Localized Simple Multiple Kernel K-means

Xinwang Liu[1]*, Sihang Zhou[1], Li Liu[1], Chang Tang[2], Siwei Wang[1], Jiyuan Liu[1] and Yi Zhang[1]
[1] National University of Defense Technology, Changsha, Hunan, 410072, China.
[2] China University of Geosciences, Wuhan, Hubei, 430074, China.
xinwangliu@nudt.edu.cn

Abstract

As a representative of multiple kernel clustering (MKC), simple multiple kernel k-means (SimpleMKKM) is recently put forward to boosting the clustering performance by optimally fusing a group of pre-specified kernel matrices. Despite achieving significant improvement in a variety of applications, we find out that SimpleMKKM could indiscriminately force all sample pairs to be equally aligned with the same ideal similarity. As a result, it does not sufficiently take the variation of samples into consideration, leading to unsatisfying clustering performance. To address these issues, this paper proposes a novel MKC algorithm with a "local" kernel alignment, which only requires that the similarity of a sample to its k-nearest neighbours be aligned with the ideal similarity matrix. Such an alignment helps the clustering algorithm to focus on closer sample pairs that shall stay together and avoids involving unreliable similarity evaluation for farther sample pairs. After that, we theoretically show that the objective of SimpleMKKM is a special case of this local kernel alignment criterion with normalizing each base kernel matrix. Based on this observation, the proposed localized SimpleMKKM can be readily implemented by existing SimpleMKKM package. Moreover, we conduct extensive experiments on several widely used benchmark datasets to evaluate the clustering performance of localized SimpleMKKM. The experimental results have demonstrated that our algorithm consistently outperforms the state-of-the-art ones, verifying the effectiveness of the proposed local kernel alignment criterion. The code of Localized SimpleMKKM is publicly available at: `https://github.com/xinwangliu/LocalizedSMKKM`.

1. Introduction

Multiple kernel clustering (MKC) provides an elegant framework to group samples into different clusters by extracting complementary information from multiple sources [25, 21, 26, 8, 5, 18, 22, 29, 10, 7, 27, 28]. Given a group of pre-defined kernel matrices, MKC integrates the available multiple kernel information to categorize data items with similar structures or patterns into the same group, which has been intensively studied and widely applied into various applications [9, 14, 9, 23, 17, 16, 13]. For example, the work in [11] proposes a multiple kernel k-means clustering algorithm with a matrix-induced regularization term to reduce the redundancy of the selected kernels. A local kernel alignment variant is then developed by sufficiently considering the variation among sample, which is experimentally verified to enhance the clustering performance in [9]. By assuming an optimal kernel residing in the neighborhood of the combined kernels, the work in [14] 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 [23]. Specifically, the pioneering work in [23] 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 [12] to handle incomplete multi-view data.

As a representative of MKC, a novel simple multiple kernel k-means (SimpleMKKM) is recently proposed [15]. 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 algorithm is designed to solve the resultant optimization. This algorithm is validated to be efficient for optimization, robust against the noisy views, and has attracted intensive attention of many researchers.

Although the recently proposed SimleMKKM bears the aforementioned merits, we observe that it strictly aligns the combined kernel matrix with an "ideal" similarity gener-

*Corresponding Author

ated by the clustering partition matrix in a *global* way. This could indiscriminately force all sample pairs to be equally aligned with the same ideal similarity. As a result, it does not effectively handle the variation among samples and ignore local structures, which could lead to unsatisfying clustering performance. To address the above issue, 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 an alignment guides the clustering algorithm to focus on closer sample pairs that shall stay together and avoid involving unreliable similarity evaluation for farther sample pairs. By this way, our proposed algorithm could sufficiently consider the variation among samples, leading to improved clustering performance. We then derive the objective function of our algorithm based on the minimization-maximization optimization framework of SimpleMKKM. After that, we theoretically show that SimpleMKKM is a special case of our proposed algorithm. Base on this observation, our proposed local variant can be readily implemented by SimpleMKKM packages via simply normalizing each base kernel. Comprehensive experiments have been conducted on several benchmark datasets, and the results have well validated the effectiveness of the proposed localized SimpleMKKM. The main contributions of this paper are summarized as follows,

- We, for the first time, identify that the recently proposed SimpleMKKM cannot effectively handle the variation among kernel matrices, and develop a local kernel alignment criterion to address this issue.
- We theoretically reveal the connection between SimpleMKKM and our proposed algorithm, and point out that the former is a special case of ours.
- Extensive experiments are conducted on several public datasets to evaluate the effectiveness of our proposed algorithm. As indicated, the experimental results have demonstrated that our algorithm consistently outperforms the state-of-the-art competitors, verifying its effectiveness and efficiency.

2. Related work

In this section, we briefly introduce multiple kernel k-means (MKKM) [3] and the recently proposed simple multiple kernel k-means (SimpleMKKM) [15], 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

$$\min_{\mathbf{Z},\{\mathbf{c}_q\}_{q=1}^k} \frac{1}{n}\sum_{i=1}^{n}\sum_{q=1}^{k} Z_{iq}\|\mathbf{x}_i - \mathbf{c}_q\|^2 \quad (1)$$

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) [20] $\mathcal{H}$ 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

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

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,

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

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. Simple Multiple Kernel K-means

Recently, it is empirically observed in [15] 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 existing $\min_{\boldsymbol{\gamma}}\min_{\mathbf{H}}$ paradigm, SimpleMKKM proposes a novel $\min_{\boldsymbol{\gamma}}\max_{\mathbf{H}}$ optimization framework as follows,

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

This new minimization-maximization formulation makes Eq. (4) cannot be solved by the widely used alternate optimization. Differently, SimpleMKKM firstly rewrites the $\min_{\boldsymbol{\gamma}}\max_{\mathbf{H}}$ into a minimization w.r.t $\boldsymbol{\gamma}$, and proves the differentiability of the resultant minimization. After that, a reduced gradient descent optimization is designed to solve the minimization w.r.t $\boldsymbol{\gamma}$.

3. The Proposed 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 a "ideal" value $\mathbf{h}_i^\top \mathbf{h}_j$, no regardless of the potential variation among kernel matrices. This would cause K_{ij}'s with high variation to be aligned with a same cluster labels. A more reasonable criterion shall get rid of the less reliable farther global similarity information in high dimensional kernel space and in the mean time concentrate more on consolidating the high confidence clustering predictions. 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 $\mathrm{round}(\tau \times n)$-nearest neighbors of the i-th sample, where $\mathrm{round}(\cdot)$ is a rounding function. We define a local alignment for the i-th sample as follows,

$$\left\langle {\mathbf{S}^{(i)}}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{S}^{(i)},\, {\mathbf{S}^{(i)}}^\top \mathbf{H}^\top \mathbf{H} \mathbf{S}^{(i)} \right\rangle_{\mathrm{F}}, \tag{5}$$

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 more reliable samples shall stay together, which makes it better utilize the variation among kernels for clustering. By taking over the local alignment in Eq. (5) for each sample, we obtain the objective function of the proposed localized SimpleMKKM as follow:

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Delta}\ \max_{\mathbf{H}\in\mathbb{R}^{n\times k}}\ & \mathrm{Tr}\left(\mathbf{H}^\top \sum\nolimits_{i=1}^{n} (\mathbf{A}^{(i)} \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{A}^{(i)}) \mathbf{H}\right) \\ & s.t.\ \ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{6}$$

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.

In the following, we build the theoretical connection between the proposed algorithm and SimpleMKKM in Theorem 1.

Theorem 1 *The objection of SimpleMKKM is a special case of Eq. (6).*

Proof 1 *The objective function in Eq. (6) can be written as*

$$\begin{aligned} & \sum\nolimits_{i=1}^{n} \mathrm{Tr}\left(\mathbf{H}^\top (\mathbf{A}^{(i)} \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{A}^{(i)}) \mathbf{H}\right) \\ &= \sum\nolimits_{i=1}^{n} \left\langle \mathbf{A}^{(i)} \otimes \mathbf{K}_{\boldsymbol{\gamma}}, \mathbf{A}^{(i)} \otimes (\mathbf{H}\mathbf{H}^\top) \right\rangle_{\mathrm{F}} \\ &= \sum\nolimits_{i=1}^{n} \left\langle \mathbf{A}^{(i)} \otimes \mathbf{K}_{\boldsymbol{\gamma}}, \mathbf{H}\mathbf{H}^\top \right\rangle_{\mathrm{F}} \\ &= \left\langle \left(\sum\nolimits_{i=1}^{n} \mathbf{A}^{(i)}\right) \otimes \mathbf{K}_{\boldsymbol{\gamma}}, \mathbf{H}\mathbf{H}^\top \right\rangle_{\mathrm{F}} \\ &= \sum\nolimits_{p=1}^{m} \gamma_p^2 \left\langle \left(\sum\nolimits_{i=1}^{n} \mathbf{A}^{(i)}\right) \otimes \mathbf{K}_p, \mathbf{H}\mathbf{H}^\top \right\rangle_{\mathrm{F}} \\ &= \sum\nolimits_{p=1}^{m} \gamma_p^2 \left\langle \tilde{\mathbf{K}}_p, \mathbf{H}\mathbf{H}^\top \right\rangle_{\mathrm{F}} \\ &= \mathrm{Tr}\left(\mathbf{H}^\top \tilde{\mathbf{K}}_{\boldsymbol{\gamma}} \mathbf{H}\right), \end{aligned} \tag{7}$$

where $\otimes$ denotes element-wise multiplication between two matrices, $\tilde{\mathbf{K}}_p = \left(\sum_{i=1}^n \mathbf{A}^{(i)}\right) \otimes \mathbf{K}_p$ can be treated as a normalized $\mathbf{K}_p$, and $\tilde{\mathbf{K}}_{\boldsymbol{\gamma}} = \sum_{p=1}^m \gamma_p^2 \tilde{\mathbf{K}}_p$. Consequently, by such normalization applied on each base kernel, we can clearly see that the global kernel alignment in [15] is a special case of the local kernel alignment criterion in Eq. (6). This completes the proof.

As can be seen from Theorem 1, our formulation in Eq. (6) reduces to SimpleMKKM when all elements of $\mathbf{A}^{(i)}$ are set as one. In that case, each sample takes the rest ones as its neighbors. This implies that SimpleMKKM can be treated as a special case of our formulation. Based on Theorem 1, our formulation in Eq. (6) can be equivalently rewritten as,

$$\min\nolimits_{\boldsymbol{\gamma}\in\Delta}\ \mathcal{J}(\boldsymbol{\gamma}), \tag{8}$$

with

$$\mathcal{J}(\boldsymbol{\gamma}) = \left\{ \max_{\mathbf{H}}\ \mathrm{Tr}\left(\mathbf{H}^\top \tilde{\mathbf{K}}_{\boldsymbol{\gamma}} \mathbf{H}\right),\ \ s.t.\ \ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k. \right\} \tag{9}$$

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.

The following Theorem 2 shows that each $\tilde{\mathbf{K}}_p$ is still kept positive semidefinite (PSD) with the aforementioned normalization.

Theorem 2 *Each $\tilde{\mathbf{K}}_p\,(1 \leq p \leq m)$ is PSD.*

Proof 2 *Note that $\mathbf{S}^{(i)} \in \{0,\,1\}^{n\times \mathrm{round}(\tau\times n)}$ and $\mathbf{A}^{(i)} = \mathbf{S}^{(i)} {\mathbf{S}^{(i)}}^\top$, which implies that $\mathbf{A}^{(i)}$ and $\sum_{i=1}^n \mathbf{A}^{(i)}$ are both PSD. Also, the element-wise multiplication between two PSD matrices is PSD. As a result, $\tilde{\mathbf{K}}_p$ is PSD.*

Based on Theorem 2, we know that each $\tilde{\mathbf{K}}_p$ keeps positive semidefinite with the aforementioned normalization, which guarantees the differentiability of $\mathcal{J}(\boldsymbol{\gamma})$. In the following, we first prove the differentiability of $\mathcal{J}(\boldsymbol{\gamma})$, show how to calculate its gradient, and use the reduced gradient descent algorithm in [15] to decrease Eq. (8).

3.2. The Calculation of Reduced Gradient and Optimization Algorithm

By following [15] and Theorem 2, Theorem 3 shows that $\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (8) is differentiable.

Theorem 3 ([15]) *$\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (8) is differentiable. Further, $\frac{\partial \mathcal{J}(\boldsymbol{\gamma})}{\partial \gamma_p} = 2\gamma_p \mathrm{Tr}\left(\mathbf{H}^{*\top} \tilde{\mathbf{K}}_p \mathbf{H}^*\right)$, where $\mathbf{H}^* = \left\{ \arg\max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top \tilde{\mathbf{K}}_{\boldsymbol{\gamma}} \mathbf{H}\right) \ s.t. \ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k \right\}$.*

The formal proof is omitted due to space limit. The core idea of this proof is to show that the global optimum for Eq. (9) with a given $\boldsymbol{\gamma}$ is unique. Interested readers are referred to [15] for the detailed proof.

In the following, we propose to solve the optimization in Eq. (8) with a reduced gradient descent algorithm. We firstly calculate the gradient of $\mathcal{J}(\boldsymbol{\gamma})$ 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 [19, 15]. Let γ_u be a non-zero component of $\boldsymbol{\gamma}$ and $\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})$ denote the reduced gradient of $\mathcal{J}(\boldsymbol{\gamma})$. The p-th $(1 \leq p \leq m)$ element of $\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})$ is

$$[\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})]_p = \frac{\partial \mathcal{J}(\boldsymbol{\gamma})}{\partial \gamma_p} - \frac{\partial \mathcal{J}(\boldsymbol{\gamma})}{\partial \gamma_u} \ \ \forall \, p \neq u, \tag{10}$$

and

$$[\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})]_u = \sum\nolimits_{p=1, p\neq u}^{m} \left(\frac{\partial \mathcal{J}(\boldsymbol{\gamma})}{\partial \gamma_u} - \frac{\partial \mathcal{J}(\boldsymbol{\gamma})}{\partial \gamma_p} \right) \tag{11}$$

Following the suggestion in [19, 15], 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 -$\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})$ is a descent direction since our aim is to minimize $\mathcal{J}(\boldsymbol{\gamma})$. However, directly using this direction would violate the positivity constraints in the case that if there is an index p such that $\gamma_p = 0$ and $[\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})]_p > 0$. In such case, the descent direction for that component should be set to 0. This gives the descent direction for updating $\boldsymbol{\gamma}$ as

$$d_p = \begin{cases} 0 & \text{if } \gamma_p = 0 \text{ and } [\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})]_p > 0 \\ -[\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})]_p & \text{if } \gamma_p > 0 \text{ and } p \neq u \\ -[\bigtriangledown \mathcal{J}(\boldsymbol{\gamma})]_u & \text{if } p = u. \end{cases} \tag{12}$$

After a descent direction $\mathbf{d} = [d_1, \cdots, d_m]^\top$ is computed by Eq. (12), $\boldsymbol{\gamma}$ 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. (6) is outlined in Algorithm 1.

Algorithm 1 The Proposed Localized SimpleMKKM

Input: $\{\mathbf{K}_p\}_{p=1}^m$, k, τ, $t = 1$.
Initialize $\boldsymbol{\gamma}^{(1)} = \mathbf{1}/m$, flag $= 1$.
Calculate the $\mathrm{round}(\tau \times n)$-nearest neighbor indicating matrices $\{\mathbf{A}^{(i)}\}_{i=1}^n$ according to the average kernel.
$\tilde{\mathbf{K}}_p = (\sum_{i=1}^n \mathbf{A}^{(i)}) \otimes \mathbf{K}$.
while flag **do**
    compute $\mathbf{H}$ by solving a kernel k-means with $\tilde{\mathbf{K}}_{\boldsymbol{\gamma}}$.
    compute $\frac{\partial \mathcal{J}(\boldsymbol{\gamma})}{\partial \gamma_p}$ $(p = 1, \cdots, m)$ and the descent direction $\mathbf{d}^{(t)}$ in Eq. (12).
    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

3.3. Computational Complexity and Convergence

We discuss the computational complexity of the proposed localized SimpleMKKM. From Algorithm 1, localized SimpleMKKM firstly calculates a neighborhood mask matrix with computational complexity $\mathcal{O}(n^2 \log_2 n)$, and then performs SimpleMKKM. Its overall complexity is $\mathcal{O}(\ell_0 * n^3 + n^2 \log_2 n)$, where ℓ_0 is the minimum of iterations to achieve convergence. As observed, localized SimpleMKKM does not significantly increase the computational complexity of existing MKKM and SimpleMKKM algorithms, whose complexity are $\mathcal{O}(n^3)$ at each iteration.

We then briefly discuss the convergence of localized SimpleMKKM. Note that with given $\boldsymbol{\gamma}$, Eq. (9) is a traditional kernel k-means which has a 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 $\mathcal{J}(\boldsymbol{\gamma})$ 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 $\mathcal{J}(\boldsymbol{\gamma})$ [19].

4. Experiments

In this section, we conduct a comprehensive experimental study to evaluate the proposed localized SimpleMKKM in terms of overall clustering performance, the learned kernel weights, the convergence and evolution of the learned $\mathbf{H}$, the parameter sensitivity analysis, and the running time.

4.1. Experimental Settings

A number of MKKM benchmark datasets are used to evaluate the performance of localized SimpleMKKM, in-

Table 1. Dataset summary.

Dataset	Number of		
	Samples	Kernels	Clusters
MSRA	210	6	7
Still	467	3	6
Cal-7	441	6	7
PFD	694	12	27
Nonpl	2732	69	3
Flo17	1360	7	17
Flo102	8189	4	102
Reuters	18758	5	6

cluding *MSRA* [24], *Still* [6], *Cal-7*[1], *PFold*[2], *Nonpl*[3], *Flo17*[4], *Flo102*[5], *Reuters*[6]. Table 1 summarizes the dataset information in detail. It can be observed that 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.

For all datasets, the true number of clusters k is prespecified and set as the true number of classes. Clustering accuracy (ACC), normalized mutual information (NMI), purity and rand index (RI) are widely applied to evaluate the clustering performance. For all algorithms, we repeat each experiment 50 times with random initialization to reduce the effect of randomness caused by k-means, and report the means and variation.

Along with localized SimpleMKKM, we ran another nine comparative algorithms in recent multiple kernel clustering literature, including

- **Average kernel k-means (Avg-KKM)**. The consensus kernel uniformly combines the base kernels, then it is taken as the input of kernel k-means.
- **Multiple kernel k-means (MKKM)** [4]. The base kernels are linearly combined into the consensus kernel. In addition, the combination weights are optimized along with clustering.
- **Localized multiple kernel k-means(LMKKM)** [2]. The base kernels are combined with sample-adaptive weights.
- **Optimal neighborhood kernel clustering (ONKC)** [14]. The consensus kernel is chosen from the neighbor of linearly combined base kernels.
- **Multiple kernel k-meanswith matrix-induced regularization (MKKM-MR)** [11]. The optimal combination weights are learned by introducing a matrix-induced regularization term to reduce the redundancy among the base kernels.
- **Multiple kernel clustering with local alignment maximization (LKAM)** [9]. The similarity of a sample to its k-nearest neighbors, instead of all samples, is aligned with the ideal similarity matrix.
- **Multi-view clustering via late fusion alignment maximization (LF-MVC)** [23]. Base partitions are first computed within corresponding data views and then integrated into a consensus partition.
- **MKKM-MM** [1]. It proposes a $\min_{\mathbf{H}}\text{-}\max_{\boldsymbol{\gamma}}$ formulation that combines views in a way to reveal high within-cluster variance in the combined kernel space and then updates clusters by minimizing such variance.
- **SimpleMKKM** [15]. It extends the widely used supervised kernel alignment criterion to multi-view clustering, and proposed a novel clustering objective which is to minimize the alignment for the kernel weights and maximize it for the clustering partition matrix.

The implementations of the compared algorithms are available in corresponding papers publicly, and we directly adopt them without adjustment in our experiments. Among all the aforementioned algorithms, ONKC [14], MKKM-MiR [11], LKAM [9] and LF-MVC [23] have hyper-parameters to be tuned. Following the same settings in the corresponding papers, we reuse the released codes and tuned the hyper-parameters carefully to produce the best possible results on each dataset.

4.2. Experimental Results

Overall Clustering Performance Comparison. Table 2 presents the ACC, NMI, purity and RI comparison of all the above algorithms. From Table 2, we obtain the following observations:

- MKKM-MM [1] makes the first attempt to improve MKKM via the minimization-maximization learning. As observed, it does improve the MKKM, yet the performance improvement over MKKM is marginal on all datasets. Meanwhile, the proposed localized SimpleMKKM significantly outperforms MKKM-MM. This once again demonstrates the advantage of our formulation and the associated optimization strategy.
- Besides our localized SimpleMKKM, SimpleMKKM achieves comparable or better clustering performance when compared with the aforementioned algorithms on all benchmark datasets. This superiority is attributed to its novel formulation and new optimization algorithm.
- The proposed localized SimpleMKKM consistently and significantly outperforms SimpleMKKM. For example, it exceeds SimpleMKKM algorithm by 4.7%, 5.2%, 8.3%, 1.2%, 17.3%, 1.8%, 1.5% and 1.1% in terms of ACC on eight benchmark datasets.

[1] http://www.vision.caltech.edu/ImageDatasets/Caltech101

[2] mkl.ucsd.edu/dataset/protein-fold-prediction

[3] https://bmi.inf.ethz.ch/supplements/protsubloc/

[4] www.robots.ox.ac.uk/~vgg/data/flowers/17/

[5] www.robots.ox.ac.uk/~vgg/data/flowers/102/

[6] http://kdd.ics.uci.edu/databases/reuters21578/

Table 2. Empirical evaluation and comparison of localized SimpleMKKM with nine baseline methods on eight benchmark datasets in terms of ACC, NMI, Purity and Rand Index. Boldface means no statistical difference from the best one.

DATASETS	AVG-KKM	MKKM [4]	LMKKM [2]	ONKC [14]	MKKM-MR [11]	LKAM [9]	LF-MVC [23]	MKKM-MM [1]	SIMPLEMKKM [15]	PROPOSED
ACC										
MSRA	83.3 ± 0.8	81.3 ± 3.1	81.9 ± 0.7	85.4 ± 0.4	88.1 ± 0.1	89.1 ± 0.2	87.8 ± 0.4	83.3 ± 0.8	86.5 ± 0.2	**91.2 ± 1.0**
STILL	31.3 ± 0.7	31.3 ± 0.6	31.1 ± 0.5	31.8 ± 1.0	31.7 ± 1.2	33.1 ± 0.3	32.0 ± 0.7	31.3 ± 0.7	31.3 ± 0.6	**36.5 ± 0.8**
CAL-7	59.2 ± 4.9	52.2 ± 4.3	53.9 ± 1.0	69.4 ± 2.5	68.4 ± 0.3	70.4 ± 1.4	71.4 ± 1.4	59.2 ± 4.9	68.2 ± 1.5	**76.5 ± 0.2**
PFD	29.0 ± 1.5	27.0 ± 1.1	22.4 ± 0.7	36.3 ± 1.5	34.7 ± 1.8	37.7 ± 1.2	33.0 ± 1.4	29.0 ± 1.5	34.7 ± 1.9	**35.9 ± 1.5**
NONPL	49.7 ± 0.2	49.3 ± 0.2	-	56.7 ± 0.0	50.4 ± 0.0	55.0 ± 0.0	48.7 ± 0.2	49.7 ± 0.2	52.0 ± 0.0	**69.3 ± 0.0**
FLO17	50.8 ± 1.5	44.9 ± 2.4	37.5 ± 1.6	54.2 ± 2.2	58.5 ± 1.5	50.0 ± 0.8	61.0 ± 0.7	50.8 ± 1.5	59.5 ± 1.3	**61.3 ± 1.3**
FLO102	27.1 ± 0.8	22.4 ± 0.5	-	39.5 ± 0.7	40.2 ± 0.9	41.4 ± 0.8	38.4 ± 1.2	27.1 ± 0.8	42.5 ± 0.8	**44.0 ± 1.0**
REUTERS	45.5 ± 1.5	45.4 ± 1.5	-	40.9 ± 2.1	39.7 ± 1.5	40.0 ± 2.2	45.4 ± 1.7	45.5 ± 1.5	45.5 ± 0.7	**46.6 ± 1.0**
NMI										
MSRA	74.0 ± 1.0	73.2 ± 1.7	75.0 ± 1.4	74.9 ± 0.7	77.6 ± 0.3	79.8 ± 0.2	79.4 ± 0.8	74.0 ± 1.0	75.2 ± 0.5	**82.6 ± 1.5**
STILL	12.8 ± 0.8	13.0 ± 0.8	13.2 ± 0.5	12.9 ± 0.3	12.9 ± 0.4	12.9 ± 0.1	11.9 ± 0.5	12.8 ± 0.8	12.8 ± 1.0	**13.8 ± 0.8**
CAL-7	59.1 ± 2.9	51.6 ± 4.1	52.1 ± 1.3	63.5 ± 2.4	64.1 ± 0.2	65.3 ± 0.7	70.1 ± 3.0	59.1 ± 2.9	63.7 ± 0.3	**74.6 ± 1.2**
PFD	40.3 ± 1.3	38.0 ± 0.6	34.7 ± 0.6	44.4 ± 0.9	43.7 ± 1.2	46.2 ± 0.6	41.7 ± 1.1	40.3 ± 1.3	44.4 ± 1.1	**45.2 ± 1.3**
NONPL	17.2 ± 0.5	15.0 ± 0.5	-	11.8 ± 0.0	14.8 ± 0.0	16.0 ± 0.0	13.0 ± 0.1	17.2 ± 0.5	11.2 ± 0.0	**22.6 ± 0.0**
FLO17	49.7 ± 1.0	44.9 ± 1.5	38.8 ± 1.1	52.6 ± 1.2	56.4 ± 0.9	49.8 ± 0.6	**58.9 ± 0.4**	49.7 ± 1.0	57.8 ± 0.9	**58.9 ± 0.5**
FLO102	46.0 ± 0.5	42.7 ± 0.2	-	56.1 ± 0.4	56.7 ± 0.5	56.9 ± 0.3	54.9 ± 0.4	46.0 ± 0.5	58.6 ± 0.5	**60.0 ± 0.4**
REUTERS	27.4 ± 0.4	27.3 ± 0.4	-	21.0 ± 1.8	21.3 ± 1.3	21.5 ± 2.3	27.2 ± 0.2	27.4 ± 0.4	**27.7 ± 0.2**	27.0 ± 0.6
PURITY										
MSRA	83.3 ± 0.8	81.5 ± 2.7	81.9 ± 0.7	85.4 ± 0.4	88.1 ± 0.1	89.1 ± 0.2	87.8 ± 0.4	83.3 ± 0.8	86.5 ± 0.2	**91.2 ± 1.0**
STILL	33.8 ± 0.8	33.8 ± 0.7	33.3 ± 0.5	34.2 ± 0.9	34.1 ± 1.0	36.1 ± 0.2	35.0 ± 0.5	33.8 ± 0.8	33.8 ± 0.7	**38.2 ± 1.1**
CAL-7	68.0 ± 3.2	63.8 ± 3.7	66.4 ± 0.6	74.0 ± 2.1	72.9 ± 0.3	76.6 ± 0.4	79.6 ± 2.9	68.0 ± 3.2	72.3 ± 0.2	**81.7 ± 1.3**
PFD	37.4 ± 1.7	33.7 ± 1.1	31.2 ± 1.0	42.7 ± 1.3	41.9 ± 1.4	43.7 ± 0.8	39.3 ± 1.5	37.4 ± 1.7	41.8 ± 1.5	**42.5 ± 1.6**
NONPL	72.5 ± 0.2	71.2 ± 0.2	-	62.3 ± 0.1	60.4 ± 0.0	61.6 ± 0.1	69.7 ± 0.1	**72.5 ± 0.2**	60.4 ± 0.0	70.6 ± 0.0
FLO17	51.9 ± 1.5	46.2 ± 2.0	39.2 ± 1.3	55.4 ± 2.2	59.7 ± 1.6	51.4 ± 0.7	**62.4 ± 0.7**	51.9 ± 1.5	60.9 ± 1.2	62.0 ± 1.3
FLO102	32.3 ± 0.6	27.8 ± 0.4	-	45.6 ± 0.7	46.3 ± 0.8	48.0 ± 0.6	44.6 ± 0.8	32.3 ± 0.6	48.6 ± 0.7	**50.3 ± 0.7**
REUTERS	53.0 ± 0.4	52.9 ± 0.5	-	51.8 ± 1.5	50.9 ± 1.4	51.9 ± 1.0	52.9 ± 0.3	53.0 ± 0.4	**53.3 ± 0.0**	52.8 ± 0.2
RAND INDEX										
MSRA	68.1 ± 1.0	66.2 ± 3.1	68.0 ± 1.1	69.8 ± 0.7	74.5 ± 0.1	76.7 ± 0.4	74.5 ± 0.8	68.1 ± 1.0	71.2 ± 0.5	**80.6 ± 1.8**
STILL	8.0 ± 0.5	8.0 ± 0.5	8.0 ± 0.2	8.2 ± 0.2	8.1 ± 0.3	7.7 ± 0.0	7.7 ± 0.4	8.0 ± 0.5	7.9 ± 0.5	**9.3 ± 0.3**
CAL-7	46.0 ± 6.5	38.3 ± 4.9	41.2 ± 1.1	56.8 ± 4.2	55.6 ± 0.6	59.4 ± 2.2	65.2 ± 3.4	46.0 ± 6.5	55.6 ± 0.3	**69.4 ± 0.7**
PFD	14.4 ± 1.8	12.1 ± 0.7	7.8 ± 0.4	18.0 ± 1.1	17.2 ± 1.5	20.1 ± 1.1	16.1 ± 1.5	14.4 ± 1.8	17.6 ± 1.9	**19.8 ± 1.2**
NONPL	17.6 ± 0.3	15.8 ± 0.4	-	14.2 ± 0.0	8.5 ± 0.0	10.4 ± 0.0	14.1 ± 0.2	17.6 ± 0.3	8.0 ± 0.0	**35.0 ± 0.0**
FLO17	32.2 ± 1.3	27.2 ± 1.8	20.6 ± 1.1	35.2 ± 1.5	39.9 ± 1.3	31.6 ± 0.8	**44.1 ± 0.4**	32.2 ± 1.3	41.5 ± 1.5	43.2 ± 0.9
FLO102	15.5 ± 0.5	12.1 ± 0.4	-	24.9 ± 0.5	25.5 ± 0.6	27.2 ± 0.6	25.5 ± 1.0	15.5 ± 0.5	28.5 ± 0.8	**29.9 ± 0.8**
REUTERS	21.8 ± 1.4	21.8 ± 1.4	-	18.8 ± 2.4	18.9 ± 2.0	16.9 ± 2.7	21.4 ± 1.1	21.8 ± 1.4	**22.1 ± 0.8**	21.5 ± 0.3

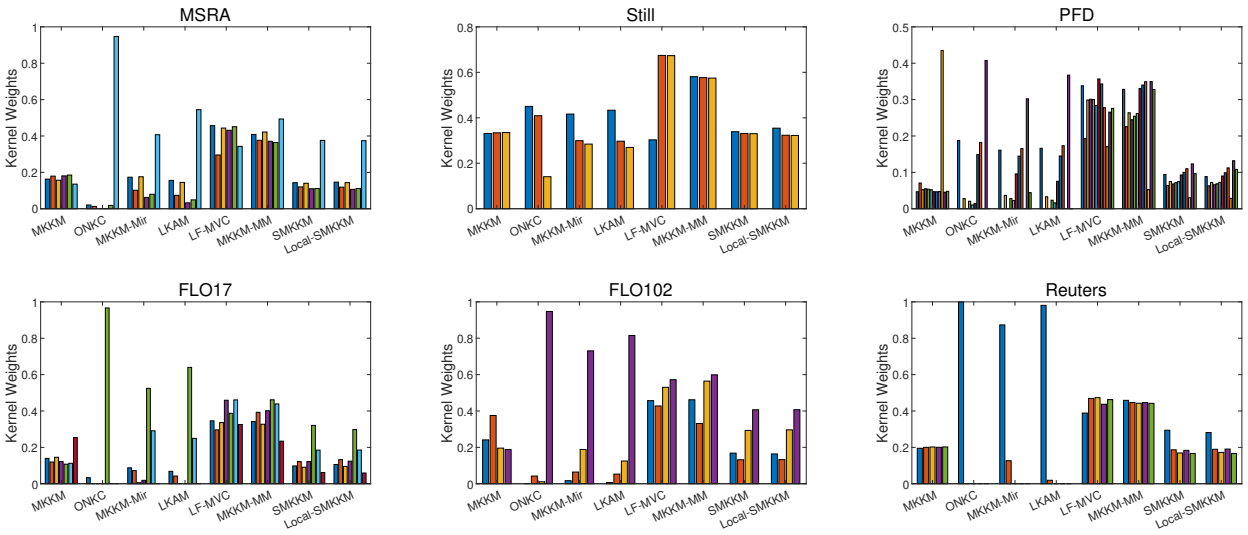


Figure 1. The kernel weights learned by different algorithms. The results on other datasets omitted due to space limit.

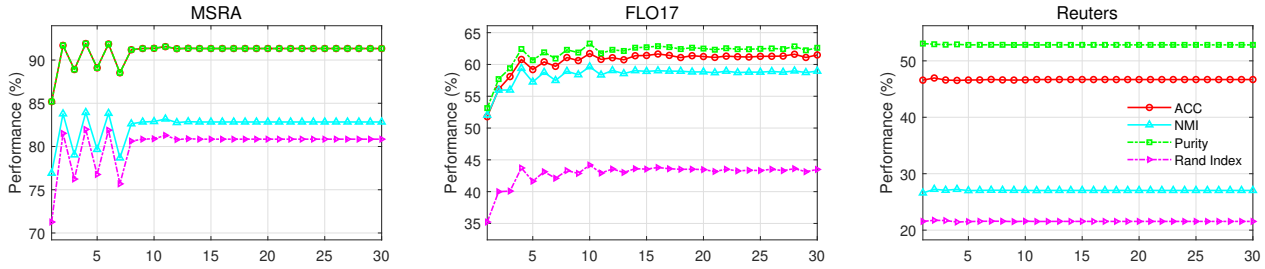


Figure 2. The clustering performance of the learned $\mathbf{H}$ by localized SimpleMKKM with iterations on three benchmark datasets. The curves on other datasets are omitted due to space limits.

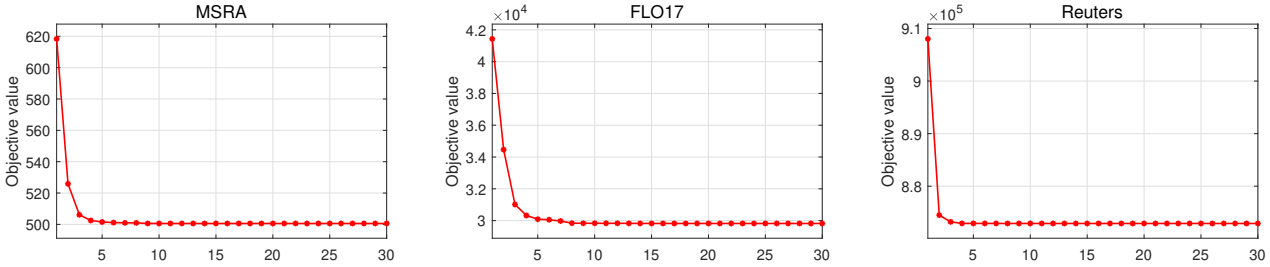
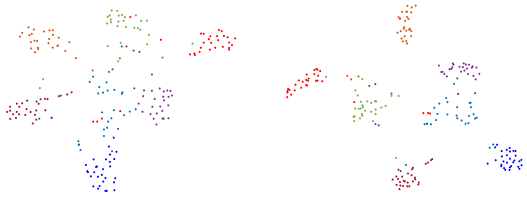


Figure 3. The objective of localized SimpleMKKM varies with iterations. The curves on other datasets are omitted due to space limits.

The improvements in terms of other criteria are similar. Furthermore, a visual comparison is presented in Fig. 4. As seen, proposed algorithm shows clearer clustering structure compared with SimpleMKKM. These results well demonstrate the superiority of the proposed localized SimpleMKKM that benefits from exploring and extracting the localized information of kernel matrix.

Besides inheriting the advanced formulation and new optimization from SimpleMKKM, the proposed algorithm adopts 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. We expect that its simplicity and efficacy will make it a good option to be applied into practical clustering applications. In addition, we point out that the results of LMKKM [2] on some datasets are not reported due to the out-of-memory error, which are caused by its cubic computational and memory complexity.



(a) SimpleMKKM (b) Proposed

Figure 4. Clustering structure Visual comparison on MSRA.

Table 3. The optimal hyper-parameters for each algorithm.

Datasets	ONKC	MKKM-MiR	LKAM	LF-MVC	Ours
MSRA	$\rho = 0.5, \lambda = 4$	$\lambda = 1$	$\tau = 0.05, \lambda = 2$	$\lambda = 0.5$	$\tau = 0.55$
STILL	$\rho = 0.25, \lambda = 0.5$	$\lambda = 0.25$	$\tau = 0.05, \lambda = 0.5$	$\lambda = 1$	$\tau = 0.55$
CAL-7	$\rho = 4, \lambda = 0.25$	$\lambda = 1$	$\tau = 0.05, \lambda = 1$	$\lambda = 1$	$\tau = 0.05$
PFD	$\rho = 2, \lambda = 4$	$\lambda = 2$	$\tau = 0.85, \lambda = 4$	$\lambda = 0.25$	$\tau = 0.85$
NONPL	$\rho = 0.25, \lambda = 4$	$\lambda = 1$	$\tau = 0.25, \lambda = 4$	$\lambda = 4$	$\tau = 0.05$
FLO17	$\rho = 4, \lambda = 0.25$	$\lambda = 0.25$	$\tau = 0.05, \lambda = 4$	$\lambda = 0.5$	$\tau = 0.35$
FLO102	$\rho = 2, \lambda = 2$	$\lambda = 2$	$\tau = 0.05, \lambda = 1$	$\lambda = 0.25$	$\tau = 0.65$
REUTERS	$\rho = 0.5, \lambda = 4$	$\lambda = 1$	$\tau = 0.05, \lambda = 2$	$\lambda = 0.5$	$\tau = 0.4$

Kernel Weight Analysis. We further investigate 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 ONKC, MKKM-MiR and LKAM vary greatly on almost all datasets, and are highly sparse on some datasets such as Nonpl and Reuters. This sparsity would make the multiple kernel matrices insufficiently exploited, leading to poor performance. For example, the clustering accuracy of MKKM-MiR and LKAM on Reuters is only 39.7% and 40.0%. In contrast, despite the ℓ_1-norm constraint on $\boldsymbol{\gamma}$, the kernel weights learned by our 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 $\min_{\boldsymbol{\gamma}}$-$\max_{\mathbf{H}}$ kernel alignment objective.

Convergence and Evolution of the Learned H. As proved in Section 3.3, localized SimpleMKKM is guaranteed to converge theoretically. To see this point in depth, we plot the objective of localized SimpleMKKM with iterations on all datasets, as shown in Figure 3. From the figures, we observe that its objective is monotonically de-

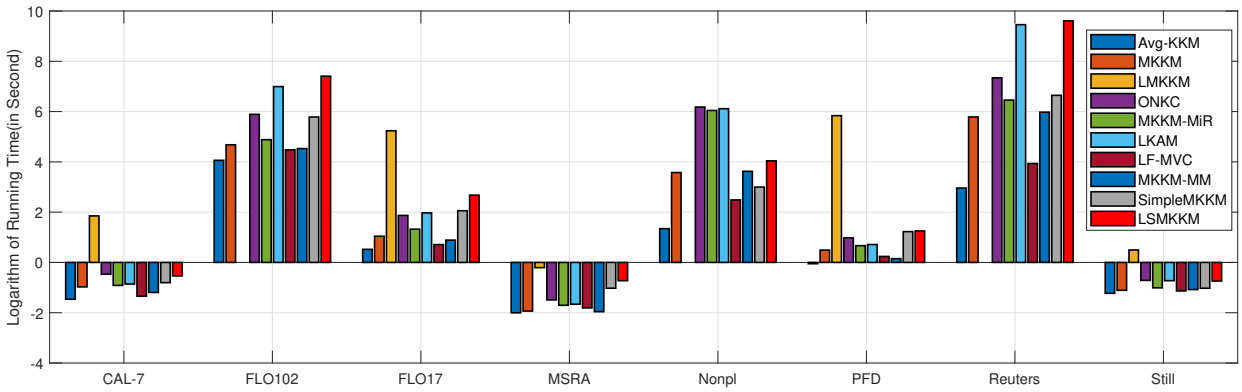


Figure 5. Running time comparison of different algorithms on all benchmark datasets (logarithm in seconds). The experiments are conducted on a PC with Intel(R) Core(TM)-i9-10900X 3.7GHz CPU and 64G RAM in MATLAB R2020b environment.

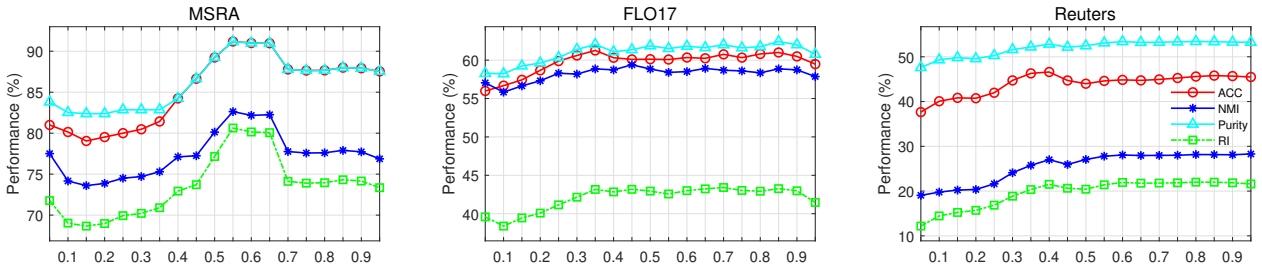


Figure 6. The effect of the size of neighborhood τ on the clustering performance on three representative datasets. The curves on other datasets are omitted due to space limits.

creased and usually converges in several iterations on all datasets. Also, to reveal the clustering performance variation of the learned $\mathbf{H}$ with the iteration increases, we calculate ACC, NMI, purity and RI at each iteration, and plot them in Figure 2. As observed, the clustering performance of localized SimpleMKKM is firstly increased with iterations, slightly oscillates and then remains stable. The result reveals the effectiveness and necessity of the learning procedure.

Parameter Sensitivity Analysis. The newly proposed localized SimpleMKKM introduces a hyper-parameter τ to preserve more reliable neighborhood structure among samples. We conduct an additional experiment to show the effect of this parameter on the clustering performance, as shown in Figure 6. Here τ varies from 0.05 to 0.95 with step size as 0.05. From this figure, we observe that the newly proposed algorithm shows stable performance across a wide range of τ values, indicating its robustness with the variation of the hyper-parameter. In addition, the optimal hyper-parameters for each algorithm (if have) have been listed in Table 3 for repeatability.

Running Time Comparison. Finally, we report the running time of the aforementioned algorithms on all datasets, as plotted in Figure 5. We observe that besides greatly improving the clustering performance, the proposed localized SimpleMKKM does not significantly increase the computational cost.

5. Conclusion

While the recently proposed SimpleMKKM demonstrates promising clustering performance, it does not sufficiently consider the variation among base kernel matrices. This paper proposes to calculate the kernel alignment in a local manner to address this issue. We uncover the theoretical connection between SimpleMKKM and the proposed algorithm. Based on this observation, we adopt a reduced gradient descent algorithm to solve the resultant optimization problem. Moreover, the proposed localized SimpleMKKM demonstrates significantly improved clustering results via extensive experiments on multiple benchmark data sets. In the future, instead of keeping the nearest neighbors of each sample unchanged, we plan to further improve clustering by automatically updating them during the learning course.

Acknowledgements

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

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

[2] 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.

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

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

[5] 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*.

[6] Nazli Ikizler, Ramazan Gokberk Cinbis, Selen Pehlivan, and Pinar Duygulu. Recognizing actions from still images. In *19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA*, pages 1–4. IEEE Computer Society, 2008.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] Xinwang Liu, Miaomiao Li, Chang Tang, Jingyuan Xia, Jian Xiong, Li Liu, Marius Kloft, and En Zhu. Efficient and effective regularized incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[13] X. Liu, L. Wang, X. Zhu, M. Li, E. Zhu, T. Liu, L. Liu, Y. Dou, and J. Yin. Absent multiple kernel learning algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[14] 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*, 2017.

[15] 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.

[16] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao. Late fusion incomplete multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2410–2423, Oct 2019.

[17] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao. Multiple kernel k-means with incomplete kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2019.

[18] 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.

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

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

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

[22] 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.

[23] 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.

[24] John M. Winn and Nebojsa Jojic. LOCUS: learning object classes with unsupervised segmentation. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, pages 756–763. IEEE Computer Society, 2005.

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

[26] Shi Yu, Leon Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan A. K. Suykens, Bart De Moor, and Yves Moreau. Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Software Engineering*, 34(5):1031–9, 2012.

[27] 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.

[28] 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.

[29] 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.