{J}(\boldsymbol{\eta})}{\partial\eta_u}\forall p \neq u, \tag{14}$$

and

$$[\nabla\mathcal{J}(\boldsymbol{\eta})]_u = -\sum_{p=1, p\neq u}^{m}[\nabla\mathcal{J}(\boldsymbol{\eta})]_p. \tag{15}$$

To improve numerical stability, we choose u as the largest unit in the vector η. The nonnegativity constraint of η also needs to be considered during gradient descent.

To minimize $\mathcal{J}(\eta)$, we define $-\nabla\mathcal{J}(\eta)$ as a descent direction. However, if there is an index p corresponding to $\eta_p = 0$, with $[\nabla\mathcal{J}(\eta)]_p \geq 0$, the situation of $\eta_p < 0$ may occur when the gradient is updated, violating the nonnegativity constraint. Under these circumstances, the descent direction for that unit p is set as zero. This makes η when the gradient is updated as follows:

$$d_p = \begin{cases} 0, & \text{if } \eta_p = 0 \text{ and } [\nabla \mathcal{J}(\boldsymbol{\eta})]_p \geq 0, \\ -[\nabla \mathcal{J}(\boldsymbol{\eta})]_p, & \text{if } \eta_p \geq 0 \text{ and } p \neq u, \\ -[\nabla \mathcal{J}(\boldsymbol{\eta})]_u, & \text{for } p = u. \end{cases} \quad (16)$$

The gradient update adopts the formula $\eta \longleftarrow \eta + \gamma \mathbf{d}$, where γ is the step size. We determine the step size γ by a one-dimensional linear search method, rather than setting it directly, and in order to ensure global convergence, this method has appropriate stopping criteria, for example, Armijo's rule [21]. □

The specific calculation steps of the algorithm in equation (13) are detailed in Algorithm 1.

Theorem 2. *The proposed algorithm is converged.*

Proof. Note that for the kth iteration, $\mathrm{Tr}(\mathbf{H}^{\top}\mathbf{T}_{\eta}^{(i)}\mathbf{H})$ will be bigger than $k+1$th iteration. In each iteration, the gradient of η_p is smaller than 0 by equation (14) because u is the component of η, and in order to get the maximum of $\mathrm{Tr}(\mathbf{H}^{\top}\mathbf{T}_{\eta}^{(i)}\mathbf{H})$, $\mathrm{Tr}(\eta_u \mathbf{H}^{\top}\mathbf{T}_{u}^{(i)}\mathbf{H})$ should be larger than other parts, so the differential of it is bigger than others. The component u has the gradient which is the opposite number of other component gradients' sum by the equation (15). According to the equation (16), the component of p will be bigger, while the coefficient of u will be smaller, and we can let Δ as the difference of the kth iteration and $k+1$th's, with $\Delta = \mathrm{Tr}\gamma(d_1\mathbf{H}^{\top} \quad \mathbf{T}_1^{(i)}\mathbf{H}) + \mathrm{Tr}\gamma(d_2\mathbf{H}^{\top}\mathbf{T}_2^{(i)}\mathbf{H}) + \ldots + \mathrm{Tr}\gamma(d_m \mathbf{H}^{\top}\mathbf{T}_m^{(i)}\mathbf{H})$, $\sum_{p=1,p\neq u}^{m} d_p$ with $\mathbf{H}^{\top}\mathbf{T}_u^{(i)}\mathbf{H}$ as the largest part of each $\mathbf{H}^{\top}\mathbf{T}^{(i)}\mathbf{H}$, γ is bigger than 0, it can be easy to get the conclusion Δ is smaller than 0, because the non-negativity of η **a**nd kernel matrix, the former term has the lower bound 0 and convex, so the former term's convergence is been proofed. □

As for the latter term $(\lambda/2)\eta^{\top}\mathbf{M}\eta$ with the similar thought, it is also decreasing monotonically because $\mathbf{M}$ is a PSD matrix, η is not negative, and λ is bigger than 0; the second derivative of $\mathcal{J}_2(\eta)$ can be easy to be calculated bigger than 0 (since each element of $\mathbf{M}$ is bigger than 0), so the latter term has the lower bound 0 and convex. At the same time, the whole equation (13) is monotonically decreasing and lower-bounded.

3.3. Computational Complexity Analysis. We theoretically analyze the time complexity of the algorithm LI-SimpleMKKM-MR. We assume that n and m denote the number of samples and the number of base kernels. LI-SimpleMKKM-MR based on Algorithm 1 first computes a neighborhood mask matrix with computational complexity $\mathcal{O}(n^2 \log_2 n)$ and then computes the regularization term with computational complexity $\mathcal{O}(m^3)$. Therefore, the time complexity of LI-SimpleMKKM-MR is $(n^3 + n^2 \log_2 n + m^3)$ per iteration.

Let us compare the complexity of LI-SimpleMKKM-MR and LI-SimpleMKKM. Since in most cases, the number of base kernels is much fewer than the number of samples $(m \ll n)$, compared with LI-SimpleMKKM (n^3), the time complexity of the proposed method does not increase significantly.

4. Experiments

4.1. Datasets. In this section, we evaluate the clustering performance of our algorithm on a set of standard MKKM benchmark datasets, including Oxford Flower17(FLO17), Flower102(FLO102) (https://www.robots.ox.ac.uk/~+vgg/data/flowers/), Protein Fold Prediction(proteinFold) (https://mkl.ucsd.edu/dataset/protein-fold-prediction/), Digital (https://ss.sysu.edu.cn/~+py/), Caltech101-25views(Cal-25views), and Caltech101-7classes(Cal-7classes) (https://files.is.tue.mpg.de/pgehler/projects/iccv09/). Caltech101-25views refers to the number of kernels randomly selected by 25, and Caltech101-7classes refers to the number of classes randomly selected by 7. The details of these can be found in Table 2. We can compare the performance of the different MKKM algorithms using these datasets.

4.2. Compared Algorithms. In addition to the localized SimpleMKKM with matrix-induced regularization, we tested nine other comparative algorithms from the other MKKM algorithms, including, average kernel k-means (**Avg-KKM**), multiple kernel k-means (**MKKM**) [10], localized multiple kernel k-mean (**LMKKM**) [12], optimal neighborhood kernel clustering (**ONKC**) [24], multiple kernel k-mean with matrix-induced regularization (**MKKM-MR**) [14], multiple kernel clustering with local alignment maximization (**LKAM**) [22], multiview clustering via late fusion alignment maximization (**LF-MVC**) [25], simple multiple kernel k-means (**SimpleMKKM**) [20], and localized SimpleMKKM (**LI-SimpleMKKM**) [21].

The implementations of the comparison algorithms are publicly available in the corresponding papers, and we directly apply them to our experiments without tuning. Among the previous algorithms, ONKC, MKKM-MR, LKAM, LF-MVC, and LI-SimpleMKKM need to adjust hyperparameters. Based on the published papers and actual experimental results, we show the best clustering results of the previous methods by tuning the hyperparameters on each dataset.

4.3. Experimental Settings. In all experiments, to reduce the difference between different views, all the base kernels are first centered and then scaled so that for all i and p, we have $\mathbf{K}_p(\mathbf{x}_i, \mathbf{x}_i) = 1$. For our proposed algorithm, its trade-off parameters λ and τ are chosen from $[2^{-15}, 2^{-13}, \ldots, 2^{10}]$ and $[0.05, 0.1, \ldots, 0.95] \times n$ by grid search, where n is the number of samples.

For all the datasets, we set the number of clusters k according to the actual number of categories in the dataset.

```
Input: $\left\{\mathbf{K}_p\right\}_{p=1}^{m}$, $k$, $\lambda$, $\tau$, $\epsilon$, and $t$
Initialization $\eta^{(0)} = 1/m$ and $t = 1$.
Calculate $\left\{\mathbf{B}^{(i)}\right\}_{i=1}^{n}$ by $\mathbf{B}^{(i)} = \mathbf{N}^{(i)}\mathbf{N}^{(i)\top}$ for $p$th samples $(1 \leq p \leq m)$ according to the average kernel.
$\mathbf{T}_p^{(i)} = (\sum_{i=1}^{n}\mathbf{B}^{(i)})$
repeat
    Update $\mathbf{H}$ by solving the first $k$ eigenvalues with $\mathbf{T}_\eta^{(i)}$.
    Compute $\partial\mathcal{J}(\eta)/\partial\eta_p\,(1 \leq p \leq m)$ and the direction of descent of the gradient $\mathbf{d}^{(t)}$ in (16).
    Update $\eta^{t+1} \leftarrow \eta^t + \gamma\mathbf{d}^t$.
    $t \leftarrow t + 1$.
Until $\max|\eta^{t+1} - \eta^t| \leq \epsilon$
```

Algorithm 1: LI-SimpleMKKM-MR.

Table 2: Specification of our 6 benchmark datasets.

Datasets	#Samples	#Kernels	#Classes
Flower17	1360	7	17
Flower102	8189	4	102
ProteinFold	694	12	27
Digital	2000	3	10
Cal-25views	1530	25	102
Cal-7classes	1474	6	7

We engage 4 indicators: clustering accuracy (ACC), normalized mutual information (NMI), purity, and rand index to measure the effect of clustering. To reduce the harmful effects of randomness, we initialized and executed all algorithms fifty times (50×) to obtain the mean and variance of the experimental indicators.

4.4. Experimental Results. Table 3 reports the ACC, NMI, purity, and rand index of the previously mentioned algorithms on all 6 datasets. The following observations were made based on the results:

The proposed localized SimpleMKKM with matrix-induced regularization significantly outperforms localized SimpleMKKM. For example, it outperforms the LI-SimpleMKKM algorithm by 1.8%, 0.1%, 3.1%, 0.3%, 0.6%, and 3.4% in terms of ACC on Flower17, Flower102, ProteinFold, DIGIT, Caltech-25 views, and Caltech-7 classes datasets. These results validate the effectiveness of enhancing the correlation between matrices.

Also, our proposed LI-SimpleMKKM-MR significantly outperforms the MKKM-MR algorithms by 3.6%, 3.8%, 4.7%, 7.5%, 3.3%, and 6.3% in terms of ACC on Flower17, Flower102, ProteinFold, DIGIT, Caltech-25 views, and Caltech-7 classes datasets. This result proves that utilizing the data's local structure and $\min_\eta - \max_H$ optimization improves the clustering effect very well.

The proposed algorithm adopts the advanced formulation and uses matrix-induced regularization to improve the correlation between kernel matrices, reducing redundancy and increasing the diversity of selected kernel matrices, making it superior to its counterpart.

Together, these factors make LI-SimpleMKKM-MR significantly improved over other algorithms on the same dataset. In addition, due to time complexity and memory constraints, the effect of LMKKM on some datasets has not been shown.

4.5. Parameter Sensitivity of LI-SimpleMKKM-MR. We designed comparative experiments to study the influence of the setting of two hyperparameters, localized alignment, and matrix-induced regularization, on the clustering effect. According to equation (7), LI-SimpleMKKM-MR tunes the clustering performance by setting two hyperparameters λ and τ, referring to the regularization balance factor and the nearest neighbor ratio.

We experimentally show the difference in clustering performance in λ and τ in all benchmark datasets.

Figure 1 shows the ACC and NMI of our algorithm by varying one of τ or λ with the other one fixed. Based on these figures, we can conclude that (1) as the value of τ increases, the ACC and NMI of each dataset increase to their highest value and, correspondingly, decrease when τ decreases and (2) by keeping the τ unchanged, the ACC and NMI will exceed SimpleMKKM and be steady when λ is small.

Hence, we conclude that our proposed algorithm presents a new state-of-the-art performance for clustering compared to other algorithms that only preserve the global kernel, such as LI-MKKM. Thus, it focuses on preserving the local structure of the data as specific results are displayed in Table 1.

On top of $\min_\eta - \max_H$ optimization, the clustering performance improves when the parameters are appropriately set by combining matrix-induced regularization and local alignment.

4.6. Convergence of LI-SimpleMKKM-MR. In addition to theoretical verification, we experimentally verify the convergence of the algorithm. We present simulations of our

Table 3: ACC, NMI, purity, and Rand index data of localized-SimpleMKKM-matrix-induced regularization with nine comparison methods on six benchmark datasets.

Dataset	AVG-KKM	MKKM [10]	LMKKM [12]	ONKC [24]	MKKM-MR [14]	LKAM [22]	LF-MVC [25]	SimpleMKKM [20]	LI-SimpleMKKM [21]	Proposed
ACC (%)										
FLO17	51.3 ± 1.4	43.6 ± 1.7	42.7 ± 1.7	43.4 ± 2.0	57.7 ± 1.2	48.9 ± 0.9	56.7 ± 1.5	59.4 ± 1.5	59.1 ± 1.1	**61.3 ± 1.5**
FLO102	27.1 ± 0.8	22.4 ± 0.8	—	39.5 ± 0.7	40.2 ± 0.9	41.4 ± 0.8	38.4 ± 1.2	42.5 ± 0.8	44.0 ± 1.0	**44.1 ± 1.0**
PFold	29.1 ± 1.4	27.1 ± 1.0	22.4 ± 0.7	35.4 ± 1.5	34.3 ± 1.6	34.2 ± 1.6	33.3 ± 2.0	34.4 ± 1.9	35.9 ± 1.5	**39.0 ± 1.4**
DIGIT	88.8 ± 0.1	47.2 ± 0.6	47.2 ± 0.7	89.5 ± 0.1	87.4 ± 0.1	**95.0 ± 0.3**	89.2 ± 0.1	90.3 ± 0.1	94.6 ± 0.1	94.9 ± 3.0
Cal-25	34.2 ± 1.0	32.8 ± 0.1	22.6 ± 0.7	34.3 ± 1.2	34.0 ± 1.2	37.2 ± 1.1	34.6 ± 1.1	36.2 ± 1.2	36.7 ± 0.9	**37.3 ± 1.1**
Cal-7	33.7 ± 0.1	33.3 ± 0.2	34.4 ± 0.1	46.0 ± 4.0	43.5 ± 3.9	49.4 ± 1.0	42.3 ± 2.7	39.4 ± 1.5	46.4 ± 0.9	**49.8 ± 0.8**
NMI (%)										
FLO17	49.9 ± 0.9	44.3 ± 1.3	43.8 ± 1.1	42.9 ± 1.3	56.1 ± 0.7	48.1 ± 0.6	54.6 ± 1.0	57.6 ± 0.1	57.7 ± 0.5	**58.9 ± 1.0**
FLO102	46.0 ± 0.5	42.7 ± 0.2	—	56.1 ± 0.4	56.7 ± 0.5	56.9 ± 0.3	54.9 ± 0.4	58.6 ± 0.5	60.0 ± 0.4	**60.1 ± 0.4**
PFold	40.3 ± 1.2	38.1 ± 0.6	34.8 ± 0.6	44.1 ± 0.8	43.2 ± 1.1	43.7 ± 1.0	42.3 ± 1.2	44.2 ± 1.2	45.2 ± 1.3	**48.0 ± 0.9**
DIGIT	80.7 ± 0.2	48.7 ± 0.7	48.3 ± 0.2	81.7 ± 0.1	79.5 ± 0.1	89.4 ± 0.1	81.2 ± 0.2	83.3 ± 0.1	90.0 ± 0.1	**90.3 ± 1.6**
Cal-25	59.7 ± 0.5	58.6 ± 0.5	51.9 ± 0.3	59.6 ± 0.8	59.3 ± 0.5	**62.0 ± 0.6**	59.5 ± 0.7	60.7 ± 0.5	61.4 ± 0.4	61.5 ± 0.5
Cal-7	34.9 ± 0.3	30.0 ± 0.3	30.7 ± 0.1	40.0 ± 1.0	43.1 ± 0.8	43.3 ± 0.2	40.0 ± 0.3	40.5 ± 0.4	39.2 ± 1.3	**44.9 ± 0.1**
Purity (%)										
FLO17	52.3 ± 1.2	45.1 ± 1.4	44.6 ± 1.5	45.1 ± 1.8	59.2 ± 1.1	50.1 ± 0.6	57.5 ± 1.6	60.5 ± 1.3	59.7 ± 0.3	**62.7 ± 1.6**
FLO102	32.3 ± 0.6	27.8 ± 0.4	—	45.6 ± 0.7	46.3 ± 0.8	48.0 ± 0.6	44.6 ± 0.8	48.6 ± 0.7	50.3 ± 0.7	**50.5 ± 0.7**
PFold	37.3 ± 1.6	33.7 ± 0.9	31.1 ± 1.0	42.0 ± 1.2	41.2 ± 1.4	41.6 ± 1.3	40.6 ± 1.6	41.4 ± 1.6	42.5 ± 1.6	**46.2 ± 1.5**
DIGIT	88.8 ± 0.1	50.1 ± 0.7	50.2 ± 0.3	89.5 ± 0.1	87.4 ± 0.1	**95.0 ± 0.3**	89.2 ± 0.1	90.3 ± 0.1	94.6 ± 0.1	94.9 ± 2.0
Cal-25	36.2 ± 1.0	34.9 ± 0.1	24.4 ± 0.6	36.6 ± 1.1	36.1 ± 1.0	**39.4 ± 1.1**	36.8 ± 1.0	38.2 ± 1.1	39.1 ± 0.8	39.3 ± 1.1
Cal-7	79.0 ± 0.2	76.7 ± 0.2	74.9 ± 0.1	81.2 ± 0.9	82.9 ± 0.4	83.2 ± 0.2	81.6 ± 0.3	83.3 ± 0.3	80.8 ± 0.7	**83.4 ± 0.1**
Rand Index (%)										
FLO17	32.2 ± 1.3	26.3 ± 1.3	20.6 ± 1.1	35.2 ± 1.5	39.9 ± 1.3	31.6 ± 0.8	44.1 ± 0.4	41.5 ± 1.1	40.9 ± 0.8	**42.6 ± 1.4**
FLO102	15.5 ± 0.5	12.1 ± 0.5	—	24.9 ± 0.5	25.5 ± 0.6	27.2 ± 0.6	25.5 ± 1.0	28.5 ± 0.8	29.9 ± 0.8	**30.0 ± 0.8**
PFold	14.4 ± 1.8	12.1 ± 0.7	7.8 ± 0.4	18.0 ± 1.1	17.2 ± 1.5	20.1 ± 1.1	16.5 ± 2.0	17.6 ± 1.9	19.8 ± 1.2	**20.8 ± 1.5**
DIGIT	77.4 ± 0.2	31.4 ± 0.6	31.3 ± 0.2	81.7 ± 0.3	81.3 ± 0.1	**90.8 ± 2.3**	78.0 ± 0.1	80.6 ± 0.2	88.2 ± 0.1	88.3 ± 2.9
Cal-25	18.5 ± 0.9	17.3 ± 0.1	8.3 ± 0.6	18.6 ± 1.2	18.4 ± 0.8	**21.5 ± 1.0**	18.9 ± 0.9	20.4 ± 0.1	20.1 ± 0.7	21.1 ± 0.8
Cal-7	26.5 ± 0.3	23.5 ± 0.3	20.3 ± 0.1	30.9 ± 1.7	31.5 ± 2.2	36.6 ± 0.5	29.3 ± 0.8	30.3 ± 0.7	32.2 ± 1.3	**37.1 ± 0.3**

Bold indicates better results in comparison with other algorithms.

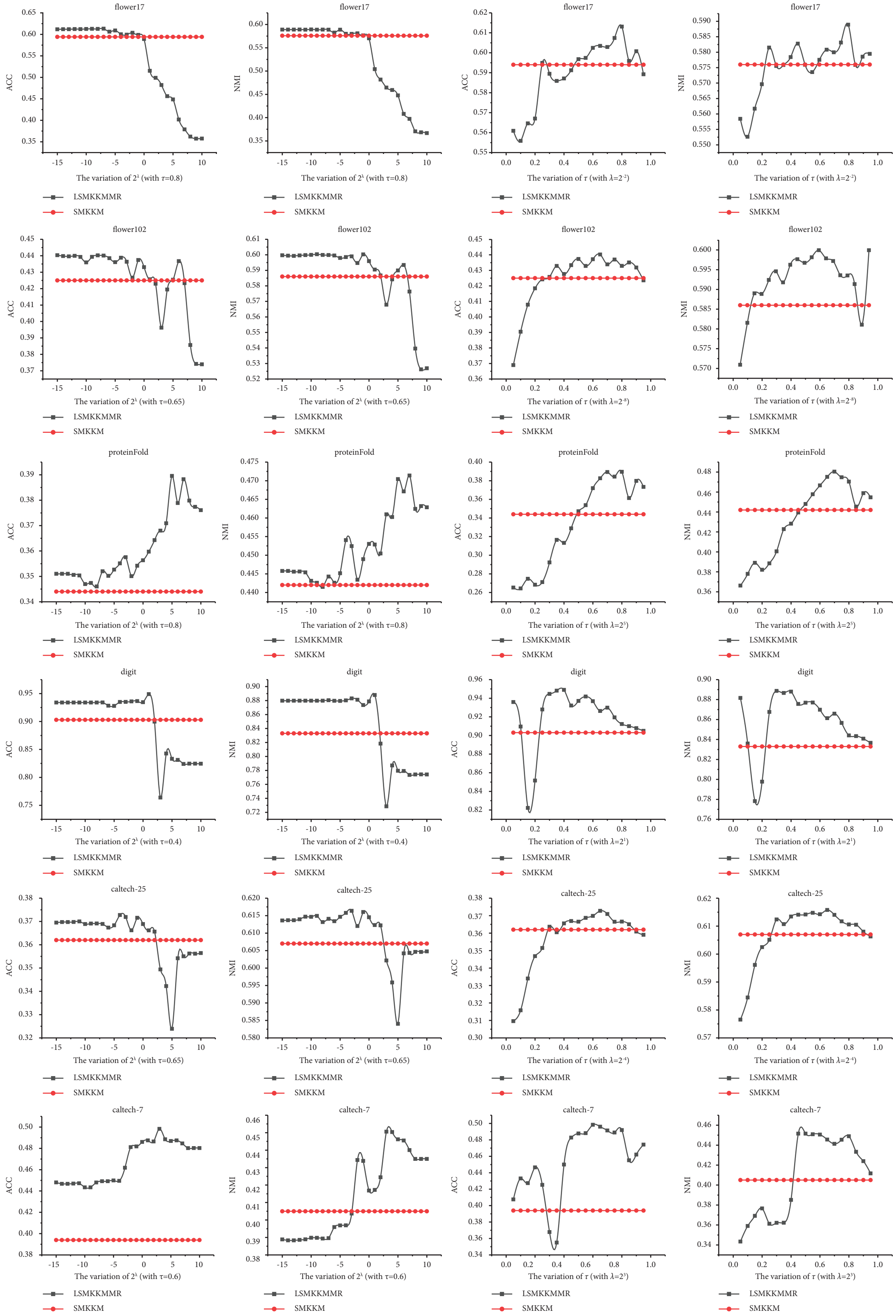

Figure 1: Sensitivity of the proposed method LI-SimpleMKKM-MR with a variation of λ and τ compared with SimpleMKKM.

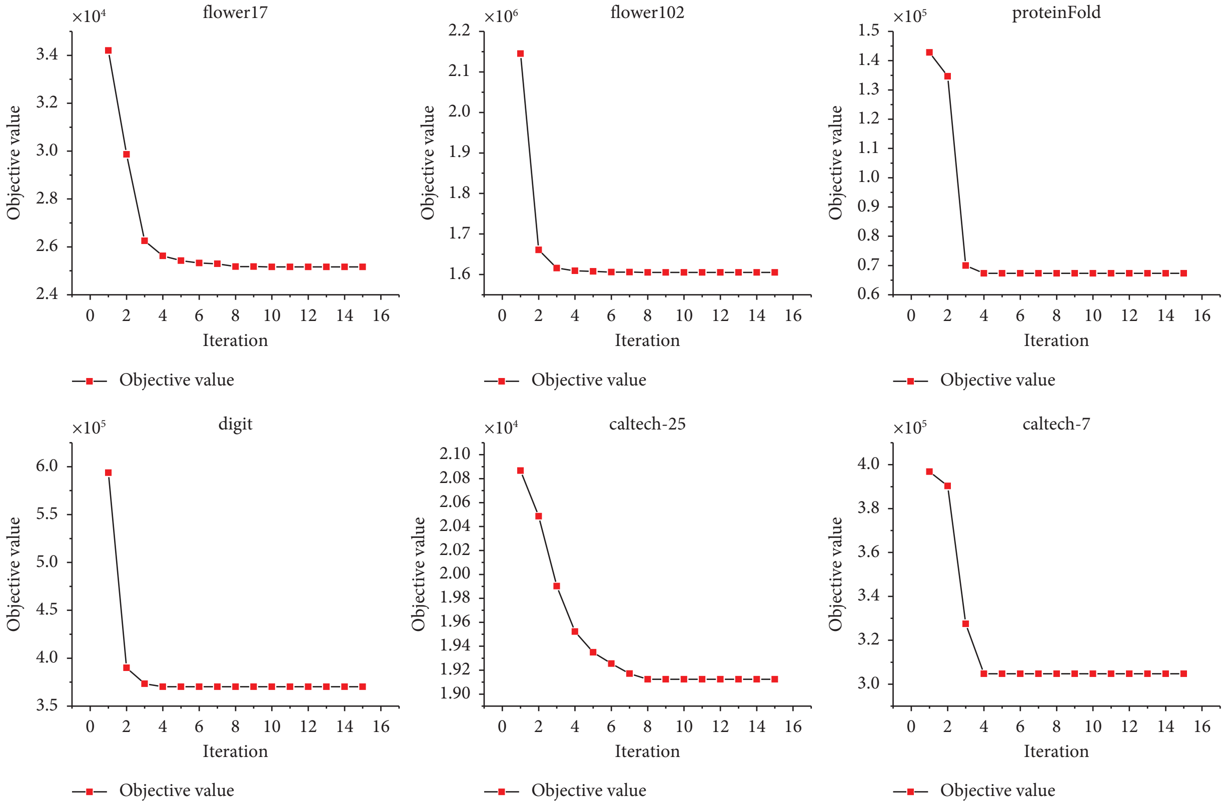

Figure 2: Proposed algorithm convergence illustration on flower17, flower102, ProteinFold, digit, Caltech-25, and Caltech-7 datasets.

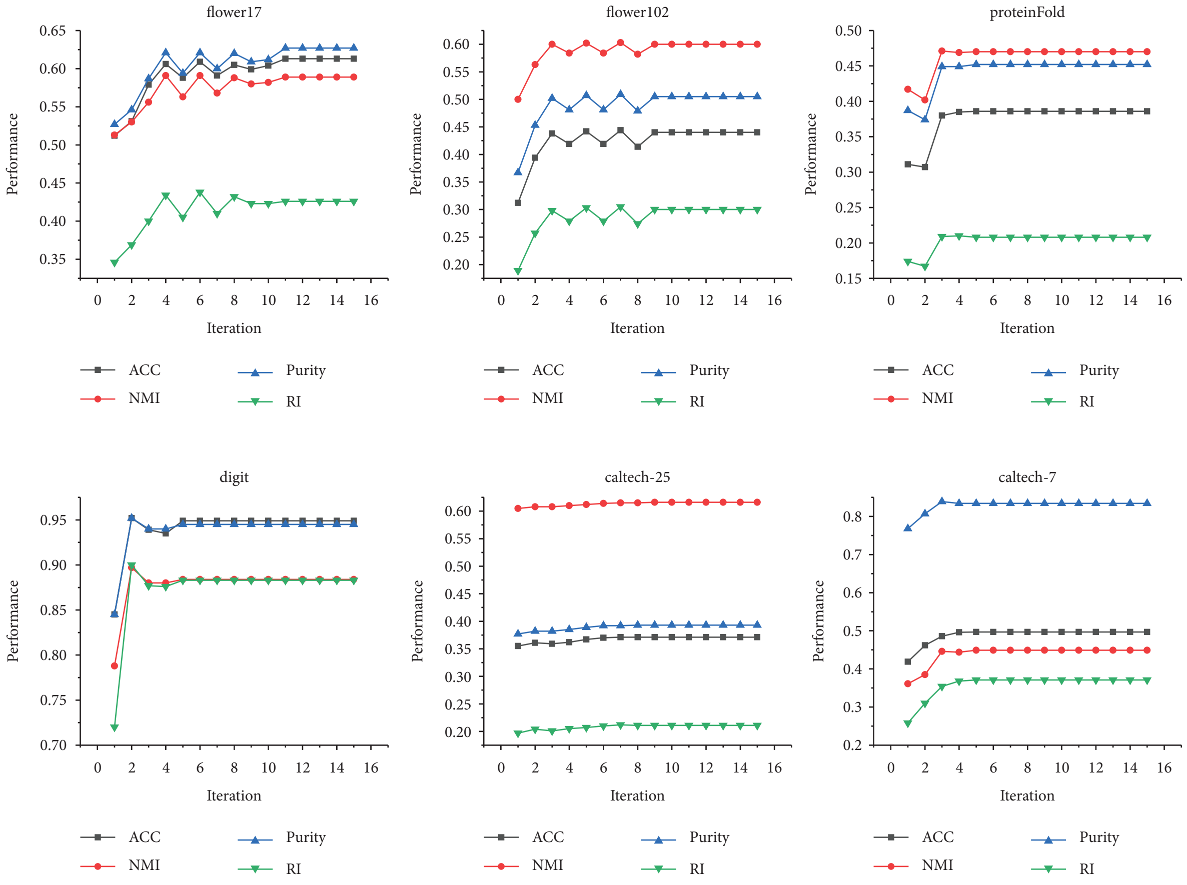

Figure 3: Clustering performance of **H** iteratively in LI-SimpleMKKM-MR learning on flower17, flower102, ProteinFold, digit, Caltech-25, and Caltech-7 datasets.

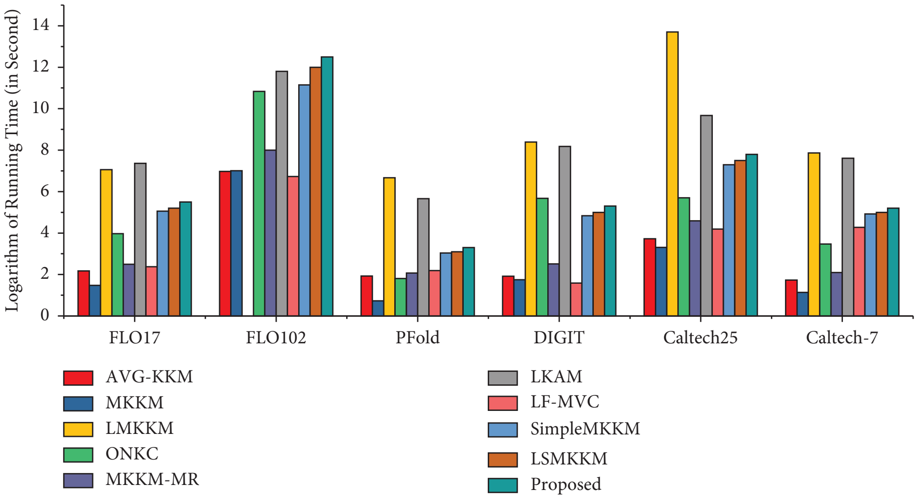

Figure 4: Running time comparison of different algorithms on all benchmark datasets (base 2 logarithm in seconds). The experimental environment is a desktop with Ubuntu 20.0 OS, Intel Core-i7-9700K cpu @ 3.60 GHz, 94.2 G RAM.

proposed algorithm using different datasets in Figure 2. According to the results, the object value of the proposed algorithm oscillates first, then decreases monotonically, and finally converges in several iterations. Moreover, we know from experiments that most datasets can converge in fewer than 10 iterations. This result is comparable to the state-of-the-art methods.

4.7. Performance of LI-SIMPLEMKKM-MR by Learned H. We calculate the 4 clustering metrics at each iteration to show the variety of clustering performance variations of the learned **H** in different datasets and plot them in Figure 3. As observed, the clustering performance increased firstly with iterations and remained stable after oscillation.

4.8. Running Time of LI-SimpleMKKM-MR. We report the running time comparison of all the baseline algorithms and LI-SimpleMKKM-MR on different datasets in Figure 4. With the analysis of the time complexity in Section 3.3 and the experiment result from Figure 4, even though there are additional computational steps, we found that LI-SimpleMKKM-MR does not significantly increase in computation time.

5. Conclusion

Although LI-SimpleMKKM can address the task of multiple kernel k-means in a $\min_{\eta} - \max_{H}$ optimization and realize the local alignment, it does not sufficiently account for the correlation between the basis kernels. This work proposes an LI-SimpleMKKM-MR algorithm that combines the sample localized alignment and matrix-induced regularization to solve this problem. Theoretically and experimentally, our method has demonstrated the best performance in clustering optimization and outperforms existing algorithms. In further research, we will apply this algorithm to incomplete MKKM problems.

Data Availability

The data that support the findings of this study are openly available at https://www.robots.ox.ac.uk/~+vgg/data/flowers/, https://mkl.ucsd.edu/dataset/protein-fold-prediction/, https://ss.sysu.edu.cn/~+py/, and https://files.is.tue.mpg.de/pgehler/projects/iccv09/.

Disclosure

A preprint has previously been published [26].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Jiaji Qiu and Huiying Xu equally contributed to the paper.

Acknowledgments

This work was supported by the Outstanding Talents of "Ten Thousand Talents Plan" in Zhejiang Province (project no. 2018R51001), Natural Science Foundation of China (project no. 61976196), and Zhejiang Provincial Natural Science Foundation of China under grant no. LZ22F030003.

References

[1] N. Y. Pehlivan and I. B. Turksen, "A novel multiplicative fuzzy regression function with a multiplicative fuzzy clustering algorithm," *Romanian Journal of Information Science and Technology*, vol. 24, no. 1, pp. 79–98, 2021.

[2] C. Pozna, R.-E. Precup, E. Horváth, and E. M. Petriu, "Hybrid particle filter–particle swarm optimization algorithm and application to fuzzy controlled servo systems," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 10, pp. 4286–4297, 2022.

[3] I.-D. Borlea, R.-E. Precup, and A.-B. Borlea, "Improvement of K-means cluster quality by post processing resulted clusters," *Procedia Computer Science*, vol. 199, pp. 63–70, 2022.

[4] C. Ieracitano, A. Paviglianiti, N. Mammone, M. Versaci, E. Pasero, and F. C. Morabito, “Socnnet: an optimized sobel filter based convolutional neural network for sem images classification of nanomaterials,” *Progresses in Artificial Intelligence and Neural Systems*, Springer, Singapore, 2020.

[5] J. Zhang and Z. Ma, “Hybrid fuzzy clustering method based on FCM and enhanced logarithmical PSO (ELPSO),” *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 1386839, 12 pages, 2020.

[6] R. De Paris, C. V. Quevedo, D. D. Ruiz, O. Norberto de Souza, and R. C. Barros, “Clustering molecular dynamics trajectories for optimizing docking experiments,” *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 916240, 9 pages, 2015.

[7] W. Gao, “Improved ant colony clustering algorithm and its performance study,” *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–14, Article ID 4835932, 2016.

[8] J. Zhou, D. Wang, L. Ling, M. Li, and K.-W. Lai, “Comparative analysis of the performance of complex texture clustering driven by computational intelligence methods using multiple clustering models,” *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 8449491, 13 pages, 2022.

[9] G. Qian, Y. Wu, D. Ferrari, P. Qiao, and F. Hollande, “Semisupervised clustering by iterative partition and regression with neuroscience applications,” *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 4037380, 13 pages, 2016.

[10] B. Zhao, J. T. Kwok, and C. Zhang, “Multiple kernel clustering,” in *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 638–649, Sparks, Nevada, May 2009.

[11] H.-C. Huang, Y.-Y. Chuang, and C. S. Chen, “Multiple kernel fuzzy clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, 2011.

[12] M. Gönen and A. A. Margolin, “Localized data fusion for kernel k-means clustering with application to cancer biology,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[13] L. Du, P. Zhou, L. Shi et al., “Robust multiple kernel k-means using l21-norm,” in *Proceedings of the 24th international joint conference on artificial intelligence*, pp. 3476–3482, Palo Alto, CA USA, July 2015.

[14] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, “Multiple kernel k-means clustering with matrix-induced regularization,” in *Proceedings of the AAAI conference on artificial intelligence*, pp. 1888–1894, Phoenix, Arizona, USA, February 2016.

[15] S. Wang, E. Zhu, J. Hu et al., “Efficient multiple kernel k-means clustering with late fusion,” *IEEE Access*, vol. 7, pp. 61109–61120, 2019.

[16] X. Liu, X. Zhu, M. Li et al., “Efficient and effective incomplete multi-view clustering,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4392–4399, Honolulu, Hawaii, USA, February 2019.

[17] Z.-z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann, “Double fusion for multimedia event detection,” in *Proceedings of the Advances in Multimedia Modeling: 18th International Conference MMM 2012*, pp. 173–185, Klagenfurt, Austria, January 2012.

[18] L. Houthuys, R. Langone, and J. A. Suykens, “Multi-view kernel spectral clustering,” *Information Fusion*, vol. 44, pp. 46–56, 2018.

[19] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, “From ensemble clustering to multi-view clustering,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2843–2849, Melbourne, Australia, August 2017.

[20] X. Liu, “Simplemkkm: simple multiple kernel k-means,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 1–13, 2022.

[21] X. Liu, S. Zhou, L. Liu et al., “Localized simple multiple kernel k-means,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9293–9301, Montreal, BC, Canada, October 2021.

[22] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, “Multiple kernel clustering with local kernel alignment maximization,” in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 1704–1710, New York, NY, USA, July 2016.

[23] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, “SimpleMKL,” *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[24] X. Liu, S. Zhou, Y. Wang et al., “Optimal neighborhood kernel clustering with multiple kernels,” in *Proceedings of the AAAI conference on artificial intelligence*, pp. 2266–2272, San Francisco, CA USA, February 2017.

[25] S. Wang, X. Liu, E. Zhu et al., “Multi-view clustering via late fusion alignment maximization,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 3778–3784, Macao China, August 2019.

[26] J. Qiu, H. Xu, and X. Zhu, “Localized simple multiple kernel K-means with matrix induced regularization,” 2020, https://arxiv.org/pdf/2005.04975.pdf.