SimpleMKKM: Simple Multiple Kernel K-means

Xinwang Liu, *Senior Member, IEEE*

Abstract—We propose a simple yet effective multiple kernel clustering algorithm, termed simple multiple kernel k-means (SimpleMKKM). It extends the widely used supervised kernel alignment criterion to multi-kernel clustering. Our criterion is given by an intractable minimization-maximization problem in the kernel coefficient and clustering partition matrix. To optimize it, we equivalently rewrite the minimization-maximization formulation as a minimization of an optimal value function, prove its differenentiablity, and design a reduced gradient descent algorithm to decrease it. Furthermore, we prove that *the resultant solution of SimpleMKKM is the global optimum*. We theoretically analyze the performance of SimpleMKKM in terms of its clustering generalization error. After that, we develop extensive experiments to investigate the proposed SimpleMKKM from the perspectives of clustering accuracy, advantage on the formulation and optimization, variation of the learned consensus clustering matrix with iterations, clustering performance with varied number of samples and base kernels, analysis of the learned kernel weight, the running time and the global convergence. The experimental study demonstrates the effectiveness of the proposed SimpleMKKM by considerably and consistently outperforming state of the art multiple kernel clustering alternatives. In addition, the ablation study shows that the improved clustering performance is contributed by both the novel formulation and new optimization. Our work provides a more effective approach to integrate multi-view data for clustering, and this could trigger novel research on multiple kernel clustering. The source code and data for SimpleMKKM are available at https://github.com/xinwangliu/SimpleMKKMcodes/.

Index Terms—multi-view clustering, multiple kernel clustering, kernel alignment maximization

1 Introduction

In multiple kernel clustering (MKC) [1], one aims to optimally fuse a set of pre-calculated base kernel matrices to achieve better clustering performance. These kernel matrices could encode heterogeneous sources or views of the data [2], [3], [4]. One popular method, multiple kernel k-means (MKKM) [5], has been studied intensively and used in various applications [2], [6], [7], [8], [9], [10], [11]. The approach is attractive also from a theoretical perspective, as it unifies the search of the optimal base kernel coefficient and clustering partition matrix into a single objective function, which is usually solved by a two-step alternate optimization on the coefficient and clustering partition matrix.

Several variants of MKKM have been developed to further improve the clustering performance [2], [6], [12], [13], [14]. Notably, [6] substantially increases the expressiveness of MKKM by allowing for a locally adaptive kernel mixtures, which can better capture sample-specific characteristics of data. [12] proposes an extension that optimizes a localized kernel alignment criterion. It aligns the local density of the samples given by the k-nearest neighbours with an ideal similarity matrix. This alignment helps to keep neighbouring sample pairs together, which avoids unreliable similarity evaluation. Observing that existing MKKM algorithms do not sufficiently take the correlation among these kernels into consideration, [13] employs a matrix regularization to reduce the redundancy and enhance the diversity of the selected kernels. Most of existing MKKM algorithms assume that the optimal kernel is a linear combination of a group of base kernels. This assumption is challenged in [14], where an optimal neighborhood kernel clustering (ONKC) algorithm is introduced to boost the representability of the optimal kernel and strengthen the negotiation between clustering and the learning of kernel weights. More recently, MKKM algorithms have been extended to handle missing views [15]. By assuming that an optimal kernel is a linear combination of base kernel matrices, the work in [16] develops a minimization-maximization criterion that aims to be robust to adversarial perturbation. More recently, many works have been devoted to extending existing MKKM to handle multiple kernel clustering with incomplete kernels [15], [17], [18], [19], [20]. All these variants potentially improve standard MKKM from different aspects and achieve promising clustering performance in various applications.

The objective functions of the mentioned methods differ, but they all share one commonality: they jointly learn the kernel coefficient and clustering partition matrix. By this way, the learned kernel coefficient can best serve the clustering, with aim to achieve superior clustering performance. However, simultaneously solving for the kernel coefficient *and* clustering partition is usually intractable. One commonly adopted remedy is to decouple the optimization of the kernel coefficient and clustering partition through a block coordinate descent algorithm, which optimizes the two alternately. This means, one block of variables is minimized while the other is kept fixed. However, such alternate optimization algorithms can get trapped into a local optimum of the objective function. As a remedy, [12], [13] propose regularization strategies to avoid getting trapped into a local minimum. The incorporation of these regularization terms comes at a price: the approach has additional hyper-parameters, which are difficult to select, given the unsupervised nature of clustering tasks.

In this paper, we propose a novel formulation for

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

Manuscript received February 19, 2022.

multiple kernel clustering, termed Simple MKKM (SimpleMKKM), to address the aforementioned shortcomings. Unlike previous approaches, SimpleMKKM optimizes the unsupervised kernel alignment criterion directly. Specifically, it minimizes the kernel alignment with respect to the kernel coefficient and maximizes it with respect to the clustering matrix. By this way, SimpleMKKM is expected to learn a good clustering matrix for clustering even under a bad kernel coefficient, leading to stable and superior clustering performance. However, this minimization-maximization optimization problem cannot be readily solved using existing alternate optimization frameworks since its objective value cannot be guaranteed to monotonically decrease anymore. To address this issue, we firstly reformulate the min-max formulation as a minimization problem, whose objective relies on the known optimal solution to kernel k-means. We then prove the differentiability of the optimal value function and calculate its reduced gradient. This leads to a solution using a reduced gradient descent algorithm, without alternate optimization. Moreover, we theoretically prove that *the resultant solution of SimpleMKKM is the global optimum*. We further show a generalization error bound for our approach, thus theoretically guaranteeing its clustering performance. After that, we conduct comprehensive experiments on eleven benchmark datasets, where we compare SimpleMKKM with ten baseline methods in terms of four common evaluation criteria. It is observed that SimpleMKKM consistently outperforms its competitors. Moreover, we conduct extra experimental study from the following aspects: advantage on the formulation and optimization, variation of the learned consensus clustering matrix with iterations, clustering performance with varied number of samples and base kernels, analysis of the learned kernel weight, running time and global convergence.

The main contributions of this work are summarized as follows:

- We, for the first time, propose a novel minimization-maximization formulation to optimize the extensively used criterion for kernel alignment. We then reformulate our formulation as a minimization of an optimal value function w.r.t. the kernel weights. Further, we prove the differentiability of the resultant minimization, and develop a reduced gradient descent to decease it. Moreover, we theoretically show that the obtained solution is the global optimum. As far as we know, *our SimpleMKKM is the first algorithm with global optimum in multiple kernel clustering literature*.
- We theoretically analyze the performance of SimpleMKKM in terms of its clustering generalization error on test data.
- Extensive experimental results on various benchmarks have demonstrated the superiority and effectiveness of the proposed SimpleMKKM. As shown by the ablation study, both our novel formulation and new optimization attribute to enhanced clustering performance.

Additionally, the proposed SimpleMKKM is parameter-free, making it readily applicable in practice. More importantly, SimpleMKKM can be taken as a strong baseline to trigger new research on multiple kernel clustering.

This section is ended up by clarifying the relationship between SimpleMKKM and one piece of our newly published work [20]. In [20], a local kernel alignment criterion which could better capture the variation among samples is proposed. It is also shown that SimpleMKKM is a special case of [20]. Though the newly proposed variant demonstrates improved clustering clustering performance in some applications, it is clearly that *both of its formulation and optimization are inherited from SimpleMKKM*. Moreover, different from SimpleMKKM which is free of hyper-parameters, the local variant has an extra hyper-parameter which controls the size of neighborhood for each sample to be pre-specified. However, selecting suitable hyper-parameters for a given clustering task itself is a puzzle due to the absence of ground truth during the learning course. In addition, SimpleMKKM is also extended to handle multiple kernel clustering with incomplete kernels [21].

2 Related Work

In this section, we briefly review the most related, including multiple kernel k-means (MKKM) and robust MKKM clustering using min-max optimization [16].

2.1 MKKM

In MKKM, an optimal kernel matrix $\mathbf{K}_{\boldsymbol{\gamma}}$ is parameterized by $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^{m} \gamma_p^2 \mathbf{K}_p$, where $\{\mathbf{K}_p\}_{p=1}^{m}$ is a group of pre-calculated kernel matrices, and γ_p donoted the weights of the p-th base kernel. MKKM simultaneously learns $\boldsymbol{\gamma}$ and a clustering partition matrix $\mathbf{H}$ by optimizing the following formulation,

$$\min_{\boldsymbol{\gamma}\in\Delta}\ \min_{\mathbf{H}\in\Gamma}\ \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}-\mathbf{H}\mathbf{H}^{\top})\right), \tag{1}$$

where $\Delta = \{\boldsymbol{\gamma} \in \mathbb{R}^m | \sum_{p=1}^{m} \gamma_p = 1,\ \gamma_p \geq 0,\ \forall p\}$ and $\Gamma = \{\mathbf{H} \in \mathbb{R}^{n\times k} | \mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_k\}$.

Existing algorithms usually solve Eq. (1) by alternatively optimizing $\mathbf{H}$ and $\boldsymbol{\gamma}$: (i) **Optimizing $\mathbf{H}$ given $\boldsymbol{\gamma}$**. For a specific kernel coefficient $\boldsymbol{\gamma}$, the optimization in Eq. (1) w.r.t $\mathbf{H}$ is equivalent to the following Eq. (2),

$$\max_{\mathbf{H}\in\Gamma}\ \mathrm{Tr}\left(\mathbf{H}^{\top}\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\right). \tag{2}$$

Eq. (2) is a classical kernel k-means that can be readily optimized by off-the-shelf packages. (ii) **Optimizing $\boldsymbol{\gamma}$ given $\mathbf{H}$**. For a specific $\mathbf{H}$, the optimization in Eq. (1) w.r.t $\boldsymbol{\gamma}$ reduces to the following Eq. (3),

$$\min_{\boldsymbol{\gamma}\in\Delta}\ \sum_{p=1}^{m} \gamma_p^2 \mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n-\mathbf{H}\mathbf{H}^{\top})\right), \tag{3}$$

which can be analytically obtained.

Algorithm 1 presents the detailed MKKM optimization procedure, where $\mathbf{H}$ and $\boldsymbol{\gamma}$ are alternately optimized until convergence.

As mentioned in [2], [6], performing a convex combination of kernels $\sum_{p=1}^{m} \gamma_p \mathbf{K}_p$ to replace $\sum_{p=1}^{m} \gamma_p^2 \mathbf{K}_p$ is not a viable option, because this could make only one single kernel activated and all the others assigned with zero, as seen from Eq. (3). Other recent works using ℓ_2-norm combinations can be found in [15], [22], [23].

Algorithm 1 MKKM

```
Input: $\{\mathbf{K}_p\}_{p=1}^m,\ k,\ t=1$.
Initialize $\boldsymbol{\gamma}^{(1)} = \mathbf{1}/m$ and flag $= 1$.
while flag do
    compute $\mathbf{H}^{(t)}$ in Eq. (2) with $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}} = \sum_{p=1}^m \left(\gamma_p^{(t)}\right)^2 \mathbf{K}_p$.
    update $\boldsymbol{\gamma}^{(t+1)}$ in Eq. (3) with $\mathbf{H}^{(t)}$.
    if $\max |\boldsymbol{\gamma}^{(t+1)} - \boldsymbol{\gamma}^{(t)}| \leq e^{-4}$ then
        flag $= 0$.
    end if
    $t \leftarrow t+1$.
end while
```

2.2 Robust MKKM Using Min-Max Optimization

Recently, [16] proposed a MKKM clustering method with the aim to be robust against adversarial perturbation. To achieve this goal, the authors use a $\min_{\mathbf{H}}$-$\max_{\gamma}$ formulation that combines views so as to achieve high within-cluster variance in the combined space $\mathbf{W}_\gamma$ and then updates clusters by minimizing such variance. Its optimization problem is,

$$\min_{\mathbf{H}\in\Gamma}\ \max_{\boldsymbol{\gamma}\in\Theta}\ \mathrm{Tr}\left(\mathbf{W}_{\gamma}(\mathbf{I}-\mathbf{H}\mathbf{H}^\top)\right) \tag{4}$$

where $\Theta = \{\boldsymbol{\gamma}\in\mathbb{R}^m |\ \sum_{p=1}^m \gamma_p^2 \leq 1,\ \gamma_p \geq 0,\ \forall p\}$ and $\mathbf{W}_\gamma = \sum_{p=1}^m \gamma_p \mathbf{K}_p$.

Similar to MKKM, the problem in Eq. (4) is solved by following the same alternate optimization framework: optimizing one variable with the other fixed. Specifically, with $\boldsymbol{\gamma}$ fixed, Eq. (4) w.r.t. $\mathbf{H}$ reduces to a classical kernel k-means with $\mathbf{W}_{\boldsymbol{\gamma}^{(t)}} = \sum_{p=1}^m \gamma_p^{(t)}\mathbf{K}_p$. With the fixed $\mathbf{H}$, Eq. (4) w.r.t. $\boldsymbol{\gamma}$ is equivalent to Eq. (5),

$$\max_{\boldsymbol{\gamma}\in\Theta}\ \sum_{p=1}^m \gamma_p \mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top)\right), \tag{5}$$

which has an analytical solution.

We present the whole optimization procedure in solving Eq. (4) in Algorithm 2. As done in MKKM, $\mathbf{H}$ and $\boldsymbol{\gamma}$ are also alternately optimized until stopping condition being satisfied.

Algorithm 2 MKKM-MM

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

Note that in contrast to Eq. (1), the above approach adopts an ℓ_2-norm constraint on the kernel weights to avoid sparse solutions. It is observed that using an ℓ_2-norm constraint can obtain a non-sparse kernel coefficient, which is helpful to better utilize the complementary information in the data [16].

Although the objective functions of MKKM and its variants may vary, they share a common alternate optimization routine. The aforementioned alternate framework could cause the optimization w.r.t. $\boldsymbol{\gamma}$ to produce high redundant or overly sparse solutions [13]. This in turn would make the multiple kernel matrices less utilized, and adversely affects the clustering performance. A direct remedy is to incorporate some regularization on $\boldsymbol{\gamma}$ to help its optimization [12], [13]. However, the incorporation of regularization may introduce extra hyper-parameters. How to determine those in unsupervised learning tasks such as clustering is difficult. In the following, we introduce our SimpleMKKM objective, and design a novel optimization procedure for it that avoids these issues.

3 SimpleMKKM: Simple MKKM

In this section, we first give the proposed SimpleMKKM kernel alignment-based objective. We then reformulate it as the minimization of an optimal value function, and prove its differentiability. After that, we develop a reduced gradient descent algorithm to solve it efficiently and effectively. Further, we discuss its computational complexity, and prove the global optimum of our SimpleMKKM.

3.1 SimpleMKKM Formulation

Kernel alignment criterion has been widely used for kernel tuning in supervised learning due to its simplicity and effectiveness [24], [25]. Our new formulation is based on unsupervised multiple kernel alignment criterion, inspired by existing supervised kernel learning. One can optimize this criterion by maximizing over both $\boldsymbol{\gamma}$ and $\mathbf{H}$. Though theoretically elegant, we empirically observe that such $\max_{\boldsymbol{\gamma}} \max_{\mathbf{H}}$ formulation does not achieve promising clustering performance, which is different from supervised kernel learning. We conjecture that is caused by the over-fitted optimization between $\boldsymbol{\gamma}$ and $\mathbf{H}$. On the other hand, from the optimization perspective of MKKM in Eq. (1), $\mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}-\mathbf{H}\mathbf{H}^\top)\right)$ should be minimized. This objective can be decomposed into two terms, $\mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\right)$ and $-\mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top\right)$. The first term can be regarded as regularization on $\boldsymbol{\gamma}$, which should be minimized w.r.t. $\boldsymbol{\gamma}$. The other one is the opposite of kernel alignment, which should be maximized w.r.t. $\mathbf{H}$. By taking both regularization and partitioning into account, our SimpleMKKM proposes to minimize the kernel alignment w.r.t. $\boldsymbol{\gamma}$ and maximize this criterion w.r.t. $\mathbf{H}$ as:

$$\min_{\boldsymbol{\gamma}\in\Delta}\ \max_{\mathbf{H}\in\Gamma}\ \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top\right), \tag{6}$$

where $\Delta = \{\boldsymbol{\gamma}\in\mathbb{R}^m |\ \sum_{p=1}^m \gamma_p = 1,\ \gamma_p \geq 0,\ \forall p\}$, $\Gamma = \{\mathbf{H}\in\mathbb{R}^{n\times k} | \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\}$ and $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^m \gamma_p^2 \mathbf{K}_p$.

Though simple, the SimpleMKKM formulation in Eq. (6) has the following merits: (1) It is the first MKKM objective that, strictly coincides with the kernel alignment criterion via $\mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top\right)$ to tune kernel weights. In contrast, MKKM and its all variants adopt $\mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}-\mathbf{H}\mathbf{H}^\top)\right)$ as the criterion by extending the objective of classic kernel k-means to multiple kernels. It is worth noting that the kernel alignment criterion is more general and can be used for any kernel tuning tasks. As a result, it can be used for multiple kernel clustering. (2) According to [16], regularisation by min-max optimization of $\boldsymbol{\gamma}$ and $\mathbf{H}$ generates more robust clusters by avoiding overfitting to noisy views or datapoints. (3) As we shall see next, while our formulation looks intractible, it actually leads to a more efficient and effective optimization algorithm than the standard alternate strategies used for MKKM. Furthermore, unlike alternatives [12], [13] relying on regularization by penalizing $\boldsymbol{\gamma}$, SimpleMKKM introduces no additional parameter beyond the number of clusters to form.

Our new formulation in Eq. (6) cannot be readily solved by the widely adopted alternate optimization strategy, as done in MKKM and its variants since its objective value cannot be guaranteed to monotonically decrease. In the following, we design an efficient and effective reduced gradient descent algorithm. Firstly, we equivalently rewrite the optimization in Eq. (6) as,

$$\min\nolimits_{\boldsymbol{\gamma}\in\Delta}\ \mathcal{J}(\boldsymbol{\gamma}), \tag{7}$$

with

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

By this means, we transform the min-max optimization procedure to a minimization one, where the corresponding objective is an optimal value function dependent on kernel k-means. In the following, we first prove the differentiability of $\mathcal{J}(\boldsymbol{\gamma})$ w.r.t. $\boldsymbol{\gamma}$, and apply the reduced gradient descent algorithm to decrease Eq. (7).

3.2 The Differentiability and Calculation of Gradient

In literature, several works discuss the existence and computation of derivatives of optimal value functions $\mathcal{J}(\boldsymbol{\gamma})$ [26], [27], [28]. The most appropriate reference for our case is Theorem 4.1 in [26], which has already been utilized to tune the hyper-parameters of SVM [27] and optimize the kernel weights in multiple kernel learning [28]. The following Theorem 1 shows that $\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (7) is differentiable.

Theorem 1. *$\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (7) is differentiable. Further, $\frac{\partial\mathcal{J}(\boldsymbol{\gamma})}{\partial\gamma_p}=2\gamma_p\mathrm{Tr}\left(\mathbf{K}_p\mathbf{H}^*\mathbf{H}^{*\top}\right)$, where $\mathbf{H}^*=\left\{\arg\max_{\mathbf{H}\in\Gamma}\ \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^{\top}\right)\right\}$.*

Proof. For any given $\boldsymbol{\gamma}\in\Delta$, the global maximum $\tilde{\mathbf{H}}^*$ of the optimization problem $\max_{\mathbf{H}\in\Gamma}\ \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^{\top}\right)$ satisfies $\tilde{\mathbf{H}}^*\in\{\tilde{\mathbf{H}}^*|\tilde{\mathbf{H}}^*=\mathbf{H}^*\mathbf{U},\ \mathbf{U}\mathbf{U}^{\top}=\mathbf{U}^{\top}\mathbf{U}=\mathbf{I}_k\}$. Though the global maximum is not unique, $\tilde{\mathbf{H}}^*\tilde{\mathbf{H}}^{*\top}$ is unique. According to Theorem 4.1 in [26], $\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (7) is differentiable, and $\frac{\partial\mathcal{J}(\boldsymbol{\gamma})}{\partial\gamma_p}=2\gamma_p\mathrm{Tr}(\mathbf{K}_p\tilde{\mathbf{H}}^*\tilde{\mathbf{H}}^{*\top})=2\gamma_p\mathrm{Tr}(\mathbf{K}_p\mathbf{H}^*\mathbf{H}^{*\top})$. □

3.3 The Calculation of Reduced Gradient and Optimization Algorithm

We propose to solve the optimization in Eq. (7) with reduced gradient descent algorithms. To do so, we firstly calculate the gradient of $\mathcal{J}(\boldsymbol{\gamma})$ according to Theorem 1, 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 [28]. Let γ_u be a non-zero component of $\boldsymbol{\gamma}$ and $\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})$ denote the reduced gradient of $\mathcal{J}(\boldsymbol{\gamma})$, of which the p-th $(1\leq p\leq m)$ element is

$$[\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})]_p=\frac{\partial\mathcal{J}(\boldsymbol{\gamma})}{\partial\gamma_p}-\frac{\partial\mathcal{J}(\boldsymbol{\gamma})}{\partial\gamma_u}\ \ \forall\, p\neq u, \tag{9}$$

and

$$[\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})]_u=\sum\nolimits_{p=1,p\neq u}^{m}\left(\frac{\partial\mathcal{J}(\boldsymbol{\gamma})}{\partial\gamma_u}-\frac{\partial\mathcal{J}(\boldsymbol{\gamma})}{\partial\gamma_p}\right). \tag{10}$$

According to the literature [28], u is selected to be the index of the largest component of vector $\boldsymbol{\gamma}$ that is considered to provide better numerical stability.

Next, the positivity constraints on $\boldsymbol{\gamma}$ are taken into consideration in the descent direction. Since our goal is to minimize $\mathcal{J}(\boldsymbol{\gamma})$, it is worth noting that $-\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})$ refers to a descent direction. Nevertheless, if we directly adopt this direction for optimization, the positivity constraints may not hold when there exists an index p satisfying $\gamma_p=0$ and $[\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})]_p>0$. In this circumstance, the descent direction for that component should be set to 0, which could update $\boldsymbol{\gamma}$ along with the descent direction

$$d_p=\begin{cases}0 & \text{if } \gamma_p=0 \text{ and } [\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})]_p>0\\ -[\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})]_p & \text{if } \gamma_p>0 \text{ and } p\neq u\\ -[\bigtriangledown\mathcal{J}(\boldsymbol{\gamma})]_u & \text{if } p=u.\end{cases} \tag{11}$$

According to Eq. (11), we can obtain a descent direction $\mathbf{d}=[d_1,\cdots,d_m]^{\top}$, hence $\boldsymbol{\gamma}$ can be computed via the parameter updating strategy $\boldsymbol{\gamma}\leftarrow\boldsymbol{\gamma}+\alpha\mathbf{d}$. Here we denote α as the optimal step length, which can be chosen via a linear search mechanism, e.g., Armijo's rule. Algorithm 3 presents the detailed optimization procedure of our proposed SimpleMKKM.

Algorithm 3 SimpleMKKM

```
Input: $\{\mathbf{K}_p\}_{p=1}^{m}$, $k$, $t=1$.
Initialize $\boldsymbol{\gamma}^{(1)}=\mathbf{1}/m$ and set flag $=1$.
while flag do
    calculate $\mathbf{H}$ by solving a kernel k-means with $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}}=\sum_{p=1}^{m}\left(\gamma_p^{(t)}\right)^2\mathbf{K}_p$.
    compute $\frac{\partial\mathcal{J}(\boldsymbol{\gamma})}{\partial\gamma_p}$ $(p=1,\cdots,m)$ and the descent direction $\mathbf{d}^{(t)}$ in Eq. (11).
    update $\boldsymbol{\gamma}^{(t+1)}\leftarrow\boldsymbol{\gamma}^{(t)}+\alpha\mathbf{d}^{(t)}$.
    if $\max|\boldsymbol{\gamma}^{(t+1)}-\boldsymbol{\gamma}^{(t)}|\leq e^{-4}$ then
        flag=0.
    end if
    $t\leftarrow t+1$.
end while
```

3.4 The Global Convergence

We analyze the convergence of SimpleMKKM in Algorithm 3. Note that Eq. (8) is a traditional kernel k-means that has the global optimal solution. Under this circumstance, the gradient computation in Theorem 1 is accurate, and our SimpleMKKM conducts the reduced gradient descent on a continuously differentiable function $\mathcal{J}(\boldsymbol{\gamma})$, which is defined on the simplex $\{\boldsymbol{\gamma}\in\mathbb{R}^m|\sum_{p=1}^{m}\gamma_p=1,\ \gamma_p\geq 0,\ \forall p\}$. It does converge to the minimum of $\mathcal{J}(\boldsymbol{\gamma})$ [28]. Furthermore, Theorem 2 illustrates that $\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (7) is a convex function of $\boldsymbol{\gamma}$.

Theorem 2. *$\mathcal{J}(\boldsymbol{\gamma})$ in Eq. (7) is convex w.r.t. $\boldsymbol{\gamma}$.*

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

$$\begin{aligned}
&\mathcal{J}(\alpha\boldsymbol{\gamma}_1 + (1-\alpha)\boldsymbol{\gamma}_2) \\
&= \max_{\mathbf{H}\in\Gamma} \mathrm{Tr}\left(\mathbf{K}_{\alpha\boldsymbol{\gamma}_1+(1-\alpha)\boldsymbol{\gamma}_2}\mathbf{H}\mathbf{H}^\top\right) \\
&= \max_{\mathbf{H}\in\Gamma} \mathrm{Tr}\left(\sum\nolimits_{p=1}^{m} \left(\alpha\gamma_{1p} + (1-\alpha)\gamma_{2p}\right)^2 \mathbf{K}_p\mathbf{H}\mathbf{H}^\top\right) \\
&\leq \max_{\mathbf{H}\in\Gamma} \mathrm{Tr}\left(\sum\nolimits_{p=1}^{m} \left(\alpha\gamma_{1p}^2 + (1-\alpha)\gamma_{2p}^2\right) \mathbf{K}_p\mathbf{H}\mathbf{H}^\top\right) \\
&\leq \alpha \max_{\mathbf{H}\in\Gamma} \mathrm{Tr}\left(\sum\nolimits_{p=1}^{m} \gamma_{1p}^2\mathbf{K}_p\mathbf{H}\mathbf{H}^\top\right) \\
&\qquad + (1-\alpha) \max_{\mathbf{H}\in\Gamma} \mathrm{Tr}\left(\sum\nolimits_{p=1}^{m} \gamma_{2p}^2\mathbf{K}_p\mathbf{H}\mathbf{H}^\top\right) \\
&= \alpha\mathcal{J}(\boldsymbol{\gamma}_1) + (1-\alpha)\mathcal{J}(\boldsymbol{\gamma}_2)
\end{aligned} \tag{12}$$

□

According to Theorem 2, the solution obtained by SimpleMKKM in Algorithm 3 is the global optimum. This implies that our SimpleMKKM is independent of any initialization, as verified by the results in Figure 6. As far as we know, *our SimpleMKKM is the first algorithm with theoretically global optimum in multiple kernel clustering literature*.

3.5 Discussion

We discuss the computational complexity of SimpleMKKM. From Algorithm 3, at each iteration, SimpleMKKM needs to solve a kernel k-means problem, calculate the reduced gradient, and search optimal step size. Therefore, its computational complexity at each iteration is $\mathcal{O}(n^3+m*n^2+m*n_0)$, where n_0 is the maximal number of operations required to find the optimal step size. As observed, SimpleMKKM does not significantly increase the computational complexity of existing MKKM algorithms, as also validated by the experimental results in Figure 7.

We conclude this section by discussing the differences with MKKM-MM [16]. Though both works share a min-max (max-min) framework, their differences can be summarized from the following three aspects: (1) The objectives are different. SimpleMKKM adopts the unsupervised kernel alignment criterion while MKKM-MM inherits the objective of MKKM, which can be clearly seen from Eq. (4) and Eq. (6). Further, MKKM-MM applies the ℓ_2-norm constraints on $\boldsymbol{\gamma}$ to avoid sparse solutions. However, although using the ℓ_1-norm constraint, our SimpleMKKM still obtains non-sparse solution, as shown by the results in Figure 5. (2) More importantly, the optimization strategies are totally different. MKKM-MM follows the widely used alternate optimization paradigm to solve Eq. (4). In contrast, we, for the first time, reformulate the SimpleMKKM as a minimization problem, and develop a reduced gradient descent algorithm to efficiently solve it. More importantly, the resultant solution is guaranteed to be the global optimum. (3) The clustering performance is different. We empirically compare their clustering performance, and observe that SimpleMKKM consistently and significantly outperforms MKKM-MM on all 11 benchmark datasets, as shown in Table 2.

4 The Generalization Analysis

Generalization error for k-means clustering has been studied by fixing the centroids obtained in the training process and computing their generalization to test data [29], [30]. In this section, we study how the centroids obtained by the proposed SimpleMKKM generalize onto test data by deriving its generalization bound.

We now define the error of SimpleMKKM. Let $\hat{\mathbf{C}} = [\hat{\mathbf{C}}_1, \cdots, \hat{\mathbf{C}}_k]$ be the learned matrix composed of the k centroids and $\hat{\boldsymbol{\gamma}}$ the learned kernel weights by the proposed SimpleMKKM, where $\hat{\mathbf{C}}_v = \frac{1}{|\hat{\mathbf{C}}_v|}\sum_{j\in\hat{\mathbf{C}}_v} \phi_{\hat{\boldsymbol{\gamma}}}(\mathbf{x}_j), 1 \leq c \leq k$. By defining $\Theta = \{\mathbf{e}_1, \cdots, \mathbf{e}_k\}$, effective SimpleMKKM clustering should make the following error small

$$1 - \mathbb{E}_{\mathbf{x}}\left[\max_{\mathbf{y}\in\Theta}\langle\phi_{\hat{\boldsymbol{\gamma}}}(\mathbf{x}), \hat{\mathbf{C}}\mathbf{y}\rangle_{\mathcal{H}^k}\right], \tag{13}$$

where $\phi_{\hat{\boldsymbol{\gamma}}}(\mathbf{x}) = [\hat{\gamma}_1\phi_1^\top(\mathbf{x}), \cdots, \hat{\gamma}_m\phi_m^\top(\mathbf{x})]^\top$ is the resultant feature map related to the kernel function $K_{\hat{\boldsymbol{\gamma}}}(\cdot,\cdot)$ and $\mathbf{e}_1, \cdots, \mathbf{e}_k$ are formulated as the orthogonal bases of $\mathbb{R}^k$. Commonly, it is expected that the test points can achieve strong alignment with the closest clustering center. In the following, we illustrate the mechanism to reach the goal of our developed SimpleMKKM.

Firstly, we formulate a function class:

$$\begin{aligned}
\mathcal{F} = \Big\{ f :\ & \mathbf{x} \mapsto 1 - \max_{\mathbf{y}\in\Theta}\langle\phi_{\boldsymbol{\gamma}}(\mathbf{x}), \mathbf{C}\mathbf{y}\rangle_{\mathcal{H}^k} \Big| \boldsymbol{\gamma}^\top\mathbf{1}_m = 1, \\
& \gamma_p \geq 0, \mathbf{C}\in\mathcal{H}^k,\ |K_p(\mathbf{x},\tilde{\mathbf{x}})| \leq b,\ \forall p, \forall\mathbf{x}\in\mathcal{X} \Big\},
\end{aligned} \tag{14}$$

where $\mathcal{H}^k$ indicates the multiple kernel Hilbert space.

Theorem 3. *For any $\delta > 0$, Eq. (15) holds with probablity not less than$1-\delta$ for all $f \in \mathcal{F}$:*

$$\mathbb{E}\left[f(\mathbf{x})\right] \leq \frac{1}{n}\sum\nolimits_{i=1}^{n} f(\mathbf{x}_i) + \frac{\sqrt{\pi/2}bk}{\sqrt{n}} + (1+b)\sqrt{\frac{\log 1/\delta}{2n}}. \tag{15}$$

The detailed proof is provided in the appendix due to conciseness and readability.

According to Theorem 3, for any learned $\hat{\boldsymbol{\gamma}}$ and $\hat{\mathbf{C}}$, to achieve a small

$$\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] = 1 - \mathbb{E}_{\mathbf{x}}\left[\max_{\mathbf{y}\in\Theta}\left\langle\phi_{\hat{\boldsymbol{\gamma}}}(\mathbf{x}), \hat{\mathbf{C}}\mathbf{y}\right\rangle_{\mathcal{H}^k}\right], \tag{16}$$

the corresponding $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$ needs to be as small as possible. Assume that $\boldsymbol{\gamma}$ and $\mathbf{C}$ are obtained by minimizing $\frac{1}{n}\sum_{i}^{n} f(\mathbf{x}_i)$ and that $\mathbf{H}$ is constrained to be orthogonal, we have

$$\frac{1}{n}\sum\nolimits_{i=1}^{n} f(\mathbf{x}_i) \leq 1 - \frac{1}{n}\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top), \tag{17}$$

because the proposed algorithm poses a constraint $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k$ which will make the corresponding centroids non-optimal for minimizing $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$. This means that $1 - \frac{1}{n}\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top)$ is an upper bound of $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$. To minimize the upper bound, we may have to maximize over $\boldsymbol{\gamma}$ and $\mathbf{H}$, leading to $\max_{\boldsymbol{\gamma}}\max_{\mathbf{H}}\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top)$. However, it is intractable to find a good solution to $\boldsymbol{\gamma}$ and $\mathbf{H}$ under this criterion, and it is prone to over-fitted solutions [16]. Instead, we take one of its lower bounds, $\min_{\boldsymbol{\gamma}}\max_{\mathbf{H}}\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^\top)$ as the the objective of SimpleMKKM in Eq. (6). This analysis verifies the good generalization ability of the proposed SimpleMKKM.

TABLE 1: Specification of our 11 benchmark datasets.

Dataset	Number of		
	Samples	Kernels	Clusters
Flo17	1360	7	17
Flo102	8189	4	102
PFold	694	12	27
CCV	6773	3	20
Digit	2000	3	10
Cal-5	510	48	102
Cal-10	1020	48	102
Cal-15	1530	48	102
Cal-20	2040	48	102
Cal-25	2550	48	102
Cal-30	3060	48	102

5 Experimental Results

In this section, we conduct a comprehensive experimental study to evaluate the proposed SimpleMKKM in terms of clustering performance, the learned kernel weights, the running time, and convergence.

5.1 Experimental Settings

A number of standard MKKM benchmark datasets are adopted to evaluate SimpleMKKM, including *Flo17*[1], *Flo102*[2], *PFold*[3], *CCV*[4], *Digit*[5], *Cal*[6]. Meanwhile, six sub-datasets, i.e. *Cal-5*, *Cal-10*, *Cal-15*, *Cal-20*, *Cal-25* and *Cal-30*, are constructed via selecting the first 5, 10, 15, 20, 25 and 30 samples from each class respectively from the *Caltech102* data. Their details are shown in Table 1. It can be observed that the number of samples, kernels and categories of these datasets shows considerable variation.

For all data sets, the number of clusters k is assumed known and is set as the true number of classes. We adopt four widely used clustering metrics to evaluate all compared algorithms, including Accuracy (ACC), Normalized Mutual Information (NMI), Purity, and Rand Index. To reduce the negative affect of randomness caused by k-means, we repeat all methods 50 times and show the average performance with standard deviation.

We next thoroughly study SimpleMKKM in terms of: clustering performance, ablation study on the formulation and optimization, evolution of the learned $\mathbf{H}$, clustering with number of samples, clustering with number of base kernels, the learned kernel weights, running time and algorithm convergence. Along with SimpleMKKM, we ran another ten comparative algorithms in recent MKC literature, including

- **Average kernel k-means (Avg-KKM)**. The consensus kernel is the uniformly combined base kernels, which is taken as the input of kernel k-means.
- **Multiple kernel k-means (MKKM)** [5]. 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)** [6]. The base kernels are combined with sample-adaptive weights.
- **Optimal neighborhood kernel clustering (ONKC)** [31]. The consensus kernel is chosen from the neighbor of linearly combined base kernels.
- **Multiple kernel k-means with matrix-induced regularization (MKKM-MiR)** [13]. The optimal combination weights are learned by introducing a matrix-induced regularization term to reduce the redundancy among the base kernels.
- **Mulitple kernel clustering with local alignment maximization (LKAM)** [12]. 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)** [32]. Base partitions are firstly calculated using each single view and then optimally integrated into a consensus partition.
- **Multiple kernel clustering based on centered kernel alignment (CKAMKC)** [33]. Two tasks of clustering and multiple kernel learning are unified into a single optimization framework.
- **Kernel-based Weighted Multi-view Clustering (MVKKM)** [34]. The weighted combination of different kernels, indicating the quality of the corresponding views, is learned during partitioning.
- **MKKM-MM** [16]. It proposes a $\min_{\mathbf{H}}$-$\max_{\gamma}$ 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.

The implementations of the above algorithms are publicly available in corresponding papers, and we directly adopt them without revision in our experiments. Among all the compared algorithms, ONKC [31], MKKM-MiR [13], LKAM [12] and LF-MVC [32] have hyper-parameters to be tuned. Note that hyper-parameter tuning is very difficult and still remains an open problem in clustering tasks. We reproduce their public released codes and report the best corresponding results by following the settings of original literature. These algorithms have several hyper-parameters to turn and by this means the resultant clustering performance may be over-estimated. Consequently, tuning hyper-parameters would hinder the compared algorithms from real-world applications. It is therefore desired to develop a parameter-free algorithm for multiple kernel clustering, as the proposed SimpleMKKM does.

5.2 Experimental Results

5.2.1 Clustering Performance

Table 2 presents the ACC, NMI, purity and rand index comparison of the above algorithms. From this table, we have the following observations:

- The proposed SimpleMKKM consistently and significantly outperforms MKKM. For example, it exceeds MKKM by 12.7%, 16%, 6.1%, 3.1%, 34.6%, 4.4%, 7.2%, 8.9%, 10.1%, 10.6% and 11.7% in terms of ACC on all

[1] www.robots.ox.ac.uk/~vgg/data/flowers/17/
[2] www.robots.ox.ac.uk/~vgg/data/flowers/102/
[3] mkl.ucsd.edu/dataset/protein-fold-prediction
[4] www.ee.columbia.edu/ln/dvmm/CCV/
[5] http://ss.sysu.edu.cn/py/
[6] www.vision.caltech.edu/Image_Datasets/Caltech101/

TABLE 2: Clustering performance of SimpleMKKM and ten baselines on five benchmarks in terms of ACC, NMI, Purity, and Rand Index. The boldface values indicate the best results.

DATASETS	AVG-KKM	MKKM	LMKKM	ONKC	MKKM-MIR	LKAM	LF-MVC	CKAMKC	MVKKM	MKKM-MM	SIMPLEMKKM
		[5]	[6]	[31]	[13]	[12]	[32]	[33]	[34]	[16]	PROPOSED
ACC											
FLO17	51.0± 1.3	43.6± 1.7	42.7± 1.5	43.4± 2.1	58.0± 1.2	48.9± 0.9	57.2± 1.3	34.2 ± 0.0	21.8 ± 0.0	51.0±1.3	**59.1± 1.2**
FLO102	27.1± 0.8	22.4± 0.5	-	39.2± 0.9	39.1± 1.3	40.4± 1.0	29.0± 1.0	21.0 ± 0.0	–	27.1±0.8	**42.5± 0.8**
PFOLD	29.0± 1.6	27.0± 1.1	22.4± 0.7	**35.3± 1.3**	34.3± 1.7	33.8± 1.7	31.6± 1.7	31.7 ± 0.0	24.1 ± 0.0	29.0±1.6	**34.7± 1.9**
CCV	19.6± 0.6	18.0± 0.5	18.6± 0.1	22.1± 0.6	20.9± 0.9	18.9± 0.3	**23.1± 0.9**	20.7 ± 0.0	16.6 ± 0.0	19.6±0.6	22.2± 0.7
DIGIT	88.8± 0.1	47.3± 0.7	47.3± 0.7	89.5± 0.1	87.4± 0.1	**95.0 ± 0.1**	89.1± 0.1	90.6 ± 0.0	39.6 ± 0.0	88.8±0.7	90.3± 0.1
AVG.	43.1	31.7	-	45.9	47.9	47.4	46.0	39.6	-	43.1	**49.8**
NMI											
FLO17	49.6± 0.8	44.3± 1.3	43.8± 1.0	43.1± 1.3	56.2± 0.6	48.2± 0.6	54.6± 0.9	39.1 ± 0.0	23.7 ± 0.0	49.7±0.8	**57.5± 0.8**
FLO102	46.0± 0.5	42.7± 0.2	-	55.7± 0.4	55.9± 0.6	55.8± 0.3	47.5± 0.3	41.9 ± 0.0	–	46.0±0.5	**58.6± 0.5**
PFOLD	40.3± 1.2	38.0± 0.6	34.7± 0.6	**44.0± 0.8**	43.1± 1.0	43.6± 1.0	41.8± 0.9	38.9 ± 0.0	12.0 ± 0.0	40.3±1.3	**44.4± 1.1**
CCV	16.8± 0.4	15.1± 0.5	14.4± 0.1	18.4± 0.3	17.9± 0.4	16.8± 0.2	**19.3± 0.3**	18.4 ± 0.0	46.8 ± 0.0	16.8±0.4	18.2± 0.3
DIGIT	80.8± 0.2	48.8± 0.7	48.7± 0.7	81.7± 0.1	79.6± 0.1	**89.4± 0.1**	81.1± 0.2	83.6 ± 0.0	15.0 ± 0.0	80.8±0.2	83.3± 0.1
AVG.	46.7	37.8	-	48.6	50.5	50.8	48.9	44.2	-	46.7	**52.4**
PURITY											
FLO17	52.0± 1.0	45.1± 1.4	44.5± 1.4	45.2± 1.9	59.4± 0.9	50.1± 0.6	58.1± 1.4	36.3 ± 0.0	22.9 ± 0.0	52.0±1.0	**60.5± 1.4**
FLO102	32.3± 0.6	27.8± 0.4	-	45.1± 0.9	45.2± 1.0	46.7± 0.6	34.5± 0.5	26.8 ± 0.0	–	32.3±0.6	**48.6± 0.7**
PFOLD	37.4± 1.7	33.7± 1.1	31.2± 1.0	**41.9± 1.0**	41.2± 1.4	**41.6± 1.3**	38.9± 1.5	36.7 ± 0.0	27.2 ± 0.0	37.4±1.7	**41.8± 1.5**
CCV	23.8± 0.5	22.2± 0.5	22.0± 0.1	24.3± 0.5	23.4± 0.7	22.2± 0.3	**26.1± 0.5**	23.3 ± 0.0	20.0 ± 0.0	23.8±0.5	25.3± 0.5
DIGIT	88.8± 0.1	50.1± 0.7	50.1± 0.7	89.5± 0.1	87.4± 0.1	**95.0± 0.1**	89.1± 0.1	90.6 ± 0.0	43.9 ± 0.0	88.8± 0.1	90.3± 0.1
AVG.	46.9	35.8	-	49.2	51.3	51.1	49.3	42.7	-	46.9	**53.3**
RAND INDEX											
FLO17	32.3± 1.0	26.4± 1.3	26.0± 1.1	24.3± 1.6	39.6± 0.8	30.2± 0.8	38.6± 1.0	20.5 ± 0.0	12.9 ± 0.0	32.3±1.3	**41.3± 1.1**
FLO102	15.5± 0.5	12.1± 0.4	-	24.5± 0.6	24.9± 1.0	26.3± 0.6	17.2± 0.8	10.7 ± 0.0	–	15.5±0.5	**28.5± 0.8**
PFOLD	14.4± 1.8	12.1± 0.7	7.8± 0.4	**17.6± 1.3**	**17.4± 1.6**	**17.3± 1.7**	16.2± 1.7	14.4 ± 0.0	5.6 ± 0.0	14.4±1.8	**17.6± 1.9**
CCV	6.6± 0.2	5.8± 0.2	5.6± 0.1	7.5 ± 0.3	7.0± 0.4	6.2± 0.1	**8.4± 0.5**	7.1 ± 0.0	5.0 ± 0.0	6.6 ±0.2	7.5± 0.2
DIGIT	77.5± 0.2	31.4± 0.6	31.3± 0.6	78.7± 0.1	75.4± 0.1	**89.2± 0.1**	78.2± 0.2	90.7 ± 0.0	31.4 ± 0.0	77.5±0.2	80.3± 0.1
AVG.	29.3	17.6	-	30.5	32.9	33.8	31.7	28.7	-	29.3	**35.0**

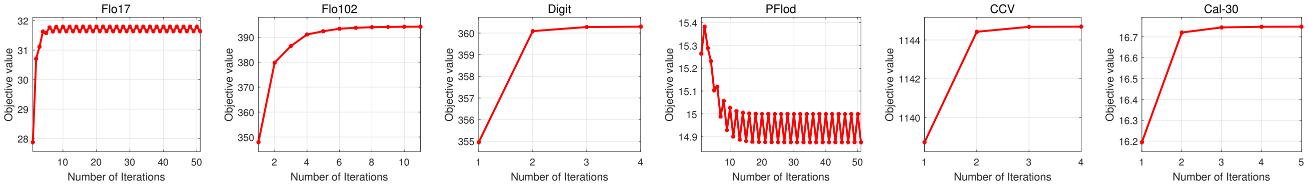

Fig. 1: The objective value of SimpleMKKM-C with the variation of iterations on all benchmarks. SimpleMKKM-C adopts an alternate optimization strategy to solve the objective of SimpleMKKM.

benchmark datasets. These results demonstrate the efficacy of its min-max formulation and associated optimization algorithm.

- MKKM-MM [16] is the first try in literature to improve MKKM via minimization-maximization. As observed, it does improve the MKKM. However the improvement over MKKM is marginal on all datasets. Meanwhile, the proposed SimpleMKKM significantly outperforms MKKM-MM. This once again demonstrates the advantage of our formulation and the associated optimization strategy.
- Our SimpleMKKM achieves comparable or slightly better performance than MKKM-MiR [13], ONKC [31], and LF-MVC [32], all of which are considered the state of the art in multi-kernel clustering. Note that all of these algorithms have several hyper-parameters to tune due to the incorporation of regularization on the kernel weights γ. Though demonstrating promising clustering performance, these algorithms need to take a lot of efforts to determine the best hyper-parameters in practical applications. And parameter tuning may be impossible in real applications where there is no ground truth clustering to optimize. In contrast, our SimpleMKKM is parameter-free.

In summary, SimpleMKKM demonstrates superior clustering performance over the alternatives on all datasets and has no hyper-parameter to be tuned. We expect that the simplicity and efficacy of SimpleMKKM will make it a good option to be considered for practical clustering applications. Note that some results of LMKKM [6] and MVKKM [34] are not presented because they require $\mathcal{O}(n^3)$ computational complexity that causes the out-of-memory error.

5.2.2 Ablation Study on the Formulation and Optimization

In order to show the advantage of the proposed formulation and optimization algorithm, we conduct an ablation study on all benchmark datasets to compare the alternatives MKKM-R and SimpleMKKM-C. MKKM-R denotes optimizing the objective of existing MKKM in Eq. (1) with reduced gradient descent, while SimpleMKKM-C denotes optimizing the criterion in Eq. (6) with coordinate descent optimization (see Section 3.1 for discussion). Note that SimpleMKKM-C has the same objective as SimpleMKKM, but it uses the widely adopted alternate optimization to solve it in place of our newly derived reduced gradient algorithm.

From the results reported in Table 3, we clearly observe that: (1) Our SimpleMKKM and SimpleMKKM-C formulations have significant advantages over MKKM and

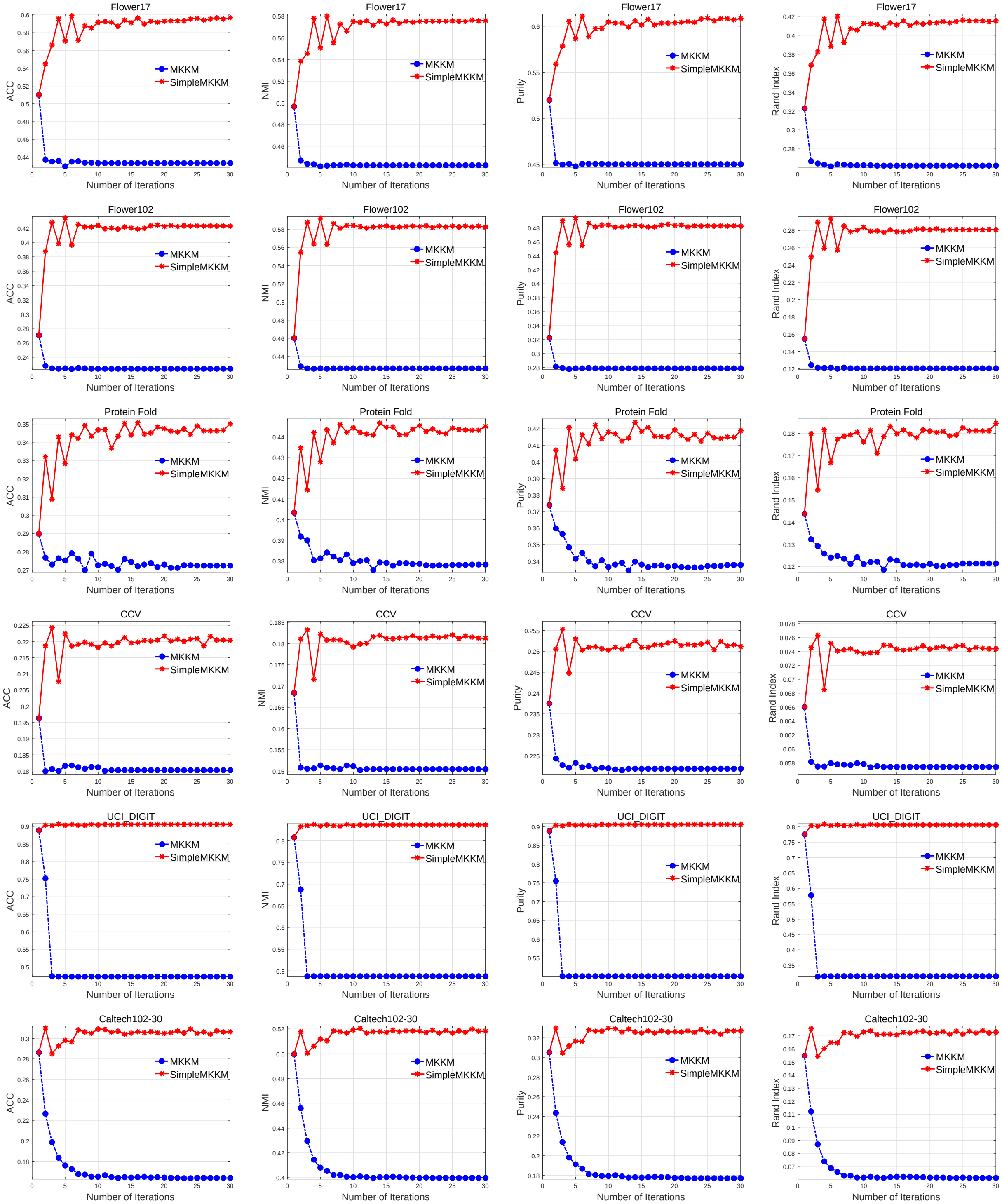

Fig. 2: Clustering comparison of the learned $\mathbf{H}$ by MKKM and the proposed SimpleMKKM with iterations.

MKKMR, demonstrating the value of our novel min-max objective; (2) It is also observed that our SimpleMKKM outperforms SimpleMKKM-C, which confirms that our new gradient based optimization algorithm is also much better than the widely used alternate optimization. This ablation study well demonstrates that both our novel formulation and new optimization attribute to the improvement of clustering performance. In addition, we plot the objective value of SimpleMKKM-C with the variation of iterations in Figure 1 on all benchmark datastes. From Figure 1, we clearly see that SimpleMKKM-C cannot be guaranteed to monotonically decrease Eq. (6) with iterations. This is consistent with our previous claim that the proposed minimization-maximization optimization cannot be directly solved by the widely used coordinate descent.

5.2.3 Evolution of the Learned $\mathbf{H}$

To verify the superiority of our SimpleMKKM, we compare it with existing MKKM algorithms and report the clustering performance in terms of ACC, NMI, purity, and rand index with iterations. As shown in Figure 2, the start points of both SimpleMKKM and MKKM on all sub-figures are the

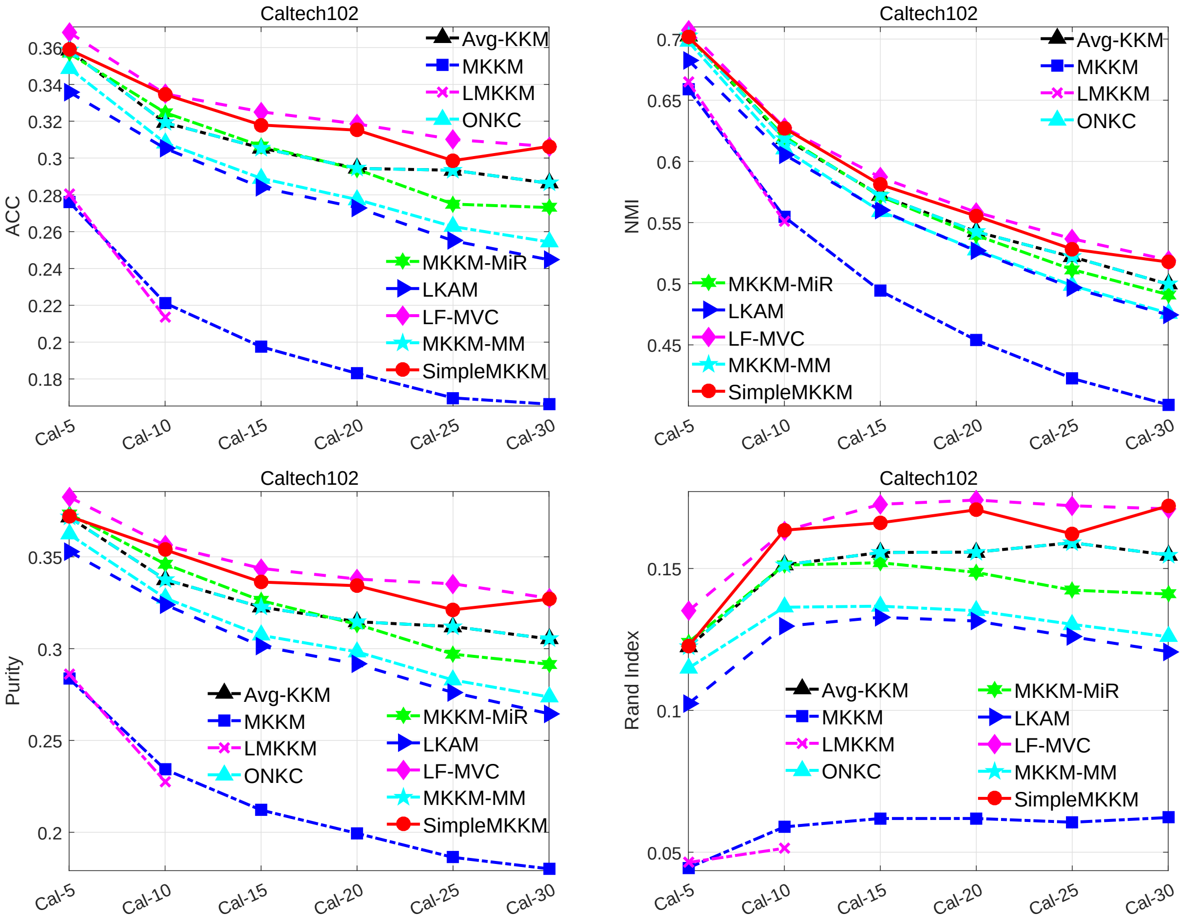

Fig. 3: Clustering performance of aforementioned algorithms with different number of samples on Caltech102.

TABLE 3: Empirical comparison of SimpleMKKM with MKKM, MKKM-R and SimpleMKKM-C on all datasets.

Dataset	MKKM [5]	MKKM-R	SimpleMKKM-C	SimpleMKKM
		ACC		
Flo17	43.6± 1.2	43.7± 1.4	54.2± 1.8	**59.1± 1.2**
Flo102	22.4± 0.5	22.4± 0.5	41.8± 1.2	**42.5± 0.8**
PFold	27.0± 1.1	26.6± 1.1	29.0± 1.4	**34.7± 1.9**
CCV	18.0± 0.5	17.9± 0.6	**22.1± 0.7**	**22.2± 0.7**
Digit	47.3± 0.7	47.3± 0.7	**90.4± 0.9**	**90.3± 0.6**
Cal-30	16.6± 0.4	16.7± 0.4	**30.4± 1.1**	**30.6± 0.9**
		NMI		
Flo17	44.3± 1.3	44.3± 1.1	54.3± 1.4	**57.5± 0.8**
Flo102	42.7± 0.2	42.6± 0.2	58.0± 0.5	**58.6± 0.5**
PFold	38.0± 0.6	37.5± 0.8	38.4± 0.8	**44.4± 1.1**
CCV	15.1± 0.5	14.8± 0.4	**18.2± 0.3**	**18.2± 0.3**
Digit	48.8± 0.7	48.7± 0.7	**83.5± 0.2**	**83.3± 0.1**
Cal-30	40.1± 0.3	40.2± 0.3	**51.8± 0.6**	**51.8± 0.5**
		Purity		
Flo17	45.1± 1.4	44.9± 1.4	55.1± 1.8	**60.5± 1.4**
Flo102	27.8± 0.4	27.8± 0.4	47.9± 0.8	**48.6± 0.7**
PFold	33.7± 1.1	33.1± 0.9	35.7± 1.0	**41.8± 1.4**
CCV	22.2± 0.5	22.3± 0.4	**25.2± 0.5**	**25.3± 0.5**
Digit	50.1± 0.7	50.1± 0.7	**90.4± 0.9**	**90.3± 0.6**
Cal-30	18.0± 0.5	18.1± 0.4	**32.5± 1.0**	**32.7± 0.8**
		Rand Index		
Flo17	45.1± 1.4	44.9± 1.4	55.1± 1.8	**60.5± 1.4**
Flo102	27.8± 0.4	27.8± 0.4	47.9± 0.8	**48.6± 0.7**
PFold	33.7± 1.1	33.1± 0.9	35.7± 1.0	**41.8± 1.4**
CCV	22.2± 0.5	22.3± 0.4	**25.2± 0.5**	**25.3± 0.5**
Digit	50.1± 0.7	50.1± 0.7	**90.4± 0.9**	**90.3± 0.6**
Cal-30	18.0± 0.5	18.1± 0.4	**32.5± 1.0**	**32.7± 0.8**

same. This is because both algorithms are initialized with the unified weights, which generates the same $\mathbf{H}$, learning to the same clustering performance. The clustering performance of the proposed SimpleMKKM presents a trend of first rising and then obtains relatively stable performance, which sufficiently verifies the superiority of our algorithm. In contrast, the clustering performance of MKKM is decreased with iterations on all sub-figures, implying that existing MKKM is inferior to average kernel k-means. This states that the widely used MKKM may not be a good choice to fuse multiple base kernels. Comparable, our proposed SimpleMKKM significantly outperforms average kernel k-means on all sub-figures, considerably showing the effectiveness and necessity of the learning procedure.

5.2.4 *Performance Comparison with Number of Samples*

In this subsection, we conduct an experiment to investigate the effect of different number samples for clustering performance of our SimpleMKKM on Caltech102. In specific, we evaluate their clustering performance on *Cal-5*, *Cal-10*, *Cal-15*, *Cal-20*, *Cal-25* and *Cal-30*, which are constructed by selecting the first 5, 10, 15, 20, 25 and 30 samples from each class respectively from the *Caltech102* data.

The ACC, NMI, purity, and rand index of these algorithms with the variation of number of samples are plotted in Figure 3. As observed, the proposed SimpleMKKM considerably boosts the clustering performance of existing MKKM and its variants. For instance, as presented in sub-figure 3a, SimpleMKKM outperforms MKKM by

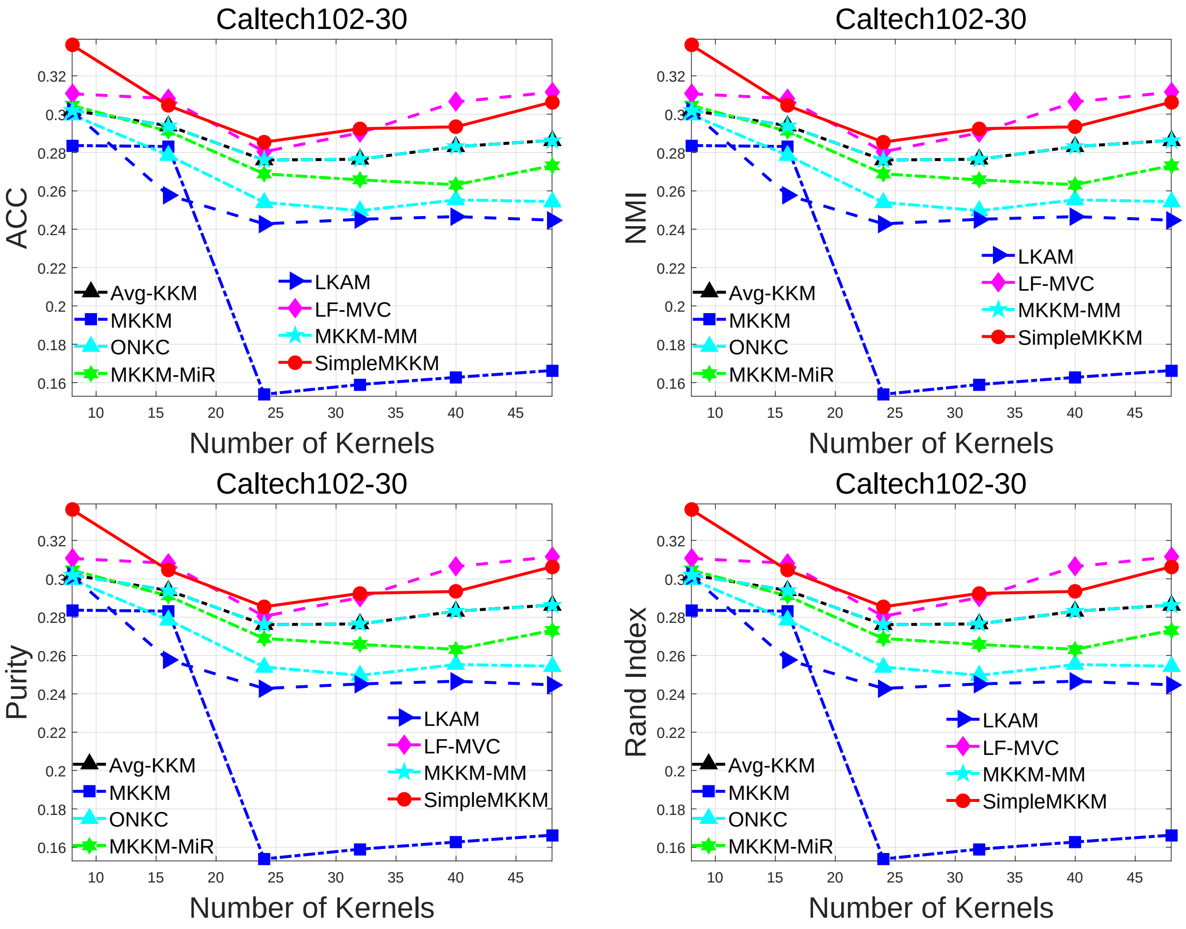

Fig. 4: Clustering performance of aforementioned algorithms with different number of base kernels on Caltech102.

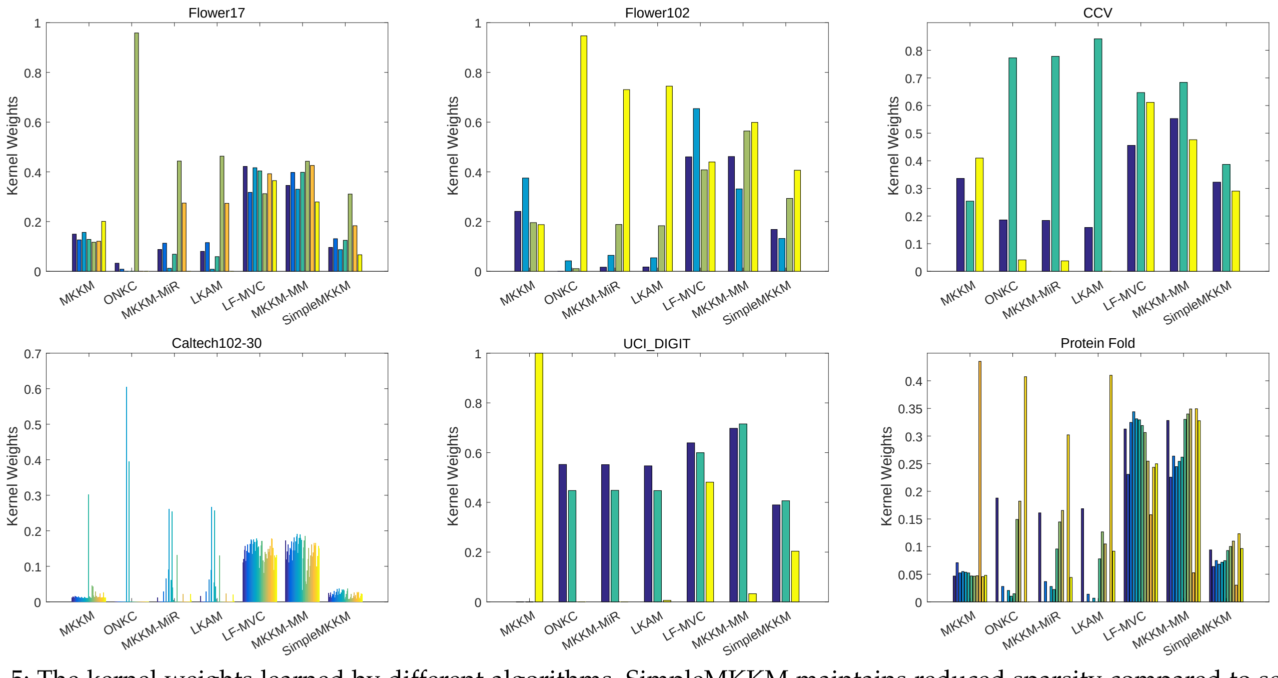

Fig. 5: The kernel weights learned by different algorithms. SimpleMKKM maintains reduced sparsity compared to several competitors. Other datasets omitted due to space limit.

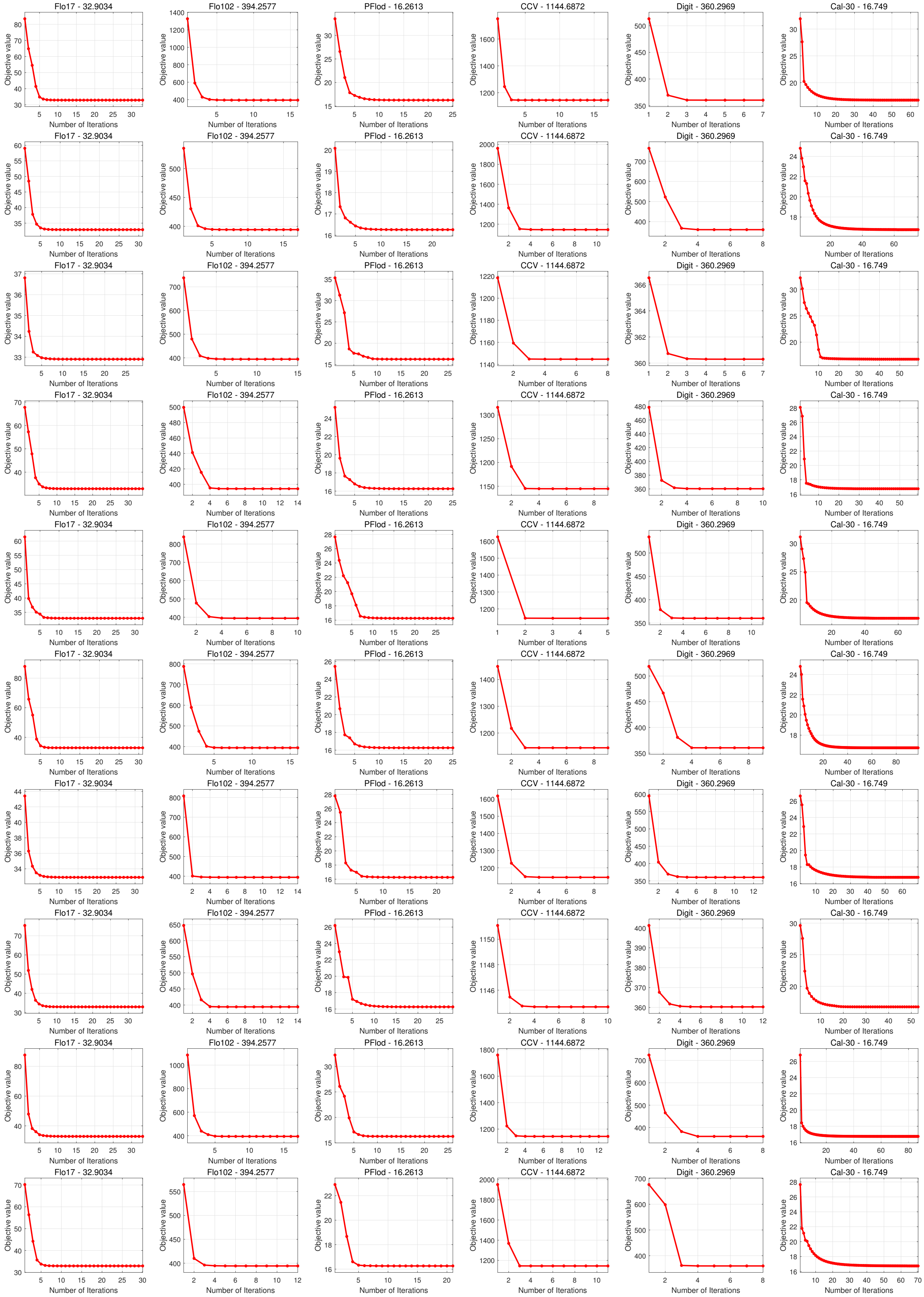

Fig. 6: The iterative objective curves of SimpleMKKM under ten different initialization on Flo17, Flo102, PFold, CCV, Digit, and Cal-30. Though with different initialization, the objective value at stopping point is the same.

8.3%, 11.3%, 12.1%, 13.2%, 12.9% and 14% with different number of samples for each cluster, respectively. It exceeds the newly developed MKKM variant, i.e., MKKM-MM [16], by 0.1%, 1.5%, 1.3%, 2.1%, 0.5% and 2%, respectively. We also observe that SimpleMKKM achieves comparable clustering performance with LF-MVC, which is considered as the SOTA in existing related clustering algorithm [32]. In sum, the proposed SimpleMKKM achieves the best clustering performance among MKKM based clustering algorithms, and is comparable with the strongest baseline among multi-view clustering in terms of four clustering metrics.

5.2.5 Clustering Performance with Variation of Base Kernels

To explore the ability of the proposed SimpleMKKM in dealing with different number of base kernels, we design an experiment on Caltech102 by selecting the first 8, 16, 24, 32, 40 and 48 base kernels. The clustering performance in terms of ACC, NMI, purity and rand index of the aforementioned methods varying with different number of base kernels are shown in Figure 4. As observed, we conclude that: i) The proposed SimpleMKKM demonstrates the overall best clustering performance among all compared ones regarding ACC, NMI, purity, and rand index. ii) With increasing the number of base kernels, the clustering performance of MKKM is dramatically decreased. In contrast, the clustering performance of SimpleMKKM is relatively stable with different number of base kernels, demonstrating its advantages in handling large number of base kernels. iii) The results in Figure 4 show that more base kernels are not necessarily helpful for improving clustering performance. In some applications, larger number of base kernels may result in worse clustering performance. This motivates us to automatically select a subset from a group of pre-specified base kernels and optimally combined the selected subset for multiple kernel clustering. This strategy could further significantly improve the clustering performance, which will be explored in our future work.

5.2.6 Kernel Weight Analysis

We next investigate the kernel weights learned by the compared algorithms. The results are plotted in Figure 5. We can see that the kernel weights learned by MKKM are extremely sparse on some datasets such as UCI-Digital, which is caused by the alternate optimization. This sparsity insufficiently exploits the multiple kernel matrices and explains the weak performance of MKKM. For example, the clustering accuracy of MKKM on UCI-Digital is only 47.2%. However, despite the ℓ_1-norm constraint on γ, the kernel weights learned by our SimpleMKKM are all 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.

5.2.7 Runtime and Global Convergence

We also report the running time of the compared algorithms in Figure 7. As observed, in addition to significantly improving performance, SimpleMKKM does not considerably increase the running time compared with MKKM and its variants.

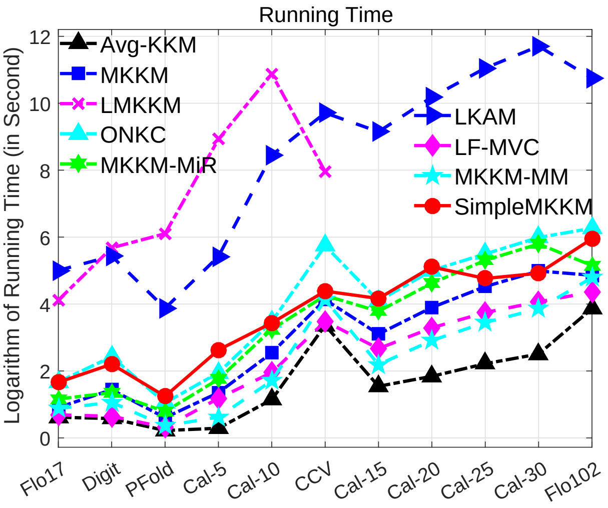

Fig. 7: Running time of different algorithms on 11 benchmark datasets (in second). The experiments are conducted on a PC with Intel(R) Core(TM)-i7-5820 3.3 GHz CPU and 32G RAM in MATLAB environment. SimpleMKKM is comparably fast to alternatives while providing superior performance and requiring no hyper-parameter tuning.

The objective of SimpleMKKM with iterations under ten different initialization is reported in Figure 6. From these figures, we observe that: 1) The objective monotonically decreases and the algorithm usually converges in less than thirty iterations on all datasets. This corroborates our earlier theoretical analysis of the nature of our proposed objective and efficient optimisation algorithm. 2) Though with different initialization, the objective value of SimpleMKKM at stopping point is the same. These experimental results are consistent with Theorem 2, and well validate that the solution obtained by our SimpleMKKM is the global optimum.

6 Conclusion

In this paper, we have extended the widely used supervised kernel alignment criterion to clustering, and introduce a novel clustering objective of by minimizing alignment for γ and maximizing it for $\mathbf{H}$. We show that this novel objective can be transformed into a minimization problem which is differentiable and amenable to a solution by reduced gradient descent. This makes SimpleMKKM unique among MKC alternatives, in not requiring a local-minimum prone alternate coordinate descent strategy.

We derive a generalization bound for our approach using global Rademacher complexity analysis. Comprehensive experiments demonstrate the effectiveness of SimpleMKKM. We expect that the simplicity, lack of hyper-parameters, and efficacy of SimpleMKKM will make it a go-to solution for practical multi-kernel clustering applications in future. Future work may aim to extend SimpleMKKM to handle incomplete kernels, study further applications, and derive convergence rates using local Rademacher complexity analysis [35], [36]. In addition, we plan to automatically select a subset from the majority of base kernels, and optimally combined them for multiple kernel clustering.

Acknowledgements

We thank Prof. Timothy M. Hospedales, Prof. Meng Wang and Prof. Junwei Han for their constructive comments on revising the manuscript. This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100), and the National Natural Science Foundation of China (project no. 61922088).

References

[1] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *SDM*, 2009, pp. 638–649.

[2] S. Yu, L.-C. Tranchevent, X. Liu, W. Glänzel, J. A. K. Suykens, B. D. Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE TPAMI*, vol. 34, no. 5, pp. 1031–1039, 2012.

[3] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[4] S. Sun, W. Dong, and Q. Liu, "Multi-view representation learning with deep gaussian processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

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

[6] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 1305–1313.

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

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

[9] A. Kumar, P. Rai, and H. Daumé, "Co-regularized multi-view spectral clustering," in *NIPS*, 2011, pp. 1413–1421.

[10] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1774–1782, 2019.

[11] X. Li, H. Zhang, R. Wang, and F. Nie, "Multi-view clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[12] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 2016, pp. 1704–1710.

[13] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, 2016, pp. 1888–1894.

[14] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel k-means with incomplete kernels," in *AAAI*, 2017, pp. 2259–2265.

[15] 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*, vol. 42, no. 5, pp. 1191–1204, May 2020.

[16] S. Bang, Y. Yu, and W. Wu, "Robust multiple kernel k-means clustering using min-max optimization," 2018.

[17] 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*, vol. 42, no. 6, pp. 1303–1316, 2020.

[18] 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*, vol. 41, no. 10, pp. 2410–2423, Oct 2019.

[19] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–13, 2020.

[20] X. Liu, S. Zhou, L. Liu, C. Tang, S. Wang, J. Liu, and Y. Zhang, "Localized simple multiple kernel k-means," in *ICCV*, 2021, pp. 9293–9301.

[21] X. Liu, "Incomplete multiple kernel alignment maximization for clustering," *IEEE Transactions on Pattern Analysis Machine Intelligence*, no. 01, pp. 1–14, sep 2021.

[22] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "lp-norm multiple kernel learning," *JMLR*, vol. 12, pp. 953–997, 2011.

[23] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," in *UAI*, 2009, pp. 109–116.

[24] ——, "Algorithms for learning kernels based on centered alignment," *JMLR*, vol. 13, pp. 795–828, 2012.

[25] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*, 2002.

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

[27] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131–159, Jan 2002.

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

[29] A. Maurer and M. Pontil, "k-dimensional coding schemes in Hilbert spaces," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5839–5846, 2010.

[30] T. Liu, D. Tao, and D. Xu, "Dimensionality-dependent generalization bounds for k-dimensional coding schemes," *Neural computation*, vol. 28, no. 10, pp. 2213–2249, 2016.

[31] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 2017, pp. 2266–2272.

[32] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2019, pp. 3778–3784.

[33] Y. Lu, L. Wang, J. Lu, J. Yang, and C. Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognition*, vol. 47, no. 11, pp. 3656–3664, 2014.

[34] G. Tzortzis and A. Likas, "Kernel-based weighted multi-view clustering," in *2012 IEEE 12th international conference on data mining*. IEEE, 2012, pp. 675–684.

[35] M. Kloft and G. Blanchard, "On the convergence rate of lp-norm multiple kernel learning," *J. Mach. Learn. Res.*, vol. 13, pp. 2465–2502, 2012.

[36] C. Cortes, M. Kloft, and M. Mohri, "Learning kernels using local rademacher complexity," in *Advances in Neural Information Processing Systems*, 2013, pp. 2760–2768.

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