Hyperparameter-Free Localized Simple Multiple Kernel K-means with Global Optimum

Xinwang Liu, *Senior Member, IEEE*

Abstract—The newly proposed localized simple multiple kernel k-means (SimpleMKKM) provides an elegant clustering framework which sufficiently considers the potential variation among samples. Although achieving superior clustering performance in some applications, we observe that it is required to pre-specify an extra hyperparameter, which determines the size of the localization. This greatly limits its availability in practical applications since there is a little guideline to set a suitable hyperparameter in clustering tasks. To overcome this issue, we firstly parameterize a neighborhood mask matrix as a quadratic combination of a set of pre-computed base neighborhood mask matrices, which corresponds to a group of hyperparameters. We then propose to jointly learn the optimal coefficient of these neighborhood mask matrices together with the clustering tasks. By this way, we obtain the proposed *hyperparameter-free localized SimpleMKKM*, which corresponds to a more intractable minimization-minimization-maximization optimization problem. We rewrite the resultant optimization as a minimization of an optimal value function, prove its differentiability, and develop a gradient based algorithm to solve it. Furthermore, we theoretically prove that the obtained optimum is the global one. Comprehensive experimental study on several benchmark datasets verifies its effectiveness, comparing with several state-of-the-art counterparts in the recent literature. The source code for hyperparameter-free localized SimpleMKKM is available at https://github.com/xinwangliu/SimpleMKKMcodes/.

Index Terms—Multiple Kernel Clustering, Multi-view Clustering, Clustering Ensemble.

1 Introduction

Multiple kernel clustering (MKC) provides an elegant learning framework to integrate complementary representation from different sources for clustering [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. With a number of given base kernel matrices, MKC optimally exploits them with the aim to partition similar samples into the same cluster and dissimilar ones into different clusters [14], [15]. Many MKC algorithms have been recently proposed and widely employed in practical applications [14], [15], [16]. In [17], an adaptive distance metric learning is proposed, which acts as a nonlinear extension for traditional k-means clustering. In [3], multiple kernel k-means (MKKM) is developed, in which the data coefficient and clustering assignment are alternately optimized until achieving convergence. The work in [18] incorporates a regularization into the objective of MKKM, with the aim to improve the usability of the selected kernels. By sufficiently considering the potential variation among samples, [14] proposes a localized MKKM variant which shows improved clustering results in applications. Motivated by the optimal neighborhood kernel learning, [15] puts forward an optimal neighborhood MKC algorithm, which is considered to be helpful in enhancing the representation ability of the optimal kernel. Different from the aforementioned MKKM based MKC algorithms, late fusion based MKC seeks to fuse multiple base partition matrices from multiple sources to achieve a consensus partition matrix [16].

As a representative of MKC, SimpleMKKM is recently proposed in [19]. Different from existing MKC algorithms which minimize the kernel weights and clustering partition matrix simultaneously, SimpleMKKM takes a novel minimization-maximization optimization framework which minimizes the kernel weights and maximizes the clustering partition matrix. This leads to an intractable minimization-maximization optimization. To optimize it, the work in [19] equivalently rewrites it as a minimization problem dependent on the clustering partition matrix, and applies a gradient based algorithm to minimize it. It is further shown that the obtained optimum is the global one. In addition, SimpleMKKM is free of hyperparameters. The ablation study empirically shows that both the novel minimization-maximization formulation and the new solving optimization algorithm attribute to its improved clustering performance.

Although SimleMKKM has the aforementioned advantages, it is pointed out in [20] that it globally maximizes the alignment between a weighted combination of base kernel matrices and an "ideal" similarity calculated by the pseudo-label matrix. This could require all pairwise samples to indiscriminately align to the same ideal similarity. Consequently, it is not able to effectively deal with the variation among samples and considerably utilize the local structures, resulting in unsatisfactory clustering performance. The work in [20] defines a local alignment criterion to overcome this issue. Specifically, it only requires maximizing the alignment between the combined kernel and ideal similarity matrix locally, i.e., in the range of the k-nearest neighborhood of each sample. This local criterion could guide clustering algorithms to concentrate on closer pairwise samples and avoid being affected by unreliable similarity of relatively farther ones. In [20], it is further shown that this localized variant can be encoded by

• *X. Liu is with School of Computer, National University of Defense Technology, Changsha, 410073, China (E-mail: xinwangliu@nudt.edu.cn).*

Manuscript submitted December 5, 2022.

element-wise multiplying each pre-specified kernel matrix with a neighborhood matrix, which is crucial to improving the clustering performance.

Localized SimpleMKKM has demonstrated superior clustering performance in various applications, as reported in [20]. However, we observe that its performance is dependent on a pre-calculated neighborhood matrix. *How to construct a suitable one for practical applications itself is intractable, especially for unsupervised learning tasks where class labels are missing*. Can we learn an optimal neighborhood matrix from data automatically? To fulfil this goal, this work firstly parameterizes the optimal neighborhood matrix as a quadratic combination of a set of pre-computed base neighborhood matrices, and jointly learns its optimal coefficient together with the clustering tasks. By this way, we can adaptively learn an optimal neighborhood matrix from data and avoid manual hyperparameter tuning. The resultant formulation induces a more difficult minimization-minimization-maximization optimization which cannot be solved by off-the-shelf alternate optimization anymore. We equivalently transform it as a minimization problem, and design a gradient based algorithm to optimize it. Extensive and substantial experimental results well indicate the superiority of our algorithm.

In sum, our work has the following main contributions.

- We identify that the recently proposed localized SimpleMKKM has to pre-specify a hyperparameter by hand, and develop a new learning paradigm to adjust it from data automatically. This learning strategy could also be applied to solve hyperparameter tuning in other learning tasks, especially in unsupervised learning scenarios.
- We parameterize the optimal neighborhood mask matrix as a quadratic combination of a set of base neighborhood mask ones, and jointly learn the optimal combination coefficient together with the optimal kernel weights and clustering partition matrix. This results in a more intractable tri-level optimization problem. To solve it, we reformulate it as a minimization problem, prove its differentiability, and develop a reduced gradient decent algorithm with guaranteed convergence to decrease it. Moreover, *we theoretically prove that the obtained solution is the global optimum.*
- We evaluate the clustering performance of the proposed hyperparameter-free localized SimleMKKM on six benchmark datasets. As seen, our algorithm has achieved superior clustering performance when compared with existing state-of-the-art competitors, verifying the effectiveness of the proposed joint learning paradigm.

Besides inheriting the flexibility of localized SimpleMKKM [20] in capturing the variation among samples, the proposed algorithm is hyperparameter-free, enabling it more applicable in practical applications.

2 Related work

In this part, we briefly review three algorithms which are closely relevant to our work, including multiple kernel k-means (MKKM) [21], simple multiple kernel k-means (SimpleMKKM) [19] and localized SimpleMKKM [20].

2.1 Multiple Kernel K-means

Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ denote a training set consisting of n samples with dimension d. k-means aims to partition $\mathbf{X}$ into k clusters $\{\mathbf{c}_j\}_{j=1}^k$ which are all disjoint. Let $\mathbf{U} \in \{0,1\}^{n\times k}$ be an indication matrix which is defined as

$$U_{ij} = \begin{cases} 1, & \text{the } i\text{-th point belongs to the } j\text{-th cluster,} \\ 0, & \text{otherwise.} \end{cases}$$

The standard k-means minimizes the following objective in Eq. (1).

$$\min_{\mathbf{U},\{\mathbf{c}_j\}_{j=1}^k} \ \frac{1}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{k} U_{ij}\|\mathbf{x}_i - \mathbf{c}_j\|^2, \tag{1}$$

in which $\sum_{j=1}^k U_{ij} = 1$, for any $i \in [n]$.

For a training set that is linearly inseparable, one can employ a feature map $\varphi(\cdot)$ to map samples into a separable Hilbert space $\mathcal{H}$ [22], then perform standard k-means on the mapped data. Instead given a feature mapping $\varphi(\cdot)$ explicitly, a kernel matrix is calculated as $K_{i,j} = \varphi(\mathbf{x}_i)^\top\varphi(\mathbf{x}_j)$ by the kernel trick. Based on the above definitions, the objective function of kernel k-means (KKM) is given as:

$$\min_{\mathbf{H}\in\Xi} \ \mathrm{Tr}\left(\mathbf{K}\left(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top\right)\right), \tag{2}$$

in which $\mathbf{H}$ is termed clustering partition matrix and $\Xi = \{\mathbf{H} \in \mathbb{R}^{n\times k} | \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\}$.

As well known, the clustering performance of KKM is largely affected by the choice of kernel functions. Instead of specifying a single kernel, the work in existing literature usually assumes that the optimal kernel matrix $\mathbf{K}_{\boldsymbol{\gamma}}$ is represented by a linear combination of base kernel matrices $\{\mathbf{K}_p\}_{p=1}^m$, i.e., $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^m \gamma_p^2\mathbf{K}_p$. By this way, KKM is extended to multiple kernel k-means (MKKM) whose objective is

$$\min_{\boldsymbol{\gamma}\in\Gamma} \ \min_{\mathbf{H}\in\Xi} \ \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I} - \mathbf{H}\mathbf{H}^\top)), \tag{3}$$

where $\Gamma = \{\boldsymbol{\gamma} \in \mathbb{R}^m \mid \sum_{p=1}^m \gamma_p = 1, \ \gamma_p \geq 0, \ \forall p\}$.

In literature, the optimization in Eq. (3) is usually solved by coordinate descent algorithms, where only one variable is optimized with the other fixed at each iteration.

Minimizing $\mathbf{H}$ with fixed $\boldsymbol{\gamma}$. With a given $\boldsymbol{\gamma}$, Eq. (3) w.r.t. $\mathbf{H}$ is equivalent to the following optimization,

$$\max_{\mathbf{H}\in\Xi} \ \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\right). \tag{4}$$

Eq. (4) is the kernel k-means, and can be readily solved by existing optimization packages.

Minimizing $\boldsymbol{\gamma}$ with fixed $\mathbf{H}$. With a given $\mathbf{H}$, Eq. (3) w.r.t. $\boldsymbol{\gamma}$ can be rewritten as the following optimization,

$$\min_{\boldsymbol{\gamma}\in\Gamma} \ \sum\nolimits_{p=1}^{m} \gamma_p^2 \mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right). \tag{5}$$

It is not easy to check that the optimal solution for Eq. (5) can be analytically obtained.

We present the whole optimization procedure in solving Eq. (3) in Algorithm 1, in which $\mathbf{H}$ and $\boldsymbol{\gamma}$ are alternately optimized until achieving convergence. After obtaining the optimal $\mathbf{H}$, the final clustering results can be acquired by performing the standard k-means on the rows of $\mathbf{H}$.

Algorithm 1 MKKM

Input: $\{\mathbf{K}_p\}_{p=1}^m,\ k$.
Initialize $\boldsymbol{\gamma}^{(1)} = \mathbf{1}_m/m$, flag $= 1$ and $t = 1$.
while flag **do**
  compute $\mathbf{H}^{(t)}$ in Eq. (4) with $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}} = \sum_{p=1}^m \left(\gamma_p^{(t)}\right)^2 \mathbf{K}_p$.
  update $\boldsymbol{\gamma}^{(t+1)}$ in Eq. (5) with $\mathbf{H}^{(t)}$.
  if $\max|\boldsymbol{\gamma}^{(t+1)} - \boldsymbol{\gamma}^{(t)}| \leq e^{-4}$ **then**
    flag $= 0$.
  end if
  $t \leftarrow t + 1$.
end while

Algorithm 2 SimpleMKKM [19]

Input: $\{\mathbf{K}_p\}_{p=1}^m,\ k$.
Output: $\mathbf{H}$, $\boldsymbol{\gamma}$.
Initialize $\boldsymbol{\gamma}^{(1)} = \mathbf{1}_m/m$, flag $= 1$ and $t = 1$.
while flag **do**
  compute $\mathbf{H}$ via performing kernel k-means on $\mathbf{K}_{\boldsymbol{\gamma}}$.
  compute $\frac{\partial \mathcal{J}(\boldsymbol{\gamma})}{\partial \gamma_p}\ (p = 1, \cdots, m)$ and the descent direction $\mathbf{d}^{(t)}$.
  update $\boldsymbol{\gamma}^{(t+1)} \leftarrow \boldsymbol{\gamma}^{(t)} + \alpha \mathbf{d}^{(t)}$.
  if $\max|\boldsymbol{\gamma}^{(t+1)} - \boldsymbol{\gamma}^{(t)}| \leq e^{-4}$ **then**
    flag=0.
  end if
  $t \leftarrow t + 1$.
end while

2.2 SimpleMKKM: Simple Multiple Kernel K-means

According to [19], it is empirically observed that the widely used $\min_{\boldsymbol{\gamma}} \min_{\mathbf{H}}$ paradigm of current MKKM algorithms could not achieve satisfactory clustering results in real-world applications. Its performance is even worse than the averaged kernel k-means in some applications. This encourages machine learning researchers to develop more innovative clustering algorithms. Different from the well-known $\min_{\boldsymbol{\gamma}} \min_{\mathbf{H}}$ learning paradigm [3], the recently proposed SimpleMKKM introduces a novel $\min_{\boldsymbol{\gamma}} \max_{\mathbf{H}}$ optimization framework, which is formulated as in Eq. (6).

$$\min_{\boldsymbol{\gamma}\in\Gamma}\ \max_{\mathbf{H}\in\Xi}\ \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top). \tag{6}$$

The novel minimization-maximization formalization in Eq. (6) cannot be solved by the widely adopted alternate optimization. In [19], the authors transform the $\min_{\boldsymbol{\gamma}} \max_{\mathbf{H}}$ into a $\min_{\boldsymbol{\gamma}}$ problem. Concretely, Eq. (6) is equivalently transformed to

$$\min_{\boldsymbol{\gamma}\in\Gamma}\ \mathcal{J}(\boldsymbol{\gamma}), \tag{7}$$

with

$$\mathcal{J}(\boldsymbol{\gamma}) = \left\{\max_{\mathbf{H}\in\Xi}\ \mathrm{Tr}\left(\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{H}\right)\right\}. \tag{8}$$

By this way, one rewrite the $\min_{\boldsymbol{\gamma}}$-$\max_{\mathbf{H}}$ optimization as a minimization one where its objective $\mathcal{J}(\boldsymbol{\gamma})$ is dependent on $\mathbf{H}$.

After theoretically showing the differentiability of $\mathcal{J}(\boldsymbol{\gamma})$, the work in [19] firstly computes the gradient of $\mathcal{J}(\boldsymbol{\gamma})$, derives its reduced gradient, and determines a feasible descent direction. The whole procedure in optimizing Eq. (6) is outlined in Algorithm 2.

In addition, the ablation study [19] on various benchmark datasets validates that both the novel minimization-maximization and new optimization attribute to the improved clustering performance. More detail on SimpleMKKM can be found in [19].

2.3 Localized SimpleMKKM

The recently proposed work in [20] tries to explore SimpleMKKM in a localized manner. $\mathbf{h}_i\,(1 \leq i \leq n)$ denotes the i-th row of clustering partition matrix $\mathbf{H}$. The alignment between $\mathbf{K}_{\boldsymbol{\gamma}}$ and $\mathbf{H}\mathbf{H}^\top$ in Eq. (6) is optimized in a *global* way. This implies that SimpleMKKM aligns each K_{ij} with a possible "ideal" value $\mathbf{h}_i^\top \mathbf{h}_j$ indiscriminately, ignoring the potential variation among samples. This would lead to aligning very various K_{ij}s with a same cluster label. It is therefore more reasonable to filter farther global similarity information that is unreliable and focus more on merging high confidence clustering predictions.

To achieve this goal, the work in [20] suggests to align $\mathbf{K}_{\boldsymbol{\gamma}}$ with $\mathbf{H}\mathbf{H}^\top$ in a localized manner. $\mathbf{S}^{(i)} \in \{0,1\}^{n\times \mathrm{round}(\tau\times n)}\,(\forall i)$ indicates the $\mathrm{round}(\tau \times n)$-closest neighborhoods of the i-th sample, in which τ is the proportion of localization and $\mathrm{round}(\cdot)$ is a rounding function. Based on this idea, one can calculate the local alignment for the i-th sample as follows,

$$\left\langle {\mathbf{S}^{(i)}}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{S}^{(i)},\ {\mathbf{S}^{(i)}}^\top \mathbf{H}^\top \mathbf{H} \mathbf{S}^{(i)} \right\rangle_{\mathrm{F}}, \tag{9}$$

where ${\mathbf{S}^{(i)}}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{S}^{(i)}$ indicates sampling elements from $\mathbf{K}_{\boldsymbol{\gamma}}$ according to the neighbors of the i-th sample. As shown, the localized alignment manner only requires closer samples to be kept together, which makes it better use of differences between kernels for final clustering. By accumulating the localized alignment in Eq. (9) for each sample, one can obtain the objective function of the localized SimpleMKKM as follows.

$$\min_{\boldsymbol{\gamma}\in\Gamma}\ \max_{\mathbf{H}\in\Xi}\ \mathrm{Tr}\left(\mathbf{H}^\top \left(\sum\nolimits_{i=1}^n \mathbf{A}^{(i)} \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{A}^{(i)}\right) \mathbf{H}\right), \tag{10}$$

where $\mathbf{A}^{(i)} = \mathbf{S}^{(i)}{\mathbf{S}^{(i)}}^\top$ denotes the neighborhood mask matrix corresponding to the i-th sample.

The following Theorem 1 uncovers the connection between SimpleMKKM and localized SimpleMKKM.

Theorem 1 ([20]). *The objection of the proposed localized SimpleMKKM in Eq. (10) can be rewritten as follows.*

$$\min_{\boldsymbol{\gamma}\in\Gamma}\ \max_{\mathbf{H}\in\Xi}\ \mathrm{Tr}\left(\mathbf{H}^\top \left(\mathbf{M} \otimes \mathbf{K}_{\boldsymbol{\gamma}}\right) \mathbf{H}\right), \tag{11}$$

where

$$\mathbf{M} = \sum\nolimits_{i=1}^n \mathbf{A}^{(i)} \tag{12}$$

is termed the neighborhood mask matrix.

According to Theorem 1, one can implement localized SimpleMKKM via SimpleMKKM by taking $\{\tilde{\mathbf{K}}_p\}_{p=1}^m$ as the input, where $\tilde{\mathbf{K}}_p = \mathbf{M} \otimes \mathbf{K}_p$.

3 Hyperparameter-Free Localized SimpleMKKM

3.1 The Proposed Formulation

Theorem 1 shows that one can encode the localization by element-wise multiplying each $\mathbf{K}_p$ with $\mathbf{M}$ in Eq. (12). On the one hand, the local alignment in Eq. (11) can sufficiently consider the variation among samples, which could help to enhance the clustering performance. On the other hand, there is an extra hyperparameter τ controlling the size of each sample's neighborhood, which is required to be pre-specified. However, it is well recognized in the literature that how to choose a suitable hyper-parameter in practical clustering tasks itself is a tough task, especially in the absence of class labels. It could be better to let clustering algorithms automatically learn the hyper-parameter. Following the multiple kernel learning framework, we assume that an optimal neighborhood mask matrix can be represented as a weighted combination of a group of base neighborhood mask matrices. That is, the optimal neighborhood mask matrix $\mathbf{M}$ in Eq. (12) can be parameterized as follows,

$$\mathbf{M}_{\boldsymbol{\mu}} = \sum\nolimits_{p=1}^{l} \mu_p^2 \mathbf{M}_p, \tag{13}$$

where $\{\mathbf{M}_p\}_{p=1}^{l}$ are a group of pre-specified neighborhood mask matrices corresponding to different sizes of the neighborhood, and $\boldsymbol{\mu}$ denotes their combination weights. As a result, choosing a suitable $\mathbf{M}$ reduces to learning an optimal combination weight $\boldsymbol{\mu}$.

By substituting $\mathbf{M}$ in Eq. (11) with $\mathbf{M}_{\boldsymbol{\mu}}$ in Eq. (13), the objective function for the proposed hyperparameter-free localized SimpleMKKM is as follows,

$$\min_{\boldsymbol{\gamma}\in\Gamma} \min_{\boldsymbol{\mu}\in\Theta} \max_{\mathbf{H}\in\Xi} \ \operatorname{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}} \otimes \mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right), \tag{14}$$

where $\Gamma = \{\boldsymbol{\gamma} \in \mathbb{R}^m \mid \sum_{p=1}^{m} \gamma_p = 1,\ \gamma_p \geq 0,\ \forall p\}$, $\Theta = \{\boldsymbol{\mu} \in \mathbb{R}^l | \boldsymbol{\mu}^\top \mathbf{e}_l = 1,\ \mu_p \geq 0,\ \forall p\}$, $\Xi = \{\mathbf{H} \in \mathbb{R}^{n\times k} | \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\}$, $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^{m} \gamma_p^2 \mathbf{K}_p$, and $\mathbf{M}_{\boldsymbol{\mu}} = \sum_{p=1}^{l} \mu_p^2 \mathbf{M}_p$.

We claim that the objective in Eq. (14) is with the following advantages: i) It calculates the kernel alignment locally, which enables it to capture the variation among base kernel matrices, leading to improved clustering performance. ii) The optimal hyper-parameter can be automatically learned from data. These advantages make the proposed algorithm more practical for applications. Though bearing such merits, the optimization in Eq. (14) is a minimization-minimization-maximization one, which makes it much more difficult to optimize than the one in SimpleMKKM. In the following, we equivalently rewrite Eq. (14) as a minimization, and adopt a gradient descent algorithm to decrease it.

3.2 Differentiability of the Objective Function

There are three variables to be optimized in Eq. (14). An alternate coordinate descent (ACD) is a common approach to optimization with multiple variables. However, if we use ACD to solve Eq. (14), it is observed that its objective function cannot be guaranteed to vary monotonically with iterations (See Figure 1 in [19] for the detail).

To solve it, we first equivalently rewrite Eq. (14) as a minimization of an optimal value function which depends on $\boldsymbol{\mu}$ and $\mathbf{H}$, and theoretically show its differentiability. Specifically, we firstly rewrite Eq. (14) as follows,

$$\min\nolimits_{\boldsymbol{\gamma}\in\Gamma} \ \mathcal{T}(\boldsymbol{\gamma}) \tag{15}$$

with

$$\mathcal{T}(\boldsymbol{\gamma}) = \Big\{ \min\nolimits_{\boldsymbol{\mu}\in\Theta} \max\nolimits_{\mathbf{H}\in\Xi} \ \operatorname{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}} \otimes \mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right) \Big\}. \tag{16}$$

Our goal is to use gradient descent to decrease $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15), which is defined in Eq. (16) and dependent on $\boldsymbol{\mu}$ and $\mathbf{H}$. To fulfil this goal, we have to prove the differentiability of $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15). To prove the differentiability, we firstly introduce the following Lemma 1 in [19].

Lemma 1 ([19]). *$\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (8) is convex w.r.t $\boldsymbol{\gamma}$.*

Lemma 1 concludes that the solution optimized by Algorithm 2 is the global optimum. With a given $\boldsymbol{\gamma}$, the optimization in Eq. (16) is the same to the one in Eq. (6), which can be directly solved by Algorithm 2, generating the global optimum. According to Lemma 1, we have the following Theorem 2.

Theorem 2. *$\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15) is differentiable w.r.t $\boldsymbol{\gamma}$. Further, $\frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_p} = 2\gamma_p \operatorname{Tr}\left(\mathbf{H}^{*\top}(\mathbf{M}_{\boldsymbol{\mu}^*} \otimes \mathbf{K}_p)\mathbf{H}^*\right)$, where $(\mathbf{H}^*, \boldsymbol{\mu}^*) = \{\arg\min_{\boldsymbol{\mu}\in\Theta} \max_{\mathbf{H}\in\Xi} \ \operatorname{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}} \otimes \mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right)\}$.*

Proof. With a given $\boldsymbol{\gamma}$, we conclude that the solution of Eq. (16) is unique according to Lemma 1. Based on Theorem 4.1 in [23], $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15) is differentiable w.r.t $\boldsymbol{\gamma}$. Further, $\frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_p} = 2\gamma_p \operatorname{Tr}\left(\mathbf{H}^{*\top}(\mathbf{M}_{\boldsymbol{\mu}^*} \otimes \mathbf{K}_p)\mathbf{H}^*\right)$, where $(\mathbf{H}^*, \boldsymbol{\mu}^*) = \{\arg\min_{\boldsymbol{\mu}\in\Theta} \max_{\mathbf{H}\in\Xi} \ \operatorname{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}} \otimes \mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right)\}$. □

3.3 The Descent Direction and Optimization Algorithm

After calculating the gradient of $\mathcal{T}(\boldsymbol{\gamma})$ according to Theorem 2, we, in the following, show how to determine a descent direction which can guarantee its equality and non-negative constraints. To achieve this goal, we firstly compute the reduced gradient of $\mathcal{T}(\boldsymbol{\gamma})$ to keep the equality constraint according to [19], [24].

We can randomly select a positive component of $\boldsymbol{\gamma}$, denoted as γ_v. Let $\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})$ represent the reduced gradient of $\mathcal{T}(\boldsymbol{\gamma})$. Then, we can calculate the p-th element of $\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})$ as follows,

$$[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_p = \frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_p} - \frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_v} \ \ \forall\, p \neq v, \tag{17}$$

and

$$[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_v = \sum\nolimits_{p=1, p\neq v}^{m} \left(\frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_v} - \frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_p}\right), \tag{18}$$

with $1 \leq p \leq m$. According to the suggestion in [19], [24], we select v to be the index corresponding to the largest component of $\boldsymbol{\gamma}$, which is usually able to maintain better numerical stability.

We then consider the non-negative constraints on $\boldsymbol{\gamma}$. To minimize $\mathcal{T}(\boldsymbol{\gamma})$, we can take $-\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})$ as a feasible descent direction. However, if we directly take it as a descent direction, the non-negative constraints may not be kept anymore when there is an index q such that $\gamma_q = 0$ and its reduced gradient $[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_q > 0$. In such a case, we should

set the descent direction for that component as 0. Together considering the equality and non-negative constraints, we give the descent direction of the p-th component as follows,

$$d_p = \begin{cases} 0 & \text{if } \gamma_p = 0 \text{ and } [\bigtriangledown \mathcal{T}(\boldsymbol{\gamma})]_p > 0 \\ -[\bigtriangledown \mathcal{T}(\boldsymbol{\gamma})]_p & \text{if } \gamma_p > 0 \text{ and } p \neq v \\ -[\bigtriangledown \mathcal{T}(\boldsymbol{\gamma})]_v & \text{if } p = v. \end{cases} \tag{19}$$

After calculating a descent direction $\mathbf{d} = [d_1, \cdots, d_m]^\top$ according to Eq. (19), we can update $\boldsymbol{\gamma}$ by $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} + \alpha \mathbf{d}$, where α is called learning rate. In our implementation, we determine it by the widely adopted Armijo's rule. Other one-dimensional line search strategies are also worth trying. We outline the whole optimization procedure to solve Eq. (14) in Algorithm 3.

Algorithm 3 Hyperparameter-free Localized SimpleMKKM

Input: $\{\mathbf{K}_p\}_{p=1}^m$, $\{\mathbf{M}_p\}_{p=1}^l$ and k.
Output: $\mathbf{H}$ and $\boldsymbol{\gamma}$, $\boldsymbol{\mu}$.
Initialize $\boldsymbol{\gamma}^{(0)} = \mathbf{1}_m/m$, $\boldsymbol{\mu}^{(0)} = \mathbf{1}_l/l$ and $t = 1$.
while flag **do**
    $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}} = \sum_{p=1}^m \left(\gamma_p^{(t-1)}\right)^2 \mathbf{K}_p$.
    compute $(\mathbf{H}, \boldsymbol{\mu})$ by SimpleMKKM in Algorithm 2 with $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}}$.
    compute $\frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_p}$ $(p = 1, \cdots, m)$ and the descent direction $\mathbf{d}^{(t)}$ in Eq. (19).
    update $\boldsymbol{\gamma}^{(t+1)} \leftarrow \boldsymbol{\gamma}^{(t)} + \alpha \mathbf{d}^{(t)}$.
    if $\max |\boldsymbol{\gamma}^{(t+1)} - \boldsymbol{\gamma}^{(t)}| \leq e^{-4}$ **then**
        flag=0.
    end if
    $t \leftarrow t + 1$.
end while

3.4 Global Convergence Analysis

In the following, we analyze the global convergence of our algorithm in Algorithm 3 by calculating the Hessian matrix of $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15), as stated in Theorem 3.

Theorem 3. *$\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15) is convex w.r.t. $\boldsymbol{\gamma}$.*

Proof. According to Theorem 2, $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15) is differentiable w.r.t $\boldsymbol{\gamma}$ and $\frac{\partial \mathcal{T}(\boldsymbol{\gamma})}{\partial \gamma_p} = 2\gamma_p \mathrm{Tr}(\mathbf{H}^{*\top} (\mathbf{M}_{\boldsymbol{\mu}^*} \otimes \mathbf{K}_p) \mathbf{H}^*)$. Furthermore, $\frac{\partial \mathcal{T}^2(\boldsymbol{\gamma})}{\partial \gamma_p \gamma_q} = 2\mathrm{Tr}(\mathbf{H}^{*\top} (\mathbf{M}_{\boldsymbol{\mu}^*} \otimes \mathbf{K}_p) \mathbf{H}^*)$ if $p = q$, and 0 otherwise. Therefore, the Hessian matrix of $\mathcal{T}(\boldsymbol{\gamma})$ is a diagonal matrix with elements $2[a_1, \cdots, a_m]^\top$, where $a_p = \mathrm{Tr}(\mathbf{H}^{*\top} (\mathbf{M}_{\boldsymbol{\mu}^*} \otimes \mathbf{K}_p) \mathbf{H}^*)$, $1 \leq p \leq m$. In addition, since both $\mathbf{M}_{\boldsymbol{\mu}^*}$ and $\mathbf{K}_p$ are symmetric positive definite, we have $a_p = \mathrm{Tr}(\mathbf{H}^{*\top} (\mathbf{M}_{\boldsymbol{\mu}^*} \otimes \mathbf{K}_p) \mathbf{H}^*) \geq 0$. As seen, the Hessian matrix is positive definite, which indicates that $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (15) is convex w.r.t. $\boldsymbol{\gamma}$. □

With a given $\boldsymbol{\gamma}$, Eq. (16) achieves the global optimum. Under this condition, the gradient calculation in Theorem 2 is exact. A reduced gradient descent algorithm is then performed on $\mathcal{T}(\boldsymbol{\gamma})$ which is a continuously differentiable function defined on $\Gamma = \{\boldsymbol{\gamma} \in \mathbb{R}^m | \sum_{p=1}^m \gamma_p = 1, \gamma_p \geq 0, \forall p\}$. This guarantees that the solution obtained by Algorithm 3 converges to the minimum of $\mathcal{T}(\boldsymbol{\gamma})$. Furthermore, according to Theorem 3, the minimum of Algorithm 3 is the global optimum, which is also validated by experiments in Figure 4.

3.5 Computational Complexity

In this subsection, we discuss the computational complexity of the proposed hyperparameter-free localized SimpleMKKM. According to Algorithm 3, at each iteration, it involves solving a SimpleMKKM problem, computing the reduced gradient, and searching for an optimal learning rate. According to [19], the computational complexity of SimpleMKKM at each iteration is $\mathcal{O}(\ell_s * n^3)$, where ℓ_s denotes the minimum of iterations to achieve convergence. The computational complexity of computing the reduced gradient and searching for an optimal learning rate are $\mathcal{O}(m * n^2)$ and $\mathcal{O}(\ell_r * m)$, where ℓ_r represents the operation of searching the optimal step size. As a result, the computational complexity of our algorithm at each iteration is $\mathcal{O}(\ell_s * n^3 + m * n^2 + \ell_r * m)$. As observed, our algorithm keeps a similar computational complexity to existing MKKM and SimpleMKKM algorithms, as validated by the experiments in Figure 5.

4 Experiments

4.1 Experimental Settings

Comprehensive experiments have been conducted on several publicly available MKKM datasets. They are *Wdbc*[1] 569/10/2, *ProteinFold*[2] 694/12/27, *Flower17*[3] 1360/7/17, *Caltech*[4] 1530/25/102, *Handwritten*[5] 2000/6/10, *Flower102*[6] 8189/4/102, SunRgbd 10335/2/45 and ALOI 10800/4/100. The three numbers above indicate the numbers of samples, kernels and clusters, respectively. For example, Flower102 dataset has 8189 samples, 4 kernels and 102 clusters. The size of datasets, the number of kernels and categories show considerable variation, which provides an excellent platform to make a performance comparison among the aforementioned algorithms. We generate a group of base neighborhood mask matrices $\{\mathbf{M}_p\}_{p=1}^l$ according to the definition in Eq. (11). Since the neighbor number is defined by $\mathrm{round}(\tau \times n)$, eight τs, i.e., $0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 1$, are pre-defined to generate base neighborhood masks.

For each benchmark, the class number k is given and taken as the input of all algorithms. Four commonly adopted clustering evaluation criteria, i.e., clustering accuracy (ACC), normalized mutual information (NMI), purity, and rand index (RI) are used for algorithm validation. To alleviate the interference of random initialization caused by the k-means algorithm, the test procedure with random initializations is implemented for 50 times. Both the mean value and the variation of the 50 trials are reported.

To evaluate the superiority of our algorithm, the following nine state-of-the-art (SOTA) multiple kernel clustering algorithms are included for comparison.

1. http://archive.ics.uci.edu/ml/datasets/
2. http://mkl.ucsd.edu/dataset/protein-fold-prediction
3. http://www.robots.ox.ac.uk/~vgg/data/flowers/17/
4. http://www.vision.caltech.edu/ImageDatasets/Caltech101
5. http://archive.ics.uci.edu/ml/datasets/
6. http://www.robots.ox.ac.uk/~vgg/data/flowers/102/

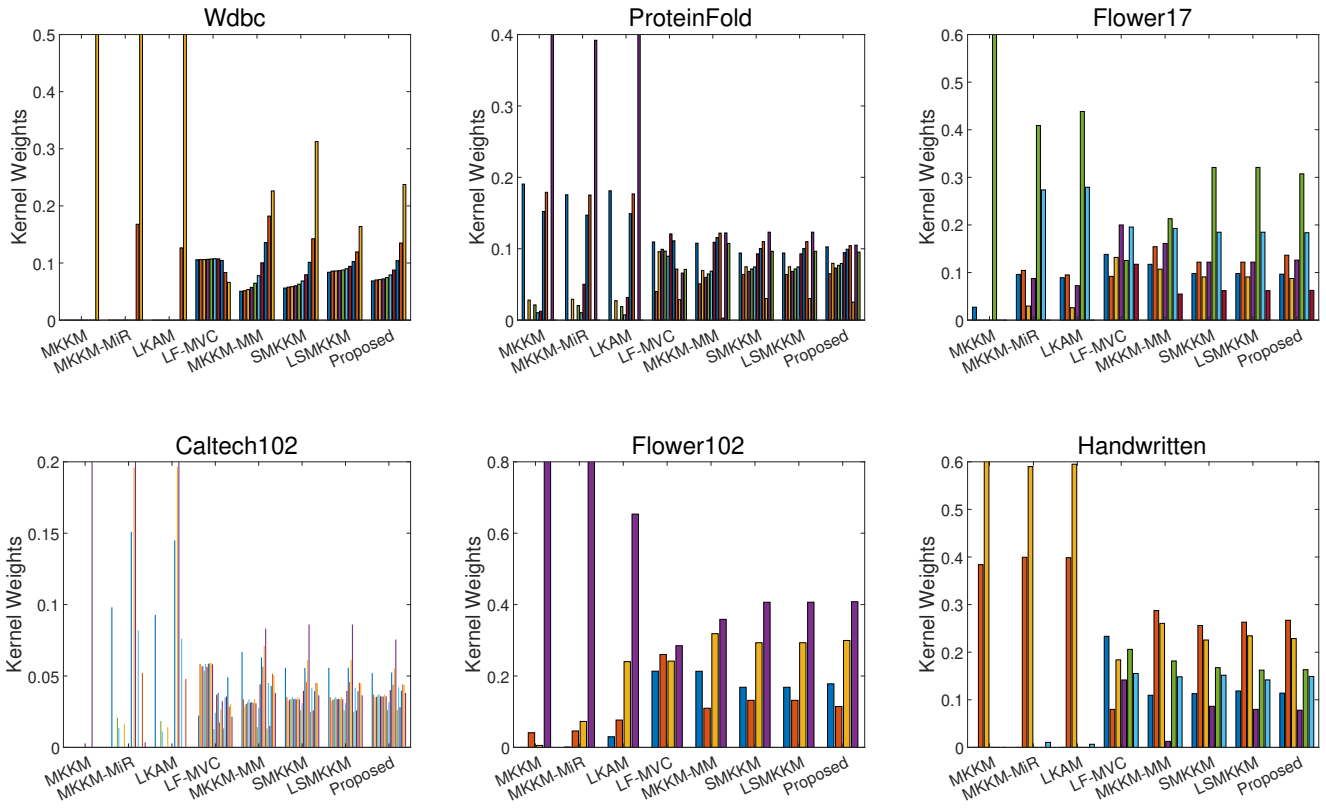
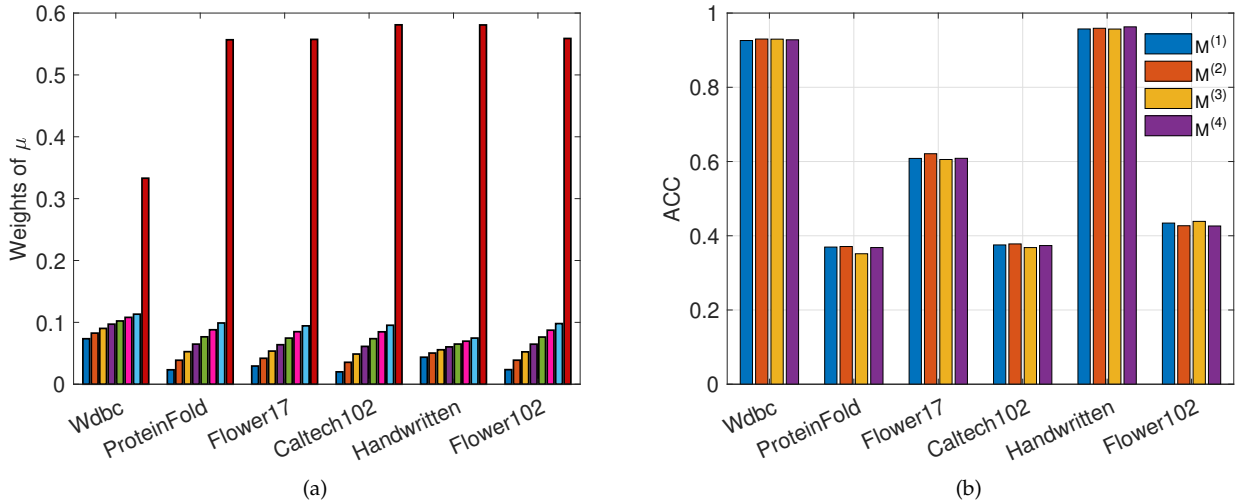


Fig. 1: The kernel weights learned by the proposed parameter-free localized SimpleMKKM and the compared algorithms.



(a) (b)

Fig. 2: (a) The learned $\boldsymbol{\mu}$ by the proposed parameter-free localized SimpleMKKM. (b) The clustering performance with four different groups of mask matrices.

DATASET	AVG-MKKM	MKKM [25]	LMKKM [26]	MKKM-MIR [18]	LKAM [14]	LF-MVC [16]	MKKM-MM [27]	SMKKM [19]	LSMKKM [20]	PROPOSED
					ACC					
WDBC	91.0 ± 0.0	91.0 ± 0.0	91.0 ± 0.0	81.5 ± 0.0	79.4 ± 0.0	91.0 ± 0.0	91.0 ± 0.0	90.5 ± 0.0	92.1 ± 0.0	**93.0 ± 0.0**
PROTEINFOLD	29.0 ± 1.5	27.0 ± 1.1	22.4 ± 0.7	34.7 ± 1.8	**37.7 ± 1.2**	33.0 ± 1.4	29.0 ± 1.5	34.7 ± 1.9	34.7 ± 1.9	37.1 ± 1.6
FLOWER17	50.8 ± 1.5	44.9 ± 2.4	37.5 ± 1.6	58.5 ± 1.5	50.0 ± 0.8	61.0 ± 0.7	50.8 ± 1.5	59.5 ± 1.3	59.5 ± 1.2	**62.1 ± 0.7**
CALTECH102	34.2 ± 1.0	32.8 ± 0.9	27.9 ± 0.8	34.8 ± 1.0	32.3 ± 1.0	34.4 ± 1.3	34.2 ± 1.0	35.8 ± 0.7	35.9 ± 0.7	**37.8 ± 0.7**
HANDWRITTEN	**96.0 ± 0.0**	64.9 ± 2.4	65.0 ± 1.4	88.7 ± 0.1	95.4 ± 3.5	95.8 ± 0.0	96.0 ± 0.0	93.6 ± 0.0	**96.5 ± 3.0**	95.9 ± 3.0
FLOWER102	27.1 ± 0.8	22.4 ± 0.5	-	40.2 ± 0.9	41.4 ± 0.8	38.4 ± 1.2	27.1 ± 0.8	42.5 ± 0.8	42.5 ± 0.8	**42.7 ± 1.0**
SUNRGBD	18.5 ± 0.5	17.2 ± 0.6	-	19.5 ± 0.5	**19.6 ± 0.5**	18.6 ± 0.6	18.5 ± 0.5	19.2 ± 0.5	19.2 ± 0.5	19.4 ± 0.2
ALOI	64.5 ± 1.3	6.4 ± 0.1	67.3 ± 1.4	**68.5 ± 1.5**	65.2 ± 1.0	68.4 ± 1.4	64.5 ± 1.3	64.3 ± 1.4	66.4 ± 1.3	68.4 ± 1.0
					NMI					
WDBC	55.2 ± 0.0	55.0 ± 0.0	55.0 ± 0.0	36.3 ± 0.0	34.2 ± 0.0	55.3 ± 0.0	55.2 ± 0.0	54.3 ± 0.0	58.9 ± 0.0	**62.5± 0.0**
PROTEINFOLD	40.3 ± 1.3	38.0 ± 0.6	34.7 ± 0.6	43.7 ± 1.2	46.2 ± 0.6	41.7 ± 1.1	40.3 ± 1.3	44.4 ± 1.1	44.4 ± 1.1	**46.7 ± 1.0**
FLOWER17	49.7 ± 1.0	44.9 ± 1.5	38.8 ± 1.1	56.4 ± 0.9	49.8 ± 0.6	58.9 ± 0.4	49.7 ± 1.0	57.8 ± 0.9	57.8 ± 0.9	**60.5 ± 0.6**
CALTECH102	59.3 ± 0.6	58.6 ± 0.5	55.3 ± 0.5	59.7 ± 0.5	58.5 ± 0.6	59.5 ± 0.6	59.3 ± 0.6	60.4 ± 0.5	60.4 ± 0.5	**62.3 ± 0.4**
HANDWRITTEN	91.1 ± 0.1	64.8 ± 1.6	64.7 ± 0.5	79.4 ± 0.2	91.8 ± 1.9	90.9 ± 0.1	91.1 ± 0.1	87.4 ± 0.0	**93.6 ± 1.6**	92.0 ± 1.8
FLOWER102	46.0 ± 0.5	42.7 ± 0.2	-	56.7 ± 0.5	56.9 ± 0.3	54.9 ± 0.4	46.0 ± 0.5	58.6 ± 0.5	58.6 ± 0.5	**59.4 ± 0.3**
SUNRGBD	22.6 ± 0.3	21.2 ± 0.4	-	23.5 ± 0.3	23.9 ± 0.3	22.6 ± 0.4	22.6 ± 0.3	23.1 ± 0.4	23.1 ± 0.4	**24.9 ± 0.2**
ALOI	77.7 ± 0.7	22.3 ± 0.2	79.7 ± 0.5	**80.9 ± 0.6**	78.2 ± 0.4	79.6 ± 0.5	77.7 ± 0.7	77.7 ± 0.7	78.9 ± 0.5	80.7 ± 0.4
					PURITY					
WDBC	91.0 ± 0.0	91.0 ± 0.0	91.0 ± 0.0	81.5 ± 0.0	79.4 ± 0.0	91.0 ± 0.0	91.0 ± 0.0	90.5 ± 0.0	92.1 ± 0.0	**93.0 ± 0.0**
PROTEINFOLD	37.4 ± 1.7	33.7 ± 1.1	31.2 ± 1.0	41.9 ± 1.4	43.7 ± 0.8	39.3 ± 1.5	37.4 ± 1.7	41.8 ± 1.5	41.8 ± 1.5	**44.3 ± 1.4**
FLOWER17	51.9 ± 1.5	46.2 ± 2.0	39.2 ± 1.3	59.7 ± 1.6	51.4 ± 0.7	62.4 ± 0.7	51.9 ± 1.5	60.9 ± 1.2	60.9 ± 1.2	**63.4 ± 1.0**
CALTECH102	36.2 ± 1.0	34.9 ± 0.9	29.6 ± 0.8	36.8 ± 0.8	34.3 ± 0.9	36.7 ± 1.3	36.2 ± 1.0	38.0 ± 0.7	38.0 ± 0.7	**40.4 ± 0.8**
HANDWRITTEN	96.0 ± 0.0	65.8 ± 2.1	65.5 ± 0.9	88.7 ± 0.1	95.4 ± 3.5	95.8 ± 0.0	96.0 ± 0.0	93.6 ± 0.0	**96.5 ± 2.9**	96.1 ± 2.5
FLOWER102	32.3 ± 0.6	27.8 ± 0.4	-	46.3 ± 0.8	48.0 ± 0.6	44.6 ± 0.8	32.3 ± 0.6	48.6 ± 0.7	48.6 ± 0.7	**49.6 ± 0.7**
SUNRGBD	38.2 ± 0.7	36.2 ± 0.7	-	39.4 ± 0.6	39.6 ± 0.4	38.1 ± 0.6	38.2 ± 0.7	39.0 ± 0.6	18.4 ± 0.5	**39.9 ± 0.2**
ALOI	77.7 ± 0.7	22.3 ± 0.2	79.7 ± 0.5	**80.9 ± 0.6**	78.2 ± 0.4	79.6 ± 0.5	77.7 ± 0.7	77.7 ± 0.7	68.3 ± 1.1	80.7 ± 0.4
					RAND INDEX					
WDBC	67.2 ± 0.0	67.2 ± 0.0	67.2 ± 0.0	39.7 ± 0.0	34.5 ± 0.0	67.2 ± 0.0	67.2 ± 0.0	65.5 ± 0.0	70.7 ± 0.0	**73.8 ± 0.0**
PROTEINFOLD	14.4 ± 1.8	12.1 ± 0.7	7.8 ± 0.4	17.2 ± 1.5	20.1 ± 1.1	16.1 ± 1.5	14.4 ± 1.8	17.6 ± 1.9	17.6 ± 1.9	**20.3 ± 2.0**
FLOWER17	32.2 ± 1.3	27.2 ± 1.8	20.6 ± 1.1	39.9 ± 1.3	31.6 ± 0.8	44.1 ± 0.4	32.2 ± 1.3	41.5 ± 1.5	41.5 ± 1.5	**44.8 ± 0.7**
CALTECH102	18.4 ± 0.9	17.3 ± 0.7	13.4 ± 0.8	18.8 ± 0.8	16.8 ± 0.9	18.8 ± 1.0	18.4 ± 0.9	19.8 ± 0.7	19.8 ± 0.7	**21.8 ± 0.7**
HANDWRITTEN	91.3 ± 0.0	51.8 ± 2.3	50.4 ± 1.2	77.2 ± 0.2	91.6 ± 3.5	91.0 ± 0.1	91.3 ± 0.0	86.5 ± 0.1	**93.5 ± 2.8**	91.9 ± 3.0
FLOWER102	15.5 ± 0.5	12.1 ± 0.4	-	25.5 ± 0.6	27.2 ± 0.6	25.5 ± 1.0	15.5 ± 0.5	28.5 ± 0.8	28.5 ± 0.8	**28.8 ± 0.9**
SUNRGBD	8.9 ± 0.3	8.1 ± 0.3	-	9.6 ± 0.3	9.9 ± 0.3	9.0 ± 0.2	8.9 ± 0.3	9.4 ± 0.3	9.4 ± 0.3	**10.3 ± 0.1**
ALOI	51.4 ± 1.5	2.0 ± 0.1	55.2 ± 1.1	**56.5 ± 1.1**	53.9 ± 0.9	54.3 ± 1.2	51.4 ± 1.5	51.5 ± 1.4	54.8 ± 1.2	56.4 ± 0.9

TABLE 1: The ACC, NMI, Purity and Rand Index comparison of the proposed algorithm with baseline methods on six benchmark datasets. The best results are marked in bold.

- **Average kernel k-means (Avg-KKM)**. A consensus kernel is firstly constructed by linearly combining the base kernels with the same weight and then taken as the kernel input k-means.
- **Multiple kernel k-means (MKKM)** [25]. The linear combination weights and the cluster indicating matrix are optimized simultaneously in a unified optimization framework.
- **Localized multiple kernel k-means (LMKKM)** [26]. A sample-adaptive base kernel combination mechanism is proposed to boost the clustering results of MKKM.
- **Multiple kernel k-means with matrix-induced regularization (MKKM-MiR)** [18]. A regularization term is integrated into the MKKM learning to enhance diverse information preservation.
- **Multiple kernel clustering with local alignment maximization (LKAM)** [14]. It learns an optimal kernel combination by aligning the ideal similarity matrix with the combined kernel matrix within only the neighborhood district.
- **Multi-view clustering via late fusion alignment maximization (LF-MVC)** [16]. It first calculates the base partitions associated with corresponding views and then integrates them into a united partition matrix.
- **MKKM-MM** [27]. It introduces a $\min_{\mathbf{H}}\text{-}\max_{\boldsymbol{\gamma}}$ formulation that integrates different views in a way indicating high within-cluster variance in the consensus kernel space and optimizes the clusters through minimizing such variance.
- **SimpleMKKM (SMKKM)** [19]. It introduces a special $\min\text{-}\max$ clustering formulation for kernel weights and cluster partition optimization.
- **Localized SimpleMKKM (LSMKKM)** [20]. It uses the min-max optimization paradigm of SMKKM, and proposes to adopt a localized manner to extract the information of kernel matrices.

The implementations of the aforementioned algorithms are publicly available. Among the compared algorithms, LKAM [14], MKKM-MiR [18], LF-MVC [16] and LSMKKM [20] have at least one hyperparameter to be tuned. We follow the algorithm settings of their original papers and run the publicly available source codes. Besides, we tune the corresponding hyperparameters by grid search. The best clustering performance and standard deviation of these algorithms are reported. As seen, the clustering performance of algorithms with hyperparameters is over-estimated.

4.2 Experimental Comparison and Discussion

4.2.1 Clustering Results

As illustrated in Table 1, we report the clustering performances, including ACC, NMI, purity, and RI, of all afore-

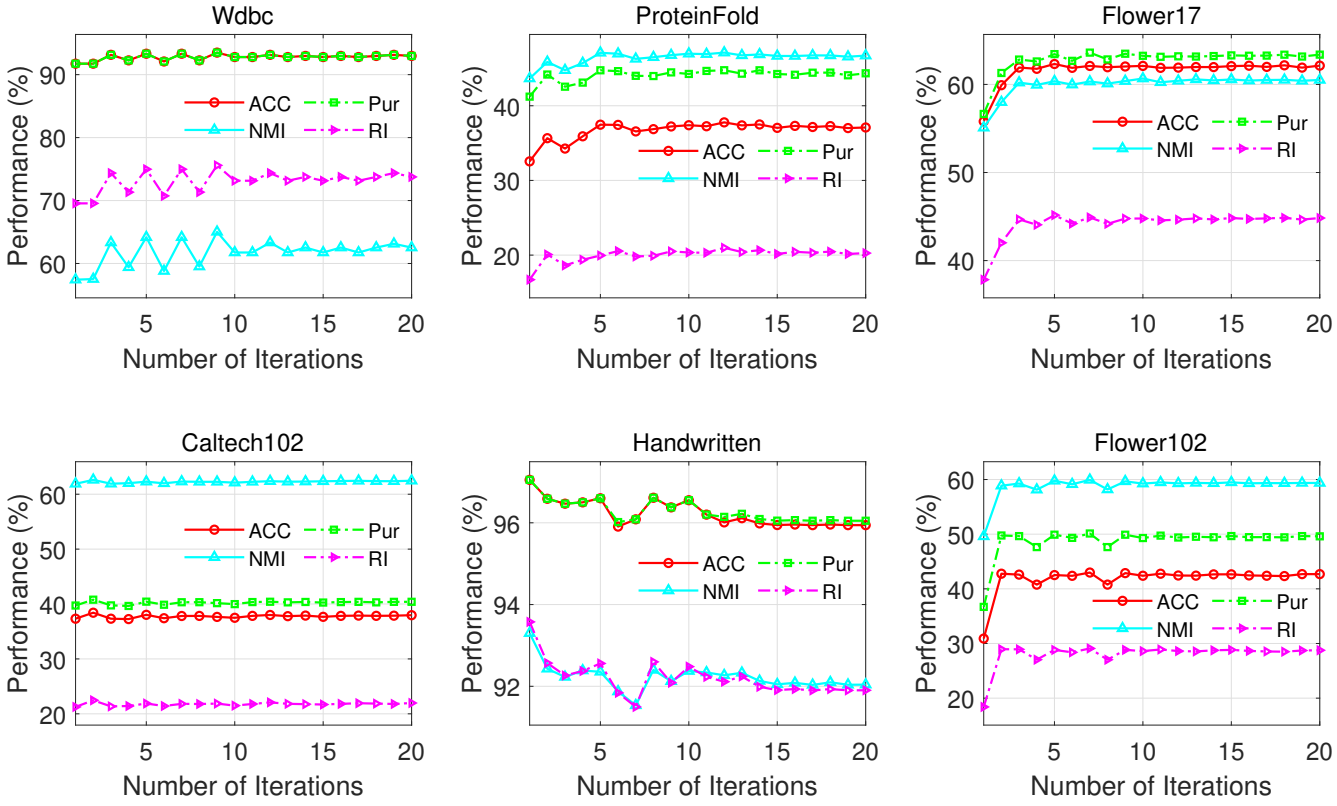


Fig. 3: The evolution of the learned $\mathbf{H}$ by the proposed algorithm with iterations.

mentioned algorithms. From these results, several observations can be concluded:

- The proposed parameter-free localized SimpleMKKM significantly outperforms the algorithms with hyper-parameters, like LF-MVC [16] and LSMKKM [20]. This demonstrates the practicability and efficacy of our formulation.
- SimpleMKKM [19], which adopts a similar min-max optimization formulation with our proposed algorithm, and achieves comparable or better clustering performance then the algorithms with hyper-parameters on most benchmark datasets. This superiority can be attributed to the novelty of its formulation with a reasonable optimization mechanism.
- Our parameter-free localized SimpleMKKM consistently outperforms all compared algorithms by a significant margin. Here we take the NMI clustering metric, for instance, and our proposed algorithm exceeds the LSMKKM algorithm by 3.6%, 2.3%, 2.7%, 1.9%, and 0.8% on Wdbc, ProteinFold, Flower17, Caltech102, and Flower102 datasets, respectively. Moreover, our algorithm exceeds the SimpleMKKM algorithm by 8.2%, 2.3%, 2.7%, 1.9%, 4.6%, 0.8% and exceeds the LF-MVC algorithm by 7.3%, 5.0%, 1.6%, 2.8%, 1.1%, 4.5% on six benchmark dataset, respectively. The enhancements with respect to other matrices are similar. The above clustering results have solidly illustrated the effectiveness of our parameter-free localized SimpleMKKM. This is because it benefits from adaptively learning the local information of the kernel matrix.
- The proposed parameter free localized SimpleMKKM performs better than MKKM-MiR [18], LKAM [14], LF-MVC [16], and LSMKKM [20], where several hyper-parameters are required to tune associated with regularization on the kernel weights. Thus they require much effort to choose the best hyper-parameters in real-world applications. Moreover, parameter tuning is very difficult or even impossible in practical scenarios where no ground truth is available. Differently, the proposed algorithm is parameter-free.

Apart from inheriting the carefully-designed formulation and advanced optimization from SimpleMKKM, this improved algorithm adaptively employs a localized learning manner to conduct the kernel alignment among different kernels. This makes the algorithm more suitable for kernel variation. These benefits jointly bring significant improvement over its counterparts on all datasets. In addition, we point out that LMKKM [26] cannot get the results reported on some datasets since it suffers from the risk of being out of memory. This is mainly caused by its huge memory and heavy computational complexity.

4.2.2 *Analysis of Kernel Weight*

In this subsection, we further analyze the kernel weights learned by the aforementioned algorithms. As observed in Figure 1, the kernel weights learned by MKKM, MKKM-MiR, and LKAM are distributed very unevenly and relatively sparse on almost all datasets. This sparsity indicates

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3233635

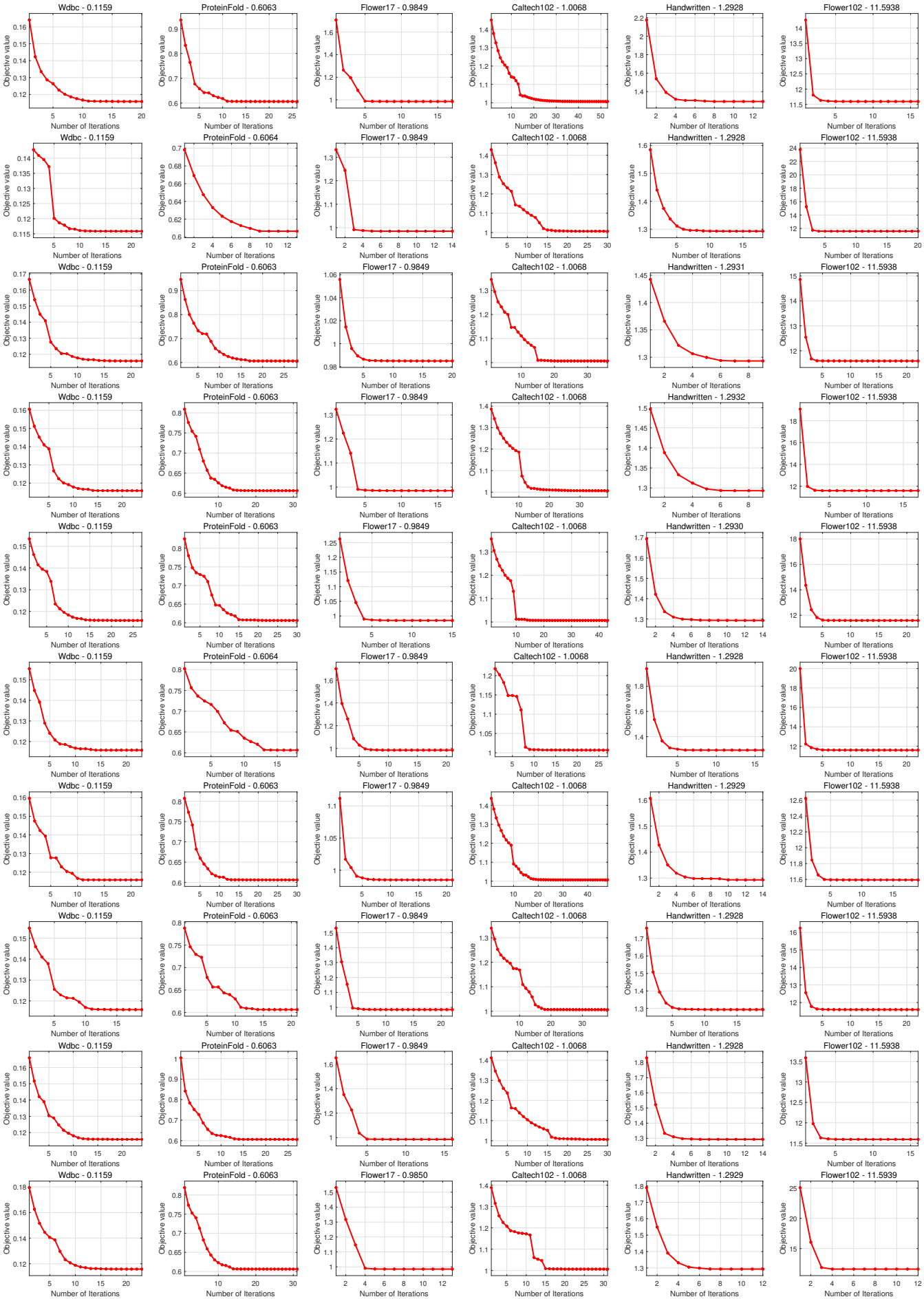


Fig. 4: The objective curves of Hyperparameter Free Localized SimpleMKKM under ten different initializations on Wdbc, ProteinFold, Flower17, Flower102, and Handwritten. Though with different initializations, the objective value stops at the same point.

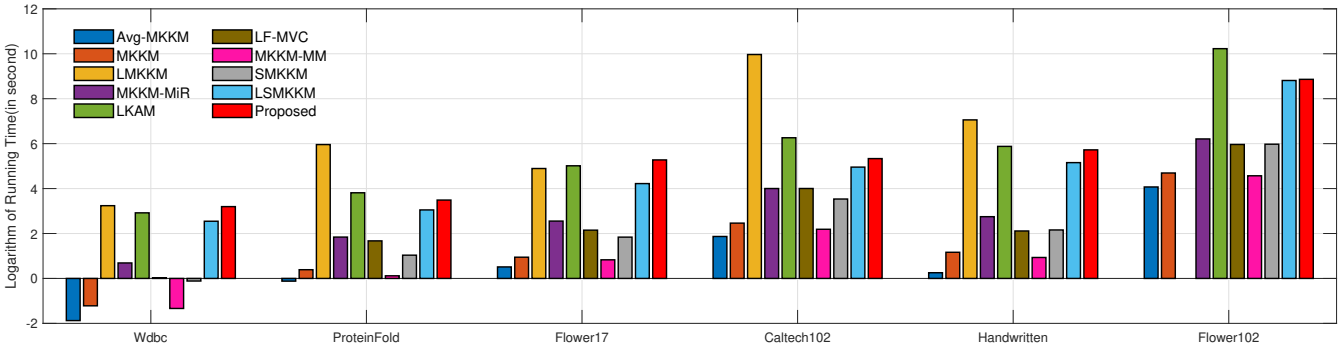


Fig. 5: Running time of the aforementioned algorithms on all datasets (logarithm in seconds). These algorithms are run on a PC with Intel(R) Core(TM)-i9-10900X 3.7GHz CPU and 64G RAM in MATLAB R2020b environment.

the insufficient exploration of the multiple kernel matrices and results in the unsatisfactory performance of MKKM. For instance, the ACC of MKKM, MKKM-MiR and, LKAM on Flower17 is only $44.9\%, 58.5\%$, and 50.0%, respectively. Differently, despite the ℓ_1-norm constraint on $\boldsymbol{\gamma}$, the kernel weights of our proposed algorithm have non-sparse properties, which contributes to its promising results on all datasets. The non-sparsity can be attributed to the proposed reduced gradient descent optimization, which in turn is derived on the basis of the proposed $\min$-$\max$ kernel alignment criterion.

4.2.3 Analysis of Mask Matrix Weight

Here, we analyze the mask matrix weights learned by the proposed algorithm, and the results on all datasets are plotted in sub-figure 2a. As observed, the obtained $\boldsymbol{\mu}$ is non-sparse, which indicates that each individual mask matrix contributes to the construction of the optimal mask matrix. We also try four different groups of $\{\mathbf{M}_p^{(q)}\}_{p=1}^l (1 \leq q \leq 4)$. The results are reported in sub-figure 2b. As seen, the performance of our algorithm is almost the same under different groups of $\{\mathbf{M}_p\}_{p=1}^l$, which shows that its clustering performance can be further boosted by incorporating prior knowledge to constructing base mask matrices, which is worth further exploring.

4.2.4 Global Convergence of the Proposed Algorithm

According to Theorem 3, our parameter-free localized SimpleMKKM is theoretically guaranteed convergent to a global optimum. To illustrate this point better, we further present the objective curves of parameter-free localized SimpleMKKM with iterations *under different initializations*. From the results on all datasets in Figure 4, we can find that: i) its objective monotonically decreases and usually converges in ten iterations. ii) Although the proposed optimization starts from different initializations, the objective value converges to the same value, validating the global convergence of our algorithm.

4.2.5 Evolution of the learned $\mathbf{H}$

To reveal the performance variation of the learned $\mathbf{H}$ with the number of iterations, we calculate four clustering metrics with iterations and present the corresponding results in Figure 3. It can be observed that the performance of the proposed algorithm firstly increases at each iteration and soon keeps stable. This phenomenon considerably verifies the effectiveness of the Learned $\mathbf{H}$.

4.2.6 Running Time Comparison

For a fair comparison, we empirically evaluate the running time of compared algorithms to evaluate the computational efficiency on all datasets, as illustrated in Figure 5. As seen, besides significantly improving the clustering performance, our proposed parameter-free localized SimpleMKKM also has a comparable time cost with other counterparts. Note that the hyper-parameter tuning time is also included for these algorithms with hyperparameters.

5 Conclusion

While the newly proposed localized SimpleMKKM is able to capture the variation among samples and demonstrates promising clustering performance, it needs to pre-specify the size of the neighborhood. However, how to select a suitable hyperparameter for unsupervised learning tasks is an open issue. In this work, we transfer the hyperparameter selection task to a learning one via parameterization, leading to a more intractable optimization. We then build a new optimization algorithm with global convergence to solve it. Our proposed algorithm demonstrates largely increased clustering results via substantial experiments on multiple benchmark datasets. Many future work are worth exploring. For example, the performance of our algorithm is dependent on $\{\mathbf{M}_p\}_{p=1}^l$, and how to utilize prior knowledge to construct them to further boost its clustering performance is worth studying. Also, we plan to enable the hyperparameter-free localized SimpleMKKM to deal with incomplete kernels in future work.

Acknowledgments

This work was supported by the National Key R&D Program of China 2020AAA0107100, the Natural Science Foundation of China (Project No. 61922088).

REFERENCES

[1] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *NIPS*, 2004, pp. 1537–1544.

[2] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *ICDM*. IEEE, 2009, pp. 1016–1021.

[3] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1031–1039, 2012.

[4] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *ICML*, 2011, pp. 393–400.

[5] Z. Huang, P. Hu, J. T. Zhou, J. Lv, and X. Peng, "Partially view-aligned clustering," in *Annual Conference on Neural Information Processing Systems 2020*.

[6] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: multi-view clustering without parameter selection," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, pp. 5092–5101.

[7] W. Liu, X. Shen, and I. W. Tsang, "Sparse embedded k-means clustering," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 3319–3327.

[8] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, vol. 28, pp. 352–360.

[9] S. Zhou, E. Zhu, X. Liu, T. Zheng, Q. Liu, J. Xia, and J. Yin, "Subspace segmentation-based robust multiple kernel clustering," *Information Fusion*, vol. 53, pp. 145–154, 2020.

[10] W. Liang, S. Zhou, J. Xiong, X. Liu, S. Wang, E. Zhu, Z. Cai, and X. Xu, "Multi-view spectral clustering with high-order optimal neighborhood laplacian matrix," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3418–3430, 2022, doi: 10.1109/TKDE.2020.3025100.

[11] Z. Kang, X. Zhao, C. Peng, H. Zhu, J. T. Zhou, X. Peng, W. Chen, and Z. Xu, "Partition level multiview subspace clustering," *Neural Networks*, vol. 122, pp. 279–288, 2020.

[12] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1582–1590.

[13] C. Zhang, H. Fu, J. Wang, W. Li, X. Cao, and Q. Hu, "Tensorized multi-view subspace representation learning," *International Journal of Computer Vision*, pp. 1–18, 2020.

[14] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *International Joint Conference on Artificial Intelligence*, 2016, pp. 1704–1710.

[15] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 2266–2272.

[16] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization." in *IJCAI*, 2019, pp. 3778–3784.

[17] J. Chen, Z. Zhao, J. Ye, and H. Liu, "Nonlinear adaptive distance metric learning for clustering," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 123–132.

[18] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k -means clustering with matrix-induced regularization," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1888–1894.

[19] X. Liu, "Simplemkkm: Simple multiple kernel k-means," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–13, 2022, doi: 10.1109/TPAMI.2022.3198638.

[20] X. Liu, S. Zhou, L. Liu, C. Tang, S. Wang, J. Liu, and Y. Zhang, "Localized simple multiple kernel k-means," in *ICCV*. IEEE, 2021, pp. 9273–9281.

[21] H. Huang, Y. Chuang, and C. Chen, "Multiple kernel fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120–134, 2012.

[22] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[23] J. F. Bonnans and A. Shapiro, "Optimization problems with perturbations: A guided tour," *SIAM Review*, vol. 40, no. 2, pp. 228–264, 1998.

[24] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *JMLR*, vol. 9, pp. 2491–2521, 2008.

[25] 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, 2012.

[26] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Advances in Neural Information Processing Systems*, 2014, pp. 1305–1313.

[27] S. Bang, Y. Yu, and W. Wu, "Robust multiple kernel k-means clustering using min-max optimization," *ArXiv preprint*, 2018, doi: 10.48550/ARXIV.1803.02458.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China, in 2013. He is now Professor at School of Computer, NUDT. His current research interests include kernel learning, multi-view clustering and unsupervised feature learning. Dr. Liu has published 100+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, CVPR, ICCV, AAAI, IJCAI, etc. He is an Associate Editor of IEEE T-NNLS and Information Fusion Journal. More information can be found at https://xinwangliu.github.io/.