Journal of Machine Learning Research 1 (2021) 1-17 Submitted 9/2021; Published xx/2022

Parameter-free Localized SimpleMKKM

Xinwang Liu XINWANGLIU@NUDT.EDU.CN
School of Computer, National University of Defense Technology, Changsha, Hunan, CHINA

Editor: Kevin Murphy and Bernhard Schölkopf

Abstract

The recently proposed simple multiple kernel k-means (SimpleMKKM) provides an elegant framework to optimally fuse multiple views of samples for clustering. Although demonstrating improved clustering performance on various applications, we observe that it *indiscriminately* forces the similarity constructed by features to be identically aligned with the similarity constructed by pseudo-labels. Such a criterion does not sufficiently consider the potential variation among kernel matrices, which could negatively affect the clustering performance. To address this issue, we propose a localized SimpleMKKM which only requires that the sub-similarity calculated by k-nearest neighbours of a sample be aligned with the one generated by corresponding pseudo-labels. We show that this localization can be encoded by element-wise multiplying each pre-specified kernel matrix with a neighborhood mask matrix. We further parameterize the neighborhood mask matrix as a quadratic combination of a group of pre-specified base neighborhood mask matrices, and jointly learn the optimal coefficient together with the clustering tasks, learning to the proposed *parameter-free localized SimpleMKKM*. After that, we rewrite the resultant optimization as an optimal value function, prove its differentiability, and develop a reduced gradient descent algorithm with proved convergence to solve it. Comprehensive experimental study on several benchmark datasets verifies its effectiveness, comparing with several state-of-the-art counterparts in the recent literature.

Keywords: Multiple Kernel Clustering, Multi-view Clustering

1. Introduction

Multiple kernel clustering (MKC) gives an subtle framework to assign samples into different clusters by extracting complementary information from different sources (Xu et al., 2004; Tang et al., 2009; Yu et al., 2012; Kumar and Daumé, 2011; Huang et al.; Peng et al.; Liu et al., 2017a; Wang et al.; Zhou et al., 2020; Liang et al., 2020; Kang et al., 2020; Zhang et al., 2015, 2020). Given a group of pre-defined kernel matrices, MKC integrates the available multiple kernel information to distribute data samples with similar structures or patterns into the same cluster, which has been substantially studied and commonly applied into practice applications (Li et al., 2016; Liu et al., 2017b; Li et al., 2016; Wang et al., 2019). For example, the work in (Chen et al., 2007) proposes a MKC algorithm which acts as the nonlinear extension for traditional k-means clustering problem. (Yu et al., 2012) proposes to iteratively optimize data coefficient and clustering assignment until convergence. Moreover, the work in (Liu et al., 2016) proposes a multiple kernel k-means clustering algorithm, which decreases the redundancy of the selected kernels by introducing a matrix-induced regularization term. A local kernel alignment variant is then developed

by sufficiently considering the variation among sample, which is experimentally verified to improve the clustering result in (Li et al., 2016). By assuming an optimal kernel residing in the neighborhood of the combined kernels, the work in (Liu et al., 2017b) proposes an optimal neighborhood multiple kernel clustering algorithm, which improves the clustering performance by enhancing the representability of the learned optimal kernel. Differently, late fusion based multiple kernel clustering strategy seeks to exploit the complementary information in kernel partition space to achieve consensus on partition level (Wang et al., 2019). Specifically, the pioneering work in (Wang et al., 2019) proposes to maximally align the multiple base partitions with the consensus partition, which enjoys considerable algorithm acceleration and satisfactory clustering performance. Along this line, an effective and efficient late fusion based algorithm is proposed in (Liu et al., 2019) to handle incomplete multi-view data.

As a masterpiece of MKC, SimpleMKKM is recently proposed in (Liu et al., 2020). Instead of jointly minimizing the kernel weights and clustering partition matrix, SimpleMKKM takes a minimization on kernel weights and maximization on clustering partition matrix optimization framework, leading to an intractable min-max optimization. After that, it is equivalently transformed into a minimization problem and a reduced gradient descent algorithm is taken to solve the resultant optimization. It is empirically observed that the novel min-max formulation and new solving optimization algorithm attribute to its improved clustering performance.

Although the recently proposed SimleMKKM bears the aforementioned merits, we observe that it strictly aligns the combined kernel matrix with an "ideal" similarity generated by the clustering partition matrix in a *global* way. This could indiscriminately guide all sample pairs to consistently align to the same ideal similarity. As a result, it does not effectively handle the variation among samples and sufficiently consider local structures, which could lead to unsatisfactory clustering performance. To solve this problem, we propose to calculate the kernel alignment in a "local" manner, which only requires that the generated combined kernel be aligned with the ideal similarity matrix locally in the k-nearest neighborhood of each sample. Such a *localized alignment* guides the clustering algorithm to concentrate on *closer* sample pairs and avoid being affected by unreliable similarity evaluation of relatively *farther* sample pairs. We derive the objective function of proposed formulation based on the min-max optimization framework of SimpleMKKM. We show that this localized variant can be encoded by element-wise multiplying each pre-specified kernel matrix with a neighborhood matrix, which is crucial to improve the clustering performance. However, how to construct an optimal neighborhood matrix for practical applications itself is intractable, especially for unsupervised learning tasks. To address this issue, we further parameterize the optimal neighborhood matrix as a quadratic combination of a group of pre-specified base neighborhood matrices, and jointly learn its optimal coefficient together with the clustering tasks, learning to the proposed parameter-free localized SimpleMKKM. The resultant formulation induces a more difficult min-min-max optimization which is hard to readily solve by existing alternate optimization. We reformulate it as an optimal value function, prove its *differentiability*, and develop a *reduced gradient decent* algorithm with guaranteed convergence to solve it. Extensive and substantial experimental results well demonstrate the superiority of the proposed algorithm.

The main contributions of this work are summarized as follows,

- We identify that the recently proposed SimpleMKKM is unable to effectively handle the variation among kernel matrices for the first time, and develop a parameter-free local kernel alignment criterion to address this issue.
- We uncover the theoretical connection between our proposed algorithm with SimpleMKKM via an optimal neighborhood mask matrix, which is set as a quadratic combination of a group of base neighborhood mask matrices. Further, we jointly learn the optimal combination coefficient during clustering, obtaining a parameter-free multi-view clustering algorithm.
- We develop a reduced gradient decent algorithm with proved convergence to efficiently solve the resultant min-min-max optimization problem.

2. Related work

In this section, we briefly introduce multiple kernel k-means (MKKM) (Huang et al., 2012a) and the recently proposed simple multiple kernel k-means (SimpleMKKM) (Liu et al., 2020), which are closely related to our work.

2.1 Multiple Kernel K-means

Given $\mathbf{X} \in \mathbb{R}^{n\times d}$ with n and d the number of samples and feature dimensions, k-means clustering aims to group $\mathbf{X}$ into k clusters. Let $\mathbf{Z} \in \{0,1\}^{n\times k}$ be a clustering assignment matrix, where $Z_{iq} = 1$ if $\mathbf{x}_i$ belongs to the q-th cluster, other $Z_{iq} = 0$. Its objective can be presented as

$$\min_{\mathbf{Z},\{\mathbf{c}_q\}_{q=1}^k} \ \frac{1}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{q=1}^{k} Z_{iq}\|\mathbf{x}_i - \mathbf{c}_q\|^2 \tag{1}$$

in which $\sum_{q=1}^{k} Z_{iq} = 1$, $\forall i$.

Considering that samples may not well clustered in its original space, they are usually mapped into a reproducing kernel Hilbert space (RKHS) (Scholkopf and Smola, 2001) $\mathcal{H}$ with a feature map $\varphi(\cdot)$, i.e. $\phi_i = \varphi(\mathbf{x}_i)$, and clustered by k-means in that space. Note that the mapping function $\varphi(\cdot)$ is usually implicitly defined, one can construct a kernel matrix with $K_{i,j} = \phi_i^\top \phi_j$. Based on these definition, the objective function of kernel k-means can be rewritten as

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \ \mathrm{Tr}\left(\mathbf{K}\left(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top\right)\right) \ s.t. \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{2}$$

in which $\mathbf{H}$ is termed clustering partition matrix and $\mathbf{I}_k$ is an identity matrix with size k.

It is well known that the performance of kernel k-means is largely dependent on the choice of kernel matrix. By assuming that the optimal kernel $\mathbf{K}_{\boldsymbol{\gamma}}$ can be expressed as a combination of pre-specified base kernel matrices, the objective function in Eq. (2) can be readily extended to multiple kernel k-means, with the objective as follows,

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

where $\Delta = \{\boldsymbol{\gamma} \in \mathbb{R}^m \mid \sum_{p=1}^{m}\gamma_p = 1, \ \gamma_p \geq 0, \ \forall p\}$ and $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^{m}\gamma_p^2\mathbf{K}_p$. In literature, a two-step alternate optimization with proved convergence is developed to jointly optimize $\boldsymbol{\gamma}$

and $\mathbf{H}$ in Eq. (3). After obtaining the clustering partition matrix $\mathbf{H}$, a standard k-means algorithm is applied to compute the discrete cluster assignments.

2.2 SimpleMKKM: Simple Multiple Kernel K-means

Recently, it is empirically observed in (Liu et al., 2020) that the widely used $\min_{\boldsymbol{\gamma}} \min_{\mathbf{H}}$ paradigm by existing MKKM may not be able to achieve promising clustering performance in practical applications, sometimes or even worse than the averaged kernel k-means. This inspires researchers to design new clustering models. Different from the widely used $\min_{\boldsymbol{\gamma}} \min_{\mathbf{H}}$ learning paradigm of the existing MKKM (Yu et al., 2012), SimpleMKKM proposes a novel $\min_{\boldsymbol{\gamma}} \max_{\mathbf{H}}$ optimization framework as follows,

$$\min_{\boldsymbol{\gamma}\in\Delta} \ \max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \ \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top) \ \ s.t. \ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k. \tag{4}$$

In this formulation, $\Delta = \{\boldsymbol{\gamma}\in\mathbb{R}^m | \sum_{p=1}^m \gamma_p = 1, \gamma_p \geq 0, \forall p\}$, $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^m \gamma_p^2 \mathbf{K}_p$, $\{\mathbf{K}_p\}_{p=1}^m$ is a group of pre-specified kernel matrices, $\mathbf{H}$ is termed clustering partition matrix and $\mathbf{I}_k$ is an identity matrix with size k.

This new minimization-maximization formulation makes Eq. (4) is hard to solve by the widely used alternate optimization. Differently, SimpleMKKM firstly rewrites the $\min_{\boldsymbol{\gamma}} \max_{\mathbf{H}}$ into a minimization problem w.r.t $\boldsymbol{\gamma}$, and proves the differentiability of the resultant minimization. Specifically, the formulation in Eq. (4) can be equivalently rewritten as,

$$\min_{\boldsymbol{\gamma}\in\Delta} \ \mathcal{J}(\boldsymbol{\gamma}), \tag{5}$$

with

$$\mathcal{J}(\boldsymbol{\gamma}) = \left\{ \max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\right), \ \ s.t. \ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k \right\}. \tag{6}$$

By this way, the $\min_{\boldsymbol{\gamma}}$-$\max_{\mathbf{H}}$ optimization is transformed to a minimization one, where its objective $\mathcal{J}(\boldsymbol{\gamma})$ is a kernel k-means optimal value function.

After proving the differentiability of $\mathcal{J}(\boldsymbol{\gamma})$, the authors in (Liu et al., 2020) show how to calculate its gradient, and use the reduced gradient descent algorithm to decrease Eq. (5). The optimization procedure of Eq. (4) is outlined in Algorithm 1. The ablation study (Liu et al., 2020) on various benchmark datasets validates that the novel $\min_{\boldsymbol{\gamma}}$-$\max_{\mathbf{H}}$ optimization and new optimization attribute to the improved clustering performance. (Liu et al., 2020) for the detail.

3. Parameter-free Localized SimpleMKKM

3.1 The Proposed Formulation

Let $\mathbf{h}_i \, (1 \leq i \leq n)$ denote the i-th row of the clustering partition matrix $\mathbf{H}$. As seen from Eq. (4), SimpleMKKM optimizes the alignment between $\mathbf{K}_{\boldsymbol{\gamma}}$ and $\mathbf{H}\mathbf{H}^\top$ in a *global* way. That is, it indiscriminately aligns each K_{ij} with an "ideal" value $\mathbf{h}_i^\top\mathbf{h}_j$, regardless of the potential variation among kernel matrices. This would cause K_{ij}s with high variation to be aligned with a same cluster label. A more reasonable criterion shall get rid of the less reliable farther global similarity information in a high dimensional kernel space and in the mean time concentrate more on consolidating the high confidence clustering predictions.

Algorithm 1 SimpleMKKM (Liu et al., 2020)

Input: $\{\mathbf{K}_p\}_{p=1}^m$, k, $t=1$.
Initialize $\boldsymbol{\gamma}^{(1)}=\mathbf{1}/m$, flag $=1$.
while flag **do**
  compute $\mathbf{H}$ by solving a kernel k-means with $\mathbf{K}_{\boldsymbol{\gamma}}$.
  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 $\max|\boldsymbol{\gamma}^{(t+1)}-\boldsymbol{\gamma}^{(t)}| \leq 1e-4$ **then**
    flag=0.
  end if
  $t \leftarrow t+1$.
end while

To fulfill this goal, we propose to align $\mathbf{K}_{\boldsymbol{\gamma}}$ with $\mathbf{H}\mathbf{H}^\top$ in a local way. Let $\mathbf{S}^{(i)} \in \{0,1\}^{n\times \mathrm{round}(\tau\times n)}$ $(\forall i)$ be a matrix indicating the round$(\tau\times n)$-nearest neighbors of the i-th sample, where τ is the proportion of localization and round$(\cdot)$ is a rounding function. We define a local alignment for the i-th sample as follows,

$$\left\langle {\mathbf{S}^{(i)}}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{S}^{(i)}, {\mathbf{S}^{(i)}}^\top \mathbf{H}^\top \mathbf{H} \mathbf{S}^{(i)} \right\rangle_{\mathrm{F}}, \tag{7}$$

where ${\mathbf{S}^{(i)}}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{S}^{(i)}$ denotes taking elements from $\mathbf{K}_{\boldsymbol{\gamma}}$ according to the neighborhood of the i-th sample. As seen, this local alignment only requires that closer samples shall stay together, which makes it better utilize the variation among kernels for clustering. By bringing the local alignment in Eq. (7) to each sample, we get the objective function of the localized SimpleMKKM as follows:

$$\min_{\boldsymbol{\gamma}\in\Delta} \max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \ \mathrm{Tr}\left(\mathbf{H}^\top \left(\sum\nolimits_{i=1}^n \mathbf{A}^{(i)} \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{A}^{(i)}\right) \mathbf{H}\right) \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k, \tag{8}$$

where $\Delta=\{\boldsymbol{\gamma}\in\mathbb{R}^m | \sum_{p=1}^m \gamma_p=1,\ \gamma_p\geq 0,\ \forall p\}$, $\mathbf{K}_{\boldsymbol{\gamma}}=\sum_{p=1}^m \gamma_p^2 \mathbf{K}_p$ and $\mathbf{A}^{(i)}=\mathbf{S}^{(i)}{\mathbf{S}^{(i)}}^\top$ is the neighborhood mask matrix of the i-th sample.

Theorem 1 *The objection of the proposed localized SimpleMKKM in Eq. (8) can be rewritten as follows.*

$$\min_{\boldsymbol{\gamma}\in\Delta} \max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \ \mathrm{Tr}\left(\mathbf{H}^\top \left(\mathbf{M}\otimes \mathbf{K}_{\boldsymbol{\gamma}}\right) \mathbf{H}\right) \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k, \tag{9}$$

where $\mathbf{M}=\sum_{i=1}^n \mathbf{A}^{(i)}$ *is termed the neighborhood mask matrix.*

Proof The objective function in Eq. (8) can be written as

$$\begin{aligned}
&\mathrm{Tr}\left(\mathbf{H}^\top\left(\sum_{i=1}^n \mathbf{A}^{(i)}\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{A}^{(i)}\right)\mathbf{H}\right)\\
&=\sum_{i=1}^n \mathrm{Tr}\left(\mathbf{H}^\top\left(\mathbf{A}^{(i)}\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{A}^{(i)}\right)\mathbf{H}\right)\\
&=\sum_{i=1}^n \left\langle \mathbf{A}^{(i)}\otimes\mathbf{K}_{\boldsymbol{\gamma}}, \mathbf{A}^{(i)}\otimes\left(\mathbf{H}\mathbf{H}^\top\right)\right\rangle_{\mathrm{F}}\\
&=\sum_{i=1}^n \left\langle \mathbf{A}^{(i)}\otimes\mathbf{K}_{\boldsymbol{\gamma}}, \mathbf{H}\mathbf{H}^\top\right\rangle_{\mathrm{F}}\\
&=\left\langle \left(\sum_{i=1}^n \mathbf{A}^{(i)}\right)\otimes\mathbf{K}_{\boldsymbol{\gamma}}, \mathbf{H}\mathbf{H}^\top\right\rangle_{\mathrm{F}}\\
&=\left\langle \left(\mathbf{M}\otimes\mathbf{K}_{\boldsymbol{\gamma}}, \mathbf{H}\mathbf{H}^\top\right\rangle_{\mathrm{F}}\\
&=\mathrm{Tr}\left(\mathbf{H}^\top\left(\mathbf{M}\otimes\mathbf{K}_{\boldsymbol{\gamma}}\right)\mathbf{H}\right),
\end{aligned} \tag{10}$$

where $\otimes$ denotes element-wise multiplication between two matrices. This completes the proof. ■

Theorem 1 builds the connection between SimpleMKKM and its local variant, and uncovers that one can encode the localization by element-wise multiplying each $\mathbf{K}_p$ with $\mathbf{M}$. On one hand, the local alignment in Eq. (9) can sufficiently consider the variation among base kernels, which could help to improve the performance. On the other hand, there is an extra hyper-parameter τ controlling the size of each sample's neighborhood, which is required to be pre-specified. However, it is well recognized in literature that how to choose a suitable hyper-parameter in practical clustering tasks itself is a tough task. It could be better to let clustering algorithms automatically learn the hyper-parameters. To do so, we parameterize the optimal neighborhood mask matrix $\mathbf{M}_{\boldsymbol{\mu}}$ as a weighted combination of a group of pre-specified neighborhood mask matrices $\{\mathbf{M}_p\}_{p=1}^l$, i.e., $\mathbf{M}_{\boldsymbol{\mu}}=\sum_{p=1}^l \mu_p^2\mathbf{M}_p$. As a result, choosing a suitable $\mathbf{M}$ reduces to learning an optimal combination weight $\boldsymbol{\mu}$.

By substituting $\mathbf{M}$ in Eq. (9) with $\mathbf{M}_{\boldsymbol{\mu}}$, we obtain the objective of the proposed parameter-free localized SimpleMKKM as follows,

$$\min_{\boldsymbol{\gamma}\in\boldsymbol{\Delta}}\ \min_{\boldsymbol{\mu}\in\boldsymbol{\Theta}}\ \max_{\mathbf{H}\in\mathbb{R}^{n\times k}}\ \ \mathrm{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}}\otimes\mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right)\ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k, \tag{11}$$

where $\mathbf{M}_{\boldsymbol{\mu}}=\sum_{p=1}^l \mu_p^2\mathbf{M}_p$ and $\boldsymbol{\Theta}=\{\boldsymbol{\mu}\in\mathbb{R}^l|\boldsymbol{\mu}^\top\mathbf{e}_l=1,\ \mu_p\geq 0,\ \forall p\}$.

The objective in Eq. (11) has the following merits: i) It calculates the kernel alignment in a local manner, 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. (11) is much more difficult to optimize than SimpleMKKM. In the following, we develop a reduced gradient descent algorithm to optimize it.

3.2 The Calculation of Reduced Gradient and Optimization Algorithm

To solve the optimization in Eq. (11), we first rewrite it as an optimal value function, prove its differentiability, and calculate its reduced gradient. After that, we update the

optimization variables with gradient descent. Specifically, we firstly rewrite Eq. (11) as follows,

$$\min_{\boldsymbol{\gamma}\in\boldsymbol{\Delta}} \ \mathcal{T}(\boldsymbol{\gamma}) \tag{12}$$

with

$$\mathcal{T}(\boldsymbol{\gamma}) = \left\{\min_{\boldsymbol{\mu}\in\boldsymbol{\Theta}} \max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \ \mathrm{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}}\otimes\mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right) \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k\right\}. \tag{13}$$

We firstly prove the differentiability of $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (12). To achieve this goal, we have the following Lemma 2.

Lemma 2 *$\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (6) is convex w.r.t $\boldsymbol{\gamma}$.*

Proof For any $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2 \in \Delta$ and $0<\alpha<1$, we have

$$\begin{aligned}
&\mathcal{J}(\alpha\boldsymbol{\gamma}_1+(1-\alpha)\boldsymbol{\gamma}_2)\\
&=\left\{\max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{K}_{\alpha\boldsymbol{\gamma}_1+(1-\alpha)\boldsymbol{\gamma}_2}\mathbf{H}\right), \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k\right\}\\
&=\left\{\max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top\left(\sum\nolimits_{p=1}^{m}\left(\alpha\gamma_{1p}+(1-\alpha)\gamma_{2p}\right)^2\mathbf{K}_p\right)\mathbf{H}\right), \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k\right\}\\
&\le\left\{\max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top\left(\sum\nolimits_{p=1}^{m}\left(\alpha\gamma_{1p}^2+(1-\alpha)\gamma_{2p}^2\right)\mathbf{K}_p\right)\mathbf{H}\right), \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k\right\}\\
&\le\left\{\alpha\max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top\left(\sum\nolimits_{p=1}^{m}\gamma_{1p}^2\mathbf{K}_p\right)\mathbf{H}\right)+(1-\alpha)\max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top\left(\sum\nolimits_{p=1}^{m}\gamma_{2p}^2\mathbf{K}_p\right)\mathbf{H}\right), \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k\right\}\\
&=\alpha\mathcal{J}(\boldsymbol{\gamma}_1)+(1-\alpha)\mathcal{J}(\boldsymbol{\gamma}_2).
\end{aligned} \tag{14}$$

This completes the proof. ■

Based on Lemma 2, we conclude that the solution optimized by Algorithm 1 is the global optimum. Given $\boldsymbol{\gamma}$, the optimization in Eq. (13) is equivalent to Eq. (4), which can be readily solved by Algorithm 1, generating the global optimum. According to Lemma 2, we have the following Theorem 3.

Theorem 3 *$\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (12) is differentiable w.r.t $\boldsymbol{\gamma}$. Further, $\frac{\partial\mathcal{T}(\boldsymbol{\gamma})}{\partial\gamma_p}=2\gamma_p\mathrm{Tr}\left(\mathbf{H}^{*\top}\left(\mathbf{M}_{\boldsymbol{\mu}^*}\otimes\mathbf{K}_p\right)\mathbf{H}^*\right)$, where $(\mathbf{H}^*,\boldsymbol{\mu}^*)=\left\{\arg\min_{\boldsymbol{\mu}\in\boldsymbol{\Theta}}\max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \ \mathrm{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}}\otimes\mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right) \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k\right\}$.*

Proof Based on Lemma 2, we conclude that the solution of optimal value function in Eq. (13) is unique with a given $\boldsymbol{\gamma}$. According to Theorem 4.1 in (Bonnans and Shapiro, 1998), $\mathcal{T}(\boldsymbol{\gamma})$ in Eq. (12) is differentiable w.r.t $\boldsymbol{\gamma}$. Further, $\frac{\partial\mathcal{T}(\boldsymbol{\gamma})}{\partial\gamma_p}=2\gamma_p\mathrm{Tr}\left(\mathbf{H}^{*\top}\left(\mathbf{M}_{\boldsymbol{\mu}^*}\otimes\mathbf{K}_p\right)\mathbf{H}^*\right)$, where $(\mathbf{H}^*,\boldsymbol{\mu}^*)=\left\{\arg\min_{\boldsymbol{\mu}\in\boldsymbol{\Theta}}\max_{\mathbf{H}\in\mathbb{R}^{n\times k}} \ \mathrm{Tr}\left(\mathbf{H}^\top(\mathbf{M}_{\boldsymbol{\mu}}\otimes\mathbf{K}_{\boldsymbol{\gamma}})\mathbf{H}\right) \ \ s.t.\ \ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k\right\}$. ■

In the following, we propose to solve the optimization in Eq. (12) with a reduced gradient descent algorithm. We firstly calculate the gradient of $\mathcal{T}(\boldsymbol{\gamma})$ according to Theorem 3, and then update $\boldsymbol{\gamma}$ with a descent direction by which the equality and non-negativity constraints

on $\boldsymbol{\gamma}$ can be guaranteed. To fulfill this goal, we firstly handle the equality constraint by computing the reduced gradient by following (Liu et al., 2020; Rakotomamonjy et al., 2008). Let γ_u be a non-zero component of $\boldsymbol{\gamma}$ and $\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})$ denote the reduced gradient of $\mathcal{T}(\boldsymbol{\gamma})$. The p-th ($1 \leq p \leq m$) element of $\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})$ is

$$[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_p = \frac{\partial\mathcal{T}(\boldsymbol{\gamma})}{\partial\gamma_p} - \frac{\partial\mathcal{T}(\boldsymbol{\gamma})}{\partial\gamma_u} \ \ \forall\, p \neq u, \tag{15}$$

and

$$[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_u = \sum\nolimits_{p=1,p\neq u}^{m} \left(\frac{\partial\mathcal{T}(\boldsymbol{\gamma})}{\partial\gamma_u} - \frac{\partial\mathcal{T}(\boldsymbol{\gamma})}{\partial\gamma_p}\right). \tag{16}$$

Following the suggestion in (Liu et al., 2020; Rakotomamonjy et al., 2008), we choose u to be the index of the largest component of vector $\boldsymbol{\gamma}$ which is considered to provide better numerical stability.

We then take the positivity constraints on $\boldsymbol{\gamma}$ into consideration in the descent direction. Note that -$\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})$ is a descent direction since our aim is to minimize $\mathcal{T}(\boldsymbol{\gamma})$. However, directly using this direction would violate the positivity constraints in the case that there is an index p such that $\gamma_p = 0$ and $[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_p > 0$. In such a case, the descent direction for that component should be set to 0. This gives the descent direction for updating $\boldsymbol{\gamma}$ as

$$d_p = \begin{cases} 0 & \text{if } \gamma_p = 0 \text{ and } [\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_p > 0 \\ -[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_p & \text{if } \gamma_p > 0 \text{ and } p \neq u \\ -[\bigtriangledown\mathcal{T}(\boldsymbol{\gamma})]_u & \text{if } p = u. \end{cases} \tag{17}$$

After a descent direction $\mathbf{d} = [d_1, \cdots, d_m]^\top$ is computed by Eq. (17), $\boldsymbol{\gamma}$ can be calculated via the updating scheme $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} + \alpha\mathbf{d}$, where α is the optimal step size. It can be selected by a one-dimensional line search strategy such as Armijo's rule. The whole algorithm procedure solving the optimization problem in Eq. (11) is outlined in Algorithm 2.

Algorithm 2 Parameter-free Localized SimpleMKKM

Input: $\{\mathbf{K}_p\}_{p=1}^m$, $\{\mathbf{M}_p\}_{p=1}^l$ and ϵ_0.
Output: $\mathbf{H}$ and $\boldsymbol{\gamma}$, $\boldsymbol{\mu}$.
Initialize $\boldsymbol{\gamma}^{(0)} = \mathbf{e}_m/m$, $\boldsymbol{\mu}^{(0)} = \mathbf{e}_l/l$ and $t = 1$.
repeat
  $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}} = \sum_{p=1}^m \left(\gamma_p^{(t-1)}\right)^2 \mathbf{K}_p$.
  compute $\mathbf{H}$ and $\boldsymbol{\mu}$ by SimpleMKKM in Algorithm 1 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. (17).
  update $\boldsymbol{\gamma}^{(t+1)} \leftarrow \boldsymbol{\gamma}^{(t)} + \alpha\mathbf{d}^{(t)}$.
  if $\max|\boldsymbol{\gamma}^{(t+1)} - \boldsymbol{\gamma}^{(t)}| \leq 1e-4$ **then**
    flag=0.
  end if
  $t \leftarrow t + 1$.
end while

Note that with given $\boldsymbol{\gamma}$, Eq. (13) has the global optimum. Under this condition, the gradient computation in Theorem 3 is exact, and our algorithm performs reduced

gradient descent on a continuously differentiable function $\mathcal{T}(\boldsymbol{\gamma})$ defined on the simplex $\{\boldsymbol{\gamma} \in \mathbb{R}^m | \sum_{p=1}^{m} \gamma_p = 1, \gamma_p \geq 0, \forall p\}$, which does converge to the minimum of $\mathcal{T}(\boldsymbol{\gamma})$ (Rakotomamonjy et al., 2008), as validated by the experiments in Figure 4.

4. Experiments

4.1 Experimental Settings

A number of MKKM benchmark datasets are adopted to conduct the comprehensive experiment, providing a good testbed to evaluate the performance of parameter-free localized SimpleMKKM. They include *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. 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 number of samples, kernels and categories of these datasets show considerable variation, providing a good platform to compare the performance of different clustering algorithms. We generate a group of base neighborhood mask matrices $\{\mathbf{M}_p\}_{p=1}^{l}$ according to the definition in Eq. (9). Since the neighbor number is defined by 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.

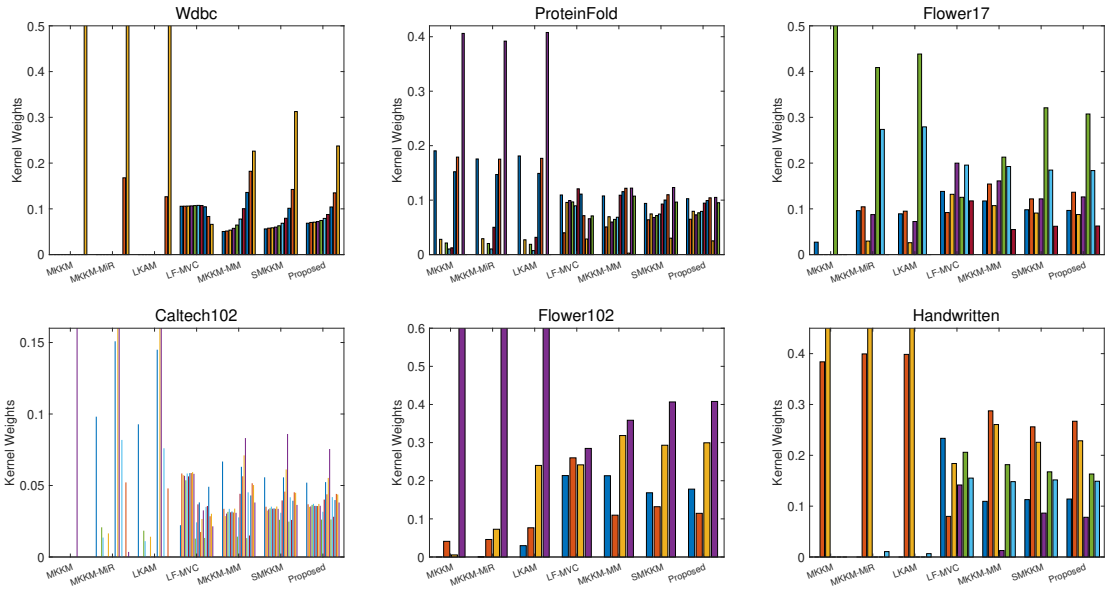
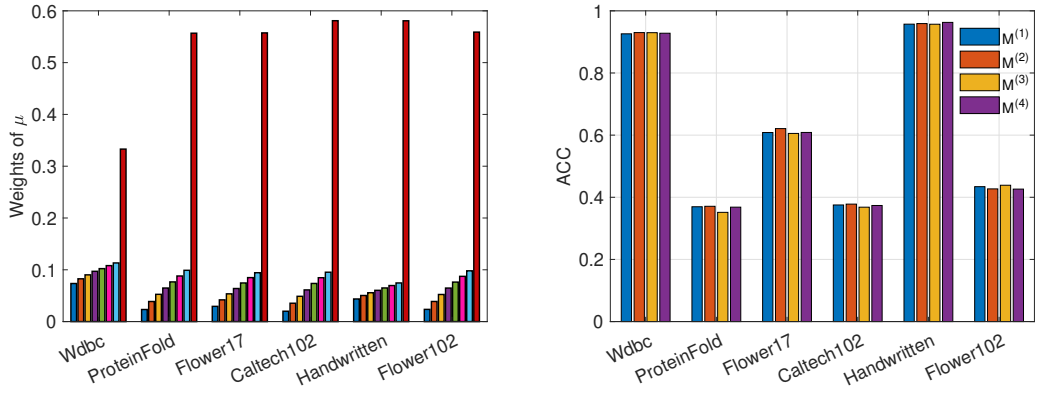


Figure 1: The kernel weights learned by different algorithms.

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/

Dataset	Avg-MKKM	MKKM	LMKKM	MKKM-MiR	LKAM	LF-MVC	MKKM-MM	SMKKM	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	**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	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	**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	**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	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	**42.7 ± 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	**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	**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	**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	**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	**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	**59.4 ± 0.3**
				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	**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	**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	**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	**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	**96.1 ± 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	**49.6 ± 0.7**
				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	**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	**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	**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	**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	**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	**28.8 ± 0.9**

Table 1: Empirical comparison of the proposed algorithm with baseline methods on eight datasets in terms of ACC, NMI, Pur and RI. Boldface results indicate no statistical difference from the best one.



(a) (b)

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

For all benchmark datasets in the experiment, the cluster number k is given and taken as the input of algorithms. Four common clustering evaluation criteria, i.e., clustering accuracy (ACC), normalized mutual information (NMI), purity and rand index (RI) are adopted for algorithm validation. To alleviate the interference of randomness caused by k-means, in the experiment, we repeat the testing procedure with random initialization for 50 times. Both the mean value and the variation of the 50 trials are reported.

For evaluating the effectiveness of the proposed algorithm, eight state-of-the-art multiple kernel clustering algorithms are included for comparison.

- **Average kernel (Avg-KKM)**. A consensus kernel is firstly constructed by linearly combining the base kernels with equal weights and then taken as the input of kernel .

- **Multiple kernel (MKKM)** (Huang et al., 2012b). The linear combination weights and the cluster indicating matrix are optimized simultaneously in a unified optimization framework.

- **Localized multiple kernel (LMKKM)** (Gönen and Margolin, 2014). A sample-adaptive base kernel combination mechanism is introduced to enhance the performance of MKKM.

- **Multiple kernel with matrix-induced regularization (MKKM-MiR)** (Liu et al., 2016). A matrix-induced regularization term is integrated to the MKKM learning to introduce diverse information preservation.

- **Multiple kernel clustering with local alignment maximization (LKAM)** (Li et al., 2016). It learns the 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)** (Wang et al., 2019). It proposes to first compute the base partitions within corresponding data views and then integrated them into a united partition matrix.

- **MKKM-MM** (Bang et al., 2018). It proposes a $\min_{\mathbf{H}}$-$\max_{\boldsymbol{\gamma}}$ formulation that combines different data views in the way indicating high within-cluster variance in the consensus kernel space and then optimize the clusters through minimizing such variance.

- **SimpleMKKM (SMKKM)** (Liu et al., 2020). It extends the widely applied supervised alignment criterion to multi-kernel clustering and proposes a special min-max clustering objective for kernel weights and cluster partition optimization.

The official implementations of the aforementioned algorithms are publicly available. Among the compared algorithms, LKAM (Li et al., 2016), MKKM-MiR (Liu et al., 2016) and LF-MVC (Wang et al., 2019) are not parameter-free. Following the recommended settings in the corresponding papers, we run the released codes and tune the hyper-parameters carefully. The best clustering results of these methods are reported.

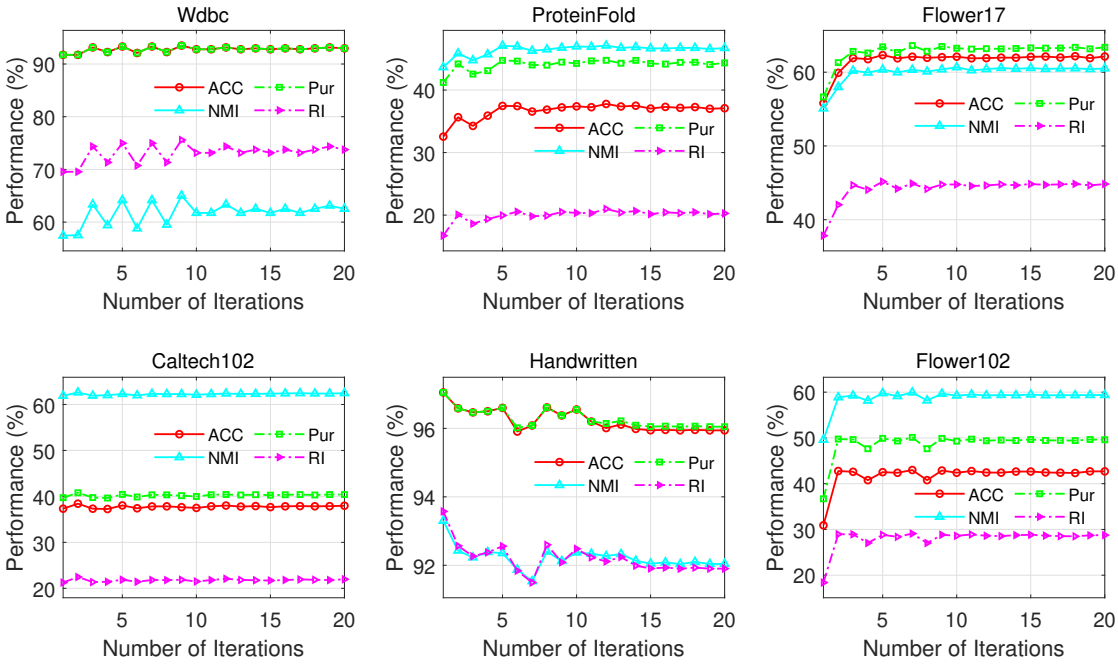


Figure 3: The evolution of the learned $\mathbf{H}$ by the proposed algorithm with iterations.

4.2 Experimental Results

4.2.1 Overall Clustering Performance Comparison

Table 1 shows the ACC, NMI, purity and RI of all the aforementioned algorithms. From Table 1, we have the following observations:

- The proposed parameter-free localized SimpleMKKM significantly outperforms the algorithms with hyper-parameters, like LF-MVC (Wang et al., 2019). This demonstrates the practicability and efficacy of our formulation.
- SimpleMKKM (Liu et al., 2020), which adopts the similar min-max optimization formulation with our proposed algorithm, achieves comparable or better clustering performance then the algorithms with hyper-parameters on most benchmark datasets. This superiority can be attributed to its novel formulation and optimization algorithm.
- The proposed parameter free localized SimpleMKKM consistently and significantly outperforms all compared algorithms. For example, it exceeds SimpleMKKM algorithm by $8.2\%, 2.3\%, 2.7\%, 1.9\%, 4.6\%, 0.8\%$ and exceeds LF-MVC algorithm by $7.3\%, 5.0\%, 1.6\%, 2.8\%, 1.1\%, 4.5\%$ in terms of NMI on six benchmark dataset, respectively. The improvements in terms of other criteria are similar. These results well illustrate the superiority of the proposed parameter free localized SimpleMKKM that benefits from adaptively extracting the localized information of kernel matrix.
- The proposed parameter free localized SimpleMKKM performs better than MKKM-MiR (Liu et al., 2016), LKAM (Li et al., 2016) and LF-MVC (Wang et al., 2019), all of

which have several hyper-parameters to tune attributed to regularization on the kernel weights. Thus they need to take huge effort to choose the best hyper-parameters in practice. And parameter tuning is very difficult or even impossible in real applications where there is no ground truth. In contrast, the proposed algorithm is parameter-free.

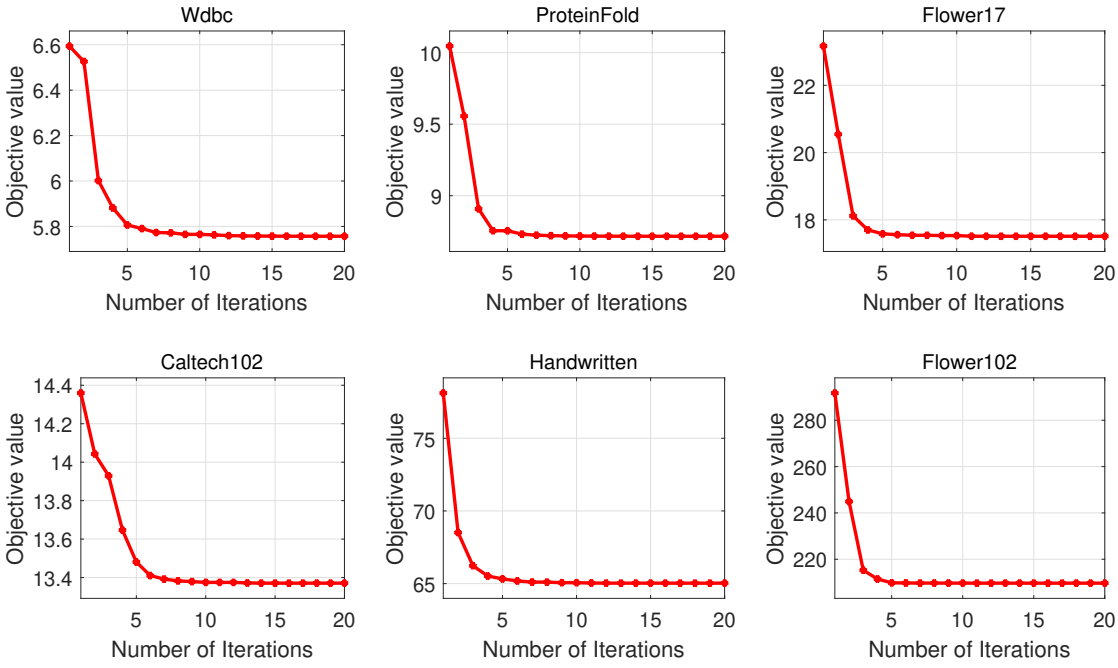


Figure 4: The objective of the proposed parameter free localized SimpleMKKM evolves with iterations.

Besides inheriting the advanced formulation and new optimization from SimpleMKKM, the proposed algorithm adaptively learns a local manner to calculate the kernel alignment, which enables it to well handle the variation among kernels. These factors jointly lead to its significant improvement over the alternatives on all datasets. In addition, we point out that LMKKM (Gönen and Margolin, 2014) cannot get the results reported on some datasets due to the out-of-memory error, which are caused by its cubic computational and memory complexity.

4.2.2 Kernel Weight Analysis

We further look into the kernel weights learned by all aforementioned algorithms on all datasets. The results are plotted in Figure 1. As seen, the kernel weights learned by MKKM, MKKM-MiR and LKAM are distributed very unevenly and are highly sparse on almost all datasets. This sparsity would make the multiple kernel matrices insufficiently exploited, leading to poor performance. For instance, the ACC of MKKM, MKKM-MiR and LKAM on Flower17 is only 44.9% , 58.5% and 50.0%, respectively. In contrast, despite the ℓ_1-norm constraint on $\boldsymbol{\gamma}$, the kernel weights learned by the proposed parameter free

localized SimpleMKKM are non-sparse on all datasets, which contributes to its superior clustering performance. This non-sparsity of the learned kernel weights is attributed to our new reduced gradient descent algorithm, which in turn is derived based on our new min-max kernel alignment objective.

4.2.3 Mask Matrix Weight Analysis

We also investigate the mask matrix weights learned by the proposed algorithm on all datasets, and the results are plotted in sub-figure 2(a). As seen, 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 $\{\mathbf{M}_p^{(q)}\}_{p=1}^{l}$ $(1 \leq q \leq 4)$, and the results are plotted in sub-figure 2(b). As seen, the performance of the proposed algorithm is almost the same under different groups of $\{\mathbf{M}_p\}_{p=1}^{l}$. We believe that its performance can be further improved by incorporating prior knowledge to constructing base mask matrices, which is worth further exploring.

4.2.4 Convergence and Evolution of the Learned $\mathbf{H}$

As proved in Section 3.2, our parameter-free localized SimpleMKKM is theoretically guaranteed to converge. To see this point in depth, we further plot the objective of parameter-free localized SimpleMKKM with iterations on all datasets, as shown in Figure 4. We observe that its objective is monotonically decreased and usually achieve convergence in fewer than ten iterations on all datasets. Also, to reveal the clustering performance variation of the learned $\mathbf{H}$ with the number of iterations, we calculate ACC, NMI, purity and RI at each iteration, and report them in Figure 3. As observed, the clustering performance of our algorithm firstly increases with the number of iterations, slightly oscillates and then remains stable. The result also reveals the effectiveness and necessity of the learning procedure.

4.2.5 Running Time Comparison

Finally, we report the execution time of the all algorithms in experiment, as plotted in Figure 5. We observe that besides greatly improving the clustering performance, the proposed parameter-free localized SimpleMKKM also has the comparable time cost with other counterparts.

5. Conclusion

While the recently proposed SimpleMKKM demonstrates promising clustering performance, it does not think over the variation among base kernel matrices sufficiently. This paper proposes to optimize the kernel alignment in a parameter-free localized manner to address this issue. We firstly uncover the theoretical connection between SimpleMKKM and the proposed algorithm. Based on this observation, we parameterize the neighborhood mask matrix as a quadratic combination of a group of pre-specified base neighborhood mask matrices, and jointly learn the optimal combination coefficient together with clustering tasks, leading to an intractable tri-level optimization problem. We then build an efficient and elegant algorithm with guaranteed convergence to solve it. The proposed parameter-free lo-

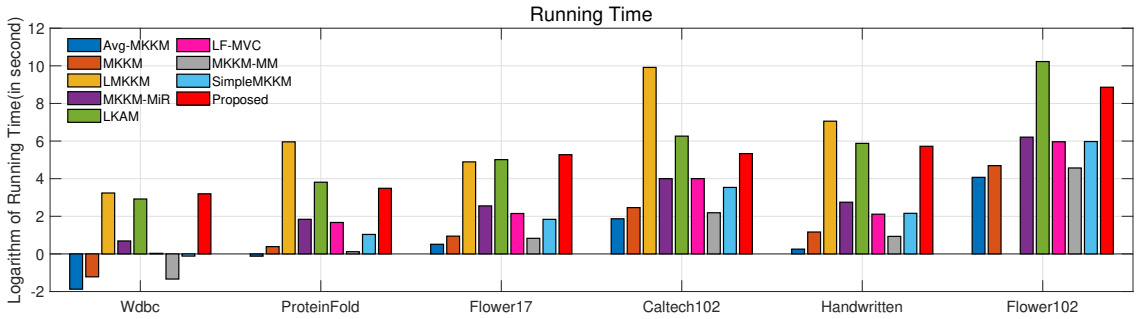


Figure 5: Running time comparison of different algorithms on all datasets (logarithm in seconds). The experiments are carried out on a PC with Intel(R) Core(TM)-i9-10900X 3.7GHz CPU and 64G RAM in MATLAB R2020b environment.

calized SimpleMKKM demonstrates significantly increased clustering results via substantial experiments on multiple benchmark datasets.

Acknowledgments

This work was supported by the National Key R&D Program of China 2020AAA0107100, the Natural Science Foundation of China (Project No. 61922088, 61773392, 61872377 and 61976196).

References

Seojin Bang, Yaoliang Yu, and Wei Wu. Robust multiple kernel k-means clustering using min-max optimization, 2018.

J. Frédéric Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40(2):228–264, 1998.

Jianhui Chen, Zheng Zhao, Jieping Ye, and Huan Liu. Nonlinear adaptive distance metric learning for clustering. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–132. ACM, 2007.

Mehmet Gönen and Adam A Margolin. Localized data fusion for kernel k-means clustering with application to cancer biology. In *Advances in Neural Information Processing Systems*, pages 1305–1313, 2014.

Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Multiple kernel fuzzy clustering. *IEEE Trans. Fuzzy Syst.*, 20(1):120–134, 2012a.

Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1):120–134, 2012b.

Zhenyu Huang, Peng Hu, Joey Tianyi Zhou, Jiancheng Lv, and Xi Peng. Partially view-aligned clustering. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*.

Zhao Kang, Xinjia Zhao, Chong Peng, Hongyuan Zhu, Joey Tianyi Zhou, Xi Peng, Wenyu Chen, and Zenglin Xu. Partition level multiview subspace clustering. *Neural Networks*, 122:279–288, 2020.

Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400, 2011.

Miaomiao Li, Xinwang Liu, Lei Wang, Yong Dou, Jianping Yin, and En Zhu. Multiple kernel clustering with local kernel alignment maximization. In *International Joint Conference on Artificial Intelligence*, pages 1704–1710, 2016.

Weixuan Liang, Sihang Zhou, Jian Xiong, Xinwang Liu, Siwei Wang, En Zhu, Zhiping Cai, and Xin Xu. Multi-view spectral clustering with high-order optimal neighborhood laplacian matrix. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Weiwei Liu, Xiao-Bo Shen, and Ivor W. Tsang. Sparse embedded k-means clustering. In *Advances in Neural Information Processing Systems 30*, pages 3319–3327, 2017a.

Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, and En Zhu. Multiple kernel k -means clustering with matrix-induced regularization. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 1888–1894, 2016.

Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, and Jianping Yin. Optimal neighborhood kernel clustering with multiple kernels. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017b.

Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Chang Tang, En Zhu, Jianping Yin, and Wen Gao. Efficient and effective incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4392–4399, 2019.

Xinwang Liu, En Zhu, Jiyuan Liu, Timothy M. Hospedales, Yang Wang, and Meng Wang. Simplemkkm: Simple multiple kernel k-means. *CoRR*, abs/2005.04975, 2020.

Xi Peng, Zhenyu Huang, Jiancheng Lv, Hongyuan Zhu, and Joey Tianyi Zhou. COMIC: multi-view clustering without parameter selection. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, pages 5092–5101.

Alain Rakotomamonjy, Francis R. Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *JMLR*, 9:2491–2521, 2008.

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Wei Tang, Zhengdong Lu, and Inderjit S Dhillon. Clustering with multiple graphs. In *ICDM*, pages 1016–1021. IEEE, 2009.

Hua Wang, Feiping Nie, and Heng Huang. Multi-view clustering and feature learning via structured sparsity. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, volume 28, pages 352–360.

Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. Multi-view clustering via late fusion alignment maximization. In *IJCAI*, pages 3778–3784, 2019.

Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In *NIPS*, pages 1537–1544, 2004.

Shi Yu, Leon Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan AK Suykens, Bart De Moor, and Yves Moreau. Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039, 2012.

Changqing Zhang, Huazhu Fu, Si Liu, Guangcan Liu, and Xiaochun Cao. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 1582–1590, 2015.

Changqing Zhang, Huazhu Fu, Jing Wang, Wen Li, Xiaochun Cao, and Qinghua Hu. Tensorized multi-view subspace representation learning. *International Journal of Computer Vision*, pages 1–18, 2020.

Sihang Zhou, En Zhu, Xinwang Liu, Tianming Zheng, Qiang Liu, Jingyuan Xia, and Jianping Yin. Subspace segmentation-based robust multiple kernel clustering. *Information Fusion*, 53:145–154, 2020.