Optimal Neighborhood Multiple Kernel Clustering with Adaptive Local Kernels

Jiyuan Liu, Xinwang Liu*, *Senior Member, IEEE,*
Jian Xiong, Qing Liao, Sihang Zhou, Siwei Wang and Yuexiang Yang*

Abstract—Multiple kernel clustering (MKC) algorithm aims to group data into different categories by optimally integrating information from a group of pre-specified kernels. Though demonstrating superiorities in various applications, we observe that existing MKC algorithms usually *do not sufficiently consider the local density around individual data samples* and *excessively limit the representation capacity of the learned optimal kernel*, leading to unsatisfying performance. In this paper, we propose an algorithm, called optimal neighborhood MKC with adaptive local kernels (ON-ALK), to address the two issues. In specific, we construct adaptive local kernels to sufficiently consider the local density around individual data samples, where different numbers of neighbors are discriminatingly selected on each sample. Further, the proposed ON-ALK algorithm boosts the representation of the learned optimal kernel via relaxing it into the neighborhood area of weighted combination of the pre-specified kernels. To solve the resultant optimization problem, a three-step iterative algorithm is designed and theoretically proven to be convergent. After that, we also study the generalization bound of the proposed algorithm. Extensive experiments have been conducted to evaluate the clustering performance. As indicated, the algorithm significantly outperforms state-of-the-art methods in recent literatures on six challenging benchmark datasets, verifying its advantages and effectiveness.

Index Terms—Multiple kernel clustering, Kernel alignment, Kernel k-mean.

1 Introduction

Kernel clustering has been widely explored in current machine learning and data mining literatures. It implicitly maps the original non-separable data into a high-dimensional Hilbert space where corresponding vertices have a clear decision boundary. Then, various clustering methods, including k-means [1], [2], fuzzy c-means [3], spectral clustering [4] and Gaussian Mixture Model (GMM) [5], are applied to group the unlabeled data into categories. Although kernel clustering algorithms have achieved great success in a large volume of applications, they are only able to handle data with a single kernel. Meanwhile, kernel functions are of different types, such as Polynomial, Gaussian, Linear, etc., and parameterized manually. How to choose the right kernel function and pre-define its parameters optimally for a specific clustering task is still an open problem. Nevertheless, sample features are collected from different sources in most practical settings. For example, news is reported by multiple news organizations; a person can be described from its fingerprint, palm veins, palm print, DNA , etc. The most common approach is to concatenate all features into one vector. But it ignores the fact that the features may not be directly comparable.

Multiple kernel clustering (MKC) algorithms, which utilize the complementary information from the pre-specified kernels, are well studied in literatures to address the aforementioned issues and can be roughly grouped into three categories. Methods in the first category intend to construct a consensus kernel for clustering by integrating low-rank optimization [6], [7], [8], [9], [10], [11]. For instance, Zhou et al. firstly recover a shared low-rank matrix from transition probability matrices of multiple kernels, and then use it as input to the standard Markov chain method for clustering [10]. Techniques in the second category compute their clustering results with the partition matrices generated from each individual kernel. Liu et al. firstly perform kernel k-means on each incomplete view and then explore the complementary information among all incomplete clustering results to obtain a final solution [12]. On the contrary, algorithms of the third category build the consensus kernel along with the clustering process. Most of them take the basic assumption that the optimal kernel is able to be represented as a weighted combination of pre-specified kernels. Huang et al. extend the fuzzy c-means by incorporating multiple kernels and automatically adjusting the kernel weights, which makes the clustering algorithm more immune to ineffective kernels and irrelevant features [13]. They also show multiple kernel k-means to be a special case of multiple kernel fuzzy c-means. The weighted combination assumption is also applied in spectral clustering, such as [14], [15]. Similarly, Yu et al. optimize the kernel weights based on the same Rayleigh quotient objective and claim their algorithm is of lower complexity [16]. Apart from this, various regularizations are formulated to help constrain the kernel weights and affinity matrix. For example, Du et al. use the $\mathcal{L}_{2,1}$-norm in the original feature space to minimize the reconstruction error [17]. Liu et al. propose Matrix-induced regularization to prevent from highly imbalanced

- *J. Liu, X. Liu, S. Zhou, S. Wang and Y. Yang are with the College of Computer, National University of Defense Technology, Changsha 410073, China. E-mail: {liujiyuan13, xinwangliu, yyx}@nudt.edu.cn*
- *J. Xiong is with School of Business Administration, Southwestern University of Finance and Economics, Chengdu, Sichuan, 611130, China.*
- *Q. Liao is with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, 518055, China.*
- *Corresponding author: Xinwang Liu and Yuexiang Yang.*

Manuscript received December 8, 2019.

weight assignment so as to reduce the mutual redundancy among kernels and enhance the diversity of selected kernels [18]. Zhao et al. assume each pre-specified kernel is constructed by the consensus matrix and a transition probability matrix [19]. They regularize the two types of matrices to be low-rank and sparse respectively. Liu et al. deal with incomplete kernels and propose a mutual kernel completion term to compute the missing items in kernels and learn the kernel weights simultaneously [20]. Instead of assuming the equality of samples in one kernel, some researches perform clustering with assigning different weights to samples, such as [21], [22].

Kernel alignment is an effective regularization in multiple kernel k-means algorithms [23], [24], [25]. However, Li et al. claim kernel alignment forces all sample pairs equally aligned with the same ideal similarity, conflicting with the well-established concept that aligning two farther samples with a low similarity is less reliable in high dimensional space [26]. Observing the local kernel trick in [27] can better capture sample-specific characteristics of the data, they use neighbors of each sample to construct local kernels and maximize the sum of their alignments with the ideal similarity matrix [26]. Additionally, the local kernel is demonstrated to be capable of helping the clustering algorithm better use the information provided by closer sample pairs [28].

The aforementioned MKC algorithms suffer from two facts, *not sufficiently considering the local density around individual data samples* and *excessively limiting the representation capacity of the learned optimal kernel*. In specific, the local kernel in [26] globally sets the number of neighbors for each sample to a constant, which cannot guarantee all sample pairs in the local kernel relatively close. It is known that performing alignment with farther sample pairs is less reliable. Therefore, this local kernel cannot reduce the unreliability to a minimum due to overlooking the local density around individual data samples. At the same time, most MKC algorithms assume that the optimal kernel is a weighted combination of pre-specified kernels, but ignore some more robust kernels in the complement set of kernel combinations. To address the two issues, we propose a MKC algorithm, called optimal neighborhood multiple kernel clustering with adaptive local kernels. Specifically, we design the adaptive local kernel which is constructed by selecting different number of neighbors whose similarities between each other are lower bounded by a pre-defined threshold. The constructed adaptive local kernels are then applied in MKC model. Meanwhile, the algorithm relaxes the rigid constraint of learning the optimal kernel from combinations of pre-specified kernels into their neighborhood areas. We also design an iterative algorithm to solve the resultant optimization problem. Our experiments show a competitive edge over state-of-the-art clustering algorithms on various datasets. The main contributions of this paper are highlighted as follows:

- In order to address the two issues in current MKC algorithms, *not sufficiently considering the local density around individual data samples* and *excessively limiting the representation capacity of the learned optimal kernel*, we design the *adaptive local kernel*, and locate the optimal kernel from the neighborhood area of linear combinations of pre-specified kernels. Then, the both techniques are utilized into a single multiple kernel clustering framework.
- We derive an algorithm, named optimal neighborhood multiple kernel clustering with adaptive local kernels, and study its generalization bound. Nevertheless, a three-step iterative algorithm is designed to solve the resultant optimization problem and we prove its convergence.
- Generalization ability of the proposed algorithm is well studied, and the generalization bound is proven to be $\mathcal{O}(\sqrt{1/n})$.
- Comprehensive experiments on six challenging benchmark datasets are conducted to validate the effectiveness of the proposed algorithm. As demonstrated, the proposed algorithm outperforms state-of-the-art clustering methods in recent literatures.

The rest of this paper is organized as follows: Section 2 presents a review of related work. Section 3 is devoted to the proposed ON-ALK algorithm. Section 4 explores its generation ability. Extensive experiments are conducted in section 5 to support our claims. We make some discussions and introduce the potential future work in Section 6, and finish the paper with conclusion in section 7.

2 Related work

In this section, we introduce some related work, including kernel k-means, multiple kernel k-means and regularized multiple kernel k-means.

2.1 Kernel k-means

Given a feature space $\mathcal{X}$ and a collection of n samples $\{x_i\}_{i=1}^n$, The feature map $\varphi(\cdot) : \mathcal{X} \to \mathcal{H}$ is denoted to map $\mathcal{X}$ into a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ [29], such that for any $x \in \mathcal{X}$ we have $\phi = \varphi(x)$. k-means algorithm aims to partition the samples into k disjoint clusters with each characterized by its centroid $\mathbf{c}_j$. The sample set associated with centroid $\mathbf{c}_j$ is defined as $\mathcal{C}_j = \{i \mid j = arg\min_{s=[k]} \|\phi_i - \mathbf{c}_s\|\}$, or in other words a point ϕ_i belongs to the j-th cluster if $\mathbf{c}_j$ is its closest centroid. The k-means objective is presented as

$$\frac{1}{n}\sum_{i=1}^{n}\min_{j=1,\cdot,k}\|\phi_i - \mathbf{c}_j\|^2. \tag{1}$$

With the cluster assignment matrix $\mathbf{Z} \in \{0,1\}^{n\times k}$ where $\mathbf{Z}_{ij} = 1$ if $i \in \mathcal{C}_j$. The objective can be written as

$$\min_{\mathbf{Z}} \frac{1}{n}\sum_{i=1,j=1}^{n,k}\mathbf{Z}_{ij}\|\phi_i - \mathbf{c}_j\|^2 \tag{2}$$

in which $\sum_{j=1}^{k}\mathbf{Z}_{ij} = 1$ and $\sum_{i=1}^{n}\mathbf{Z}_{ij} = |\mathcal{C}_j|$.

In most cases, $\phi \in \mathbb{R}^D$ with $D \gg n$ or even infinite and the feature map $\varphi(\cdot)$ is implicit. Therefore, it is difficult to directly apply k-means on the produced RKHS. By using the kernel trick where $\mathcal{K}(x, x') = \phi^T\phi'$ in which the kernel function $\mathcal{K}(\cdot,\cdot)$ can be explicitly given, the kernel matrix $\mathbf{K}$ is calculated. By expanding the quadratic item in Eq. (2), the objective is formated as

$$\min_{\mathbf{Z}} \mathrm{Tr}(\mathbf{K}) - \mathrm{Tr}\left(\mathbf{L}^{\frac{1}{2}}\mathbf{Z}^\top\mathbf{K}\mathbf{Z}\mathbf{L}^{\frac{1}{2}}\right) \tag{3}$$

where $\mathbf{L} = diag(\left[|\mathcal{C}_1|^{-1}, |\mathcal{C}_2|^{-1}, \cdots, |\mathcal{C}_k|^{-1}\right])$ and $\mathrm{Tr}(\cdot)$ represents the matrix trace. Observing the discrete $\mathbf{Z}$ makes optimization problem hard to solve, a common approach is to relax $\mathbf{Z}$ to take real values. Specially, by defining $\mathbf{H} = \mathbf{Z}\mathbf{L}^{\frac{1}{2}}$ and letting $\mathbf{H}$ take real values, a relaxed version of the above problem can be obtained as

$$\begin{aligned} \min_{\mathbf{H}} \ & \mathrm{Tr}\left(\mathbf{K}\left(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top\right)\right) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k. \end{aligned} \tag{4}$$

With the obtained $\mathbf{H}$, k-means is applied to compute the discrete cluster assignments.

2.2 Multiple kernel k-means

In multiple kernel settings, two circumstances in section 1, i.e. multiple kernel fuctions on single view data and multiple view data, are considered. Given $\{x_i\}_{i=1}^n \subseteq \mathcal{X}$ as a collection of n samples in single view, and $\{\{x_i^{(p)}\}_{i=1}^n\}_{p=1}^m \subseteq \{\mathcal{X}^{(p)}\}_{p=1}^m$ as m views' data with each consisting of n samples, $\phi_p(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}_p$ is the p-th feature map that maps x onto a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_p$ $(1 \le p \le m)$. Each sample in $\mathcal{H}_p$ is represented as $\varphi_p(x_i)$ or $\varphi_p(x_i^{(p)})$ which can be writen in an unified form, i.e. $\phi_i^{(p)}$.

Current literatures assign the feature maps with different weights into $\boldsymbol{\phi}_{\boldsymbol{\beta},i} = [\sqrt{\beta_1}\phi_i^{(1)}, \sqrt{\beta_2}\phi_i^{(2)}, \cdots, \sqrt{\beta_m}\phi_i^{(m)}]^T$, where the kernel function can be expressed as

$$\mathcal{K}_{\boldsymbol{\beta}}(x_i, x_j) = \phi_{\boldsymbol{\beta},i}^\top \phi_{\boldsymbol{\beta},j} = \sum\nolimits_{p=1}^m \beta_p \mathcal{K}_p(x_i^{(p)}, x_j^{(p)}). \tag{5}$$

With imputed samples and pre-specified kernel functions, the weighted combination of kernel matrices is computed as

$$\mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^m \beta_p \mathbf{K}_p. \tag{6}$$

Applying the hybrid kernel matrix $\mathbf{K}_{\boldsymbol{\beta}}$ into Eq. (4), the objective of multiple kernel k-means is formatted as

$$\begin{aligned} \min_{\mathbf{H},\boldsymbol{\beta}} \ & \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ & \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \ge 0, \ \forall p. \end{aligned} \tag{7}$$

in which $\mathbf{I}_k$ is an identity matrix of size $k \times k$.

The optimization problem is able to be solved by alternatively updating $\mathbf{H}$ and $\boldsymbol{\beta}$. Fixing $\boldsymbol{\beta}$, $\mathbf{H}$ can be obtained by solving a kernel k-means clustering objective shown in Eq. (4), while, with given $\mathbf{H}$, $\boldsymbol{\beta}$ can be optimized via solving

$$\begin{aligned} \min_{\boldsymbol{\beta}} \ & \sum_{p=1}^m \beta_p \mathrm{Tr}(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ & \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \ge 0, \ \forall p. \end{aligned} \tag{8}$$

Nevertheless, some literatures also adopt $\mathcal{L}_2$-norm combination to construct the optimal kernel, rather than the linear combination of the pre-specified kernels in Eq. (6) [30], [31].

2.3 Regularized multiple kernel k-means

Multiple kernel k-means in Eq. (7) achieves a promising clustering performance, but it fails to sufficiently consider the relationships among pre-specified kernels. Several regularizations are proposed to address this issue. Given a set of kernel matrices $\{\mathbf{K}_p\}_{p=1}^m$, a constant matrix $\mathbf{M}$ is defined as $\mathbf{M}_{pq} = \mathrm{Tr}(\mathbf{K}_p^\top \mathbf{K}_q)$. Ivano et al. notice that a small set of base kernels may contain a large quantity of helpful clustering information, and design Spectral Ratio (SR) regularization, i.e. Eq. (9), to allocate big weights to those kernels but leave the others close to zeros [32].

$$\max_{\boldsymbol{\beta}} \boldsymbol{\beta}^\top \mathbf{M}\boldsymbol{\beta} \quad s.t. \ \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \ge 0, \ \forall p. \tag{9}$$

Meanwhile, Liu et al. find similar kernels consist of less complementary clustering information but kernels of large differences contain more. Therefore, they propose the Matrix-induced regularization, as shown in Eq. (10), to reduce the redundancy of similar kernels by assigning relatively small weights to them [18].

$$\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^\top \mathbf{M}\boldsymbol{\beta} \quad s.t. \ \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \ge 0, \ \forall p. \tag{10}$$

The both regularizations have their own application settings, and we regularize kernel relationships via the Matrix-induced item in this paper. The regularized multiple kernel k-means model corresponding to Eq. (7) is formulated as

$$\begin{aligned} \min_{\mathbf{H},\boldsymbol{\beta}} \ & \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) + \lambda\boldsymbol{\beta}^\top \mathbf{M}\boldsymbol{\beta} \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ & \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \ge 0, \ \forall p. \end{aligned} \tag{11}$$

3 The Proposed Algorithm

In this section, the proposed optimal neighborhood multiple kernel clustering with adaptive local kernels (ON-ALK) algorithm is derived. Then, a three-step iterative algorithm is further designed to solve the resultant optimization problem. Finally, convergence of the iterative algorithm is proven and its computation complexity is analyzed.

3.1 The proposed formula

To address the aforementioned issues, i.e. insufficient consideration of sample similarity variances and limited representation capacity of the learned optimal kernel, we correspondingly propose the *adaptive local kernel* and utilized it with *optimal neighborhood kernel* [33] into a single MKC framework.

Current multiple kernel clustering algorithms indiscriminatingly forces all samples aligning with the ideal affinity matrix, i.e. $\mathbf{H}\mathbf{H}^\top$ in Eq. (11), without considering the non-negligible unreliability from the alignment of farther samples with low similarities. By observing this drawback, we construct the adaptive local kernel, which is a sub-matrix of kernel and reflects the relationships between a sample and its neighbors. Firstly, a threshold ζ is defined and the corresponding index set $\Omega^{(i)}$ for i-th sample can be written as

$$\Omega^{(i)} = \{j \mid \mathbf{K}(i,j) \ge \zeta\}. \tag{12}$$

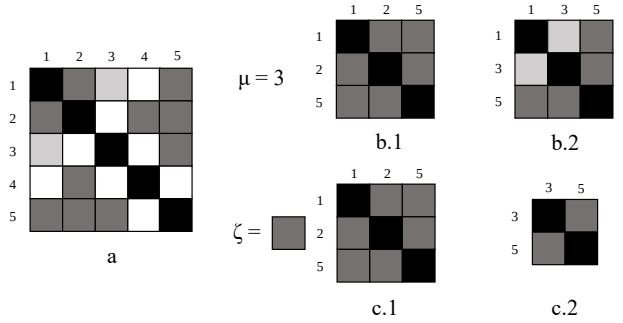

Fig. 1: Local kernel comparison: The darkness of boxes indicates similarity degree between sample pairs. The darker a box is, the more similar the corresponding sample pair is. Fig. (a) is the original kernel matrix. Fig. (b.1) and (b.2) are the local kernel generated in [26] corresponding to 1/3-th sample. Its size, μ, is fixed to 3; Fig. (c.1) and (c.2) are the proposed adaptive local kernels corresponding to 1/3-th sample. The similarities with its neighbors are higher than ζ.

Then, the corresponding indicator matrix $\mathbf{S}^{(i)} \in \{0,1\}^{n\times\mu^{(i)}}$ $s.t.$ $\mu^{(i)} = length(\Omega^{(i)})$ is defined

$$\mathbf{S}^{(i)}(i',j') = \begin{cases} 1 & s.t.\ i' \in \Omega^{(i)}, j' \textit{ is the index of } i' \textit{ in } \Omega^{(i)} \\ 0 & \textit{otherwise} \end{cases} \tag{13}$$

The i-th adaptive local kernel of matrix $\mathbf{K}$ can be written as

$$\mathbf{K}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{K}\mathbf{S}^{(i)} \in \mathbb{R}^{\mu^{(i)}\times\mu^{(i)}}. \tag{14}$$

In other words, Eq. (14) selects $\mu^{(i)}$ neighbor samples whose kernel values corresponding to i-th sample are larger than ζ and removes the others. Using the constructed local kernels in multiple kernel k-means and setting the trade-off on Matrix-induced regularization λ to 1, Eq. (11) can be rewritten as where .

$$\min_{\mathbf{H},\boldsymbol{\beta}} \ \frac{1}{n}\sum_{i=1}^{n}[\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}^{(i)}(\mathbf{I}_{\mu^{(i)}} - \mathbf{H}^{(i)}\mathbf{H}^{(i)\top})) \ + \ \boldsymbol{\beta}^\top\mathbf{M}^{(i)}\boldsymbol{\beta}] \tag{15}$$

in which $\mathbf{K}_{\boldsymbol{\beta}}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{K}_{\boldsymbol{\beta}}\mathbf{S}^{(i)}$, $\mathbf{M}_{pq}^{(i)} = \mathrm{Tr}(\mathbf{K}_p^{(i)}\mathbf{K}_q^{(i)})$, $\mathbf{H}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{H}$, $\mathbf{I}_{\mu^{(i)}}$ is the identity matrix of size $\mu^{(i)}$ and $\mu^{(i)}$ varies with the density around samples.

The proposed *adaptive local kernel* is extended from the local kernel in [26] which requires the sizes of local kernels to a constant indiscriminatingly. However, it cannot guarantee all sample pairs in one local kernel of high similarity. On the contrary, we construct the i-th adaptive local kernel by selecting the samples whose similarities to sample i are higher than a threshold, ζ. Fig. 1 compares the two types of local kernel comprehensively. It can be seen that the local kernels generated in [26] are of the same size, while the proposed adaptive local kernels are constructed from similarities of sample pairs. Comparing b.1/2 and c.1/2 in Fig. 1, we can notice the proposed *adaptive local kernel* is usually smaller than the local kernel in [26], guaranteeing all neighbors of relatively high similarities and reducing the unreliabilities from aligning farther sample pairs.

Eq. (15) can be regarded as two parts. The first item represents the loss sum of multiple kernel clustering with local kernels, while the second item can be seen as a regularization to balance the weight assignment, enhancing the diversity of selected kernels. It assumes the linear combination of pre-specified kernels, i.e. $\mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^{m}\beta_p\mathbf{K}_p$, as the optimal kernel. However, this excessively limits representativity of the optimal kernel, preventing from finding a more robust kernel in the complement set of kernel combinations. Therefore, we assume that the optimal kernel, termed as $\mathbf{J}$, resides in the neighborhood of kernel combinations, presented as

$$\mathcal{N} = \{\mathbf{J} \mid \|\mathbf{J} - \mathbf{K}_{\boldsymbol{\beta}}\|_F^2 \le \theta, \ \mathbf{J} \succeq 0\}. \tag{16}$$

The assumption is further applied in Eq. (15) , contributing to

$$\begin{aligned} \min_{\mathbf{H},\boldsymbol{\beta},\mathbf{J}} \ & \frac{1}{n}\sum_{i=1}^{n}[\mathrm{Tr}(\mathbf{J}^{(i)}(\mathbf{I}_{\mu^{(i)}} - \mathbf{H}^{(i)}\mathbf{H}^{(i)\top})) \ + \ \boldsymbol{\beta}^\top\mathbf{M}^{(i)}\boldsymbol{\beta}] \\ s.t. \ & \mathbf{J} \in \mathcal{N}. \end{aligned} \tag{17}$$

The objective in Eq. (17) is difficult to optimize, due to the constraints on $\mathbf{J}$. Observing that $\mathbf{K}_{\boldsymbol{\beta}}$ provides the prior knowledge for clustering, $\mathbf{J}$ is more likely to reach its optimality with a closer gap between $\mathbf{K}_{\boldsymbol{\beta}}$. Instead of setting the maximal gap, θ, explicitly, we learn the real gap along with clustering process, which formulates our final objective as

$$\begin{aligned} \min_{\mathbf{H},\boldsymbol{\beta},\mathbf{J}} \ & \frac{1}{n}\sum_{i=1}^{n}[\mathrm{Tr}(\mathbf{J}^{(i)}(\mathbf{I}_{\mu^{(i)}} - \mathbf{H}^{(i)}\mathbf{H}^{(i)\top})) \ + \ \boldsymbol{\beta}^\top\mathbf{M}^{(i)}\boldsymbol{\beta}] \\ & + \frac{\rho}{2}\|\mathbf{J} - \mathbf{K}_{\boldsymbol{\beta}}\|_F^2 \\ s.t. \ & \mathbf{H} \in R^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \ \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \\ & \beta_p \ge 0, \ \forall p, \ \mathbf{J} \succeq 0. \end{aligned} \tag{18}$$

where $\mathbf{M}_{pq}^{(i)} = \mathrm{Tr}(\mathbf{K}_p^{(i)}\mathbf{K}_q^{(i)})$, $\mathbf{K}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{K}\mathbf{S}^{(i)}$, $\mathbf{J}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{J}\mathbf{S}^{(i)}$, $\mathbf{S}^{(i)} \in \{0,1\}^{n\times\mu^{(i)}}$ indicates the $\mu^{(i)}$-nearest neighbors of i-th sample where n is the number of all samples, $\mathbf{I}_{\mu^{(i)}}$ is the identity matrix of size $\mu^{(i)}$. The optimal kernel, $\mathbf{J}$, serves as a bridge to connect the clustering process, i.e. the first term of Eq. (18), with the knowledge obtaining process, i.e. the last item of Eq. (18). In this circumstance, it explores the complementary information in pre-specified kernels to help with clustering process, and uses the information from clustering to help with weight assignment of pre-specified kernels as a feedback.

3.2 Alternate optimization

In order to solve the objective in Eq. (18), we carefully design a three-step iterative algorithm, in which two variables are fixed while optimizing the other one.

i) **Optimizing H with fixed J and $\boldsymbol{\beta}$.** Given $\mathbf{J}$ and $\boldsymbol{\beta}$, $\sum_{i=1}^{n}\boldsymbol{\beta}^\top\mathbf{M}^{(i)}\boldsymbol{\beta}$ and $\|\mathbf{J} - \mathbf{K}_{\boldsymbol{\beta}}\|_F^2$ are constants and Eq. (18) can be reduced as follows.

$$\begin{aligned} \min_{\mathbf{H}} \ & \frac{1}{n}\sum_{i=1}^{n}\mathrm{Tr}(\mathbf{J}^{(i)}(\mathbf{I}_{\mu^{(i)}} - \mathbf{H}^{(i)}\mathbf{H}^{(i)\top})) \\ s.t. \ & \mathbf{H} \in R^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \ \mathbf{H}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{H} \\ & \mathbf{J}^{(i)} = \mathbf{S}^{(i)\top}\mathbf{J}\mathbf{S}^{(i)}. \end{aligned} \tag{19}$$

With defining $\mathbf{A}^{(i)} = \mathbf{S}^{(i)}\mathbf{S}^{(i)\top}$, it can be further transformed to

$$\begin{aligned} \min_{\mathbf{H}} \ & \mathrm{Tr}[(\frac{1}{n}\sum_{i=1}^{n}\mathbf{A}^{(i)}\mathbf{J}\mathbf{A}^{(i)})(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)] \\ s.t. \ & \mathbf{H} \in R^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k. \end{aligned} \tag{20}$$

As seen from ***Proposition 1***, $\sum_{i=1}^{n}\mathbf{A}^{(i)}\mathbf{J}\mathbf{A}^{(i)}$ is a positive semi-definite matrix. Therefore, Eq. (20) is a standard kernel k-means problem which can be efficiently solved by off-the-shelf packages.

Proposition 1. Given a kernel matrix, $\mathbf{J} \in \mathbb{R}^{n\times n}$, and a set of sparse matrices, $\mathcal{S} = \{\mathbf{S}^{(i)}\}_{i=1}^{n}$ where $\mathbf{S}^{(i)}$ is defined in Eq. (13), the matrix $\sum_{i=1}^{n}\mathbf{A}^{(i)}\mathbf{J}\mathbf{A}^{(i)}$, in which $\mathbf{A}^{(i)} = \mathbf{S}^{(i)}\mathbf{S}^{(i)\top}$, is a positive semi-definite matrix.

Proof. *For such* $\mathbf{S}^{(i)}$, $\mathbf{A}^{(i)}$ *is a diagonal matrix sized* n *with* $\mu^{(i)}$ *ones. Donating* $\mathcal{P}^{(i)} = \{i' \mid \mathbf{A}^{(i)}_{i'i'} = 1\}$, $\mathbf{A}^{(i)}\mathbf{J}\mathbf{A}^{(i)}$ *is a* $n \times n$ *matrix* $\mathbf{G}^{(i)}$ *with* $\mathbf{G}^{(i)}_{i'j'} = \mathbf{J}_{i'j'}$ *for* $\{i', j'\} \in \mathcal{P}^{(i)}$ *and* $\mathbf{G}^{(i)}_{i'j'} = 0$ *for* $\{i', j'\} \notin \mathcal{P}^{(i)}$. *In such setting,* $\mathbf{G}^{(i)}_{i'j'}$ *for* $\{i', j'\} \in \mathcal{P}^{(i)}$ *is a smaller kernel matrix of sample pairs indexed in* $\mathcal{P}^{(i)}$*, whose eigenvalues are greater than or equal to* 0*, demonstrating that* $\mathbf{G}^{(i)}$ *is a positive semi-definite matrix.* $\sum_{i=1}^{n}\mathbf{A}^{(i)}\mathbf{J}\mathbf{A}^{(i)}$ *is the sum of* $\mathbf{G}^{(i)}$ *for* $i = \{1, 2, \cdots, n\}$*, therefore contributes to a positive semi-definite matrix.*

ii) **Optimizing J with fixed $\boldsymbol{\beta}$ and H**. With given $\boldsymbol{\beta}$ and $\mathbf{H}$, the optimization problem can be rewritten as

$$\begin{aligned} \min_{\mathbf{J}} \ & \frac{1}{n}\sum_{i=1}^{n}\mathrm{Tr}(\mathbf{J}^{(i)}(\mathbf{I}_{\mu^{(i)}} - \mathbf{H}^{(i)}\mathbf{H}^{(i)\top})) + \frac{\rho}{2}\left\|\mathbf{J} - \mathbf{K}_{\boldsymbol{\beta}}\right\|_F^2 \\ s.t. \ & \mathbf{J} \succeq 0. \end{aligned} \tag{21}$$

which, by defining $\mathbf{G} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{S}^{(i)}\mathbf{S}^{(i)\top}) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{A}^{(i)}$, can be further transformed into

$$\begin{aligned} \min_{\mathbf{J}} \ & \left\|\mathbf{J} - \mathbf{B}\right\|_F^2 \\ s.t. \ & \mathbf{B} = \mathbf{K}_{\boldsymbol{\beta}} - \frac{1}{\rho}\mathbf{G}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\mathbf{G}, \\ & \mathbf{J} \succeq 0. \end{aligned} \tag{22}$$

Eq. (22) can be solved by finding the projection of $\mathbf{B}$ in PSD space. It is claimed in Theorem 2 of [10] that the optimization is able to be written as $\mathbf{J} = \mathbf{U}_B\boldsymbol{\Sigma}_{\boldsymbol{B}}^{+}\mathbf{V}_B^{+}$, where $\mathbf{B} = \mathbf{U}_B\boldsymbol{\Sigma}_{\boldsymbol{B}}\mathbf{V}_B^\top$ is the singular value decomposition (SVD) and $\boldsymbol{\Sigma}_{\boldsymbol{B}}{}^{+}$ is a diagonal matrix which keeps the non-negative values of $\boldsymbol{\Sigma}_{\boldsymbol{B}}$.

iii) **Optimizing $\boldsymbol{\beta}$ with fixed J and H**. With given $\mathbf{J}$ and $\mathbf{H}$, the optimization problem can be reduced into

$$\begin{aligned} \min_{\boldsymbol{\beta}} \ & \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\beta}^\top\mathbf{M}^{(i)}\boldsymbol{\beta} + \frac{\rho}{2}\left\|\mathbf{J} - \mathbf{K}_{\boldsymbol{\beta}}\right\|_F^2 \\ s.t. \ & \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \ \beta_p \geq 0, \ \forall p. \end{aligned} \tag{23}$$

which can be transformed into

$$\begin{aligned} \min_{\boldsymbol{\beta}} \ & \boldsymbol{\beta}^\top(\frac{1}{n}\sum_{i=1}^{n}\mathbf{M}^{(i)} + \frac{\rho}{2}\mathbf{M})\boldsymbol{\beta} + \boldsymbol{\alpha}^\top\boldsymbol{\beta} \\ s.t. \ & \mathbf{M}^{(i)}_{pq} = \mathrm{Tr}(\mathbf{K}^{(i)}_p\mathbf{K}^{(i)}_q), \ \mathbf{M}_{pq} = \mathrm{Tr}(\mathbf{K}_p\mathbf{K}_q), \\ & \boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_m], \ \alpha_p = -\rho\mathrm{Tr}(\mathbf{J}\mathbf{K}_p). \end{aligned} \tag{24}$$

Algorithm 1 optimal neighborhood multiple kernel clustering with adaptive local kernels

Input: $\{\mathbf{K}_p\}_{p=1}^{m}$
Parameters: ρ and ζ
Output: $\mathbf{H}$ and $\boldsymbol{\beta}$

```
Initialize β^(1) = 1_m/m, J^(1) = K_{β^(1)} and t = 1.
Generate S^(i) for i-th sample with K_{β^(1)} and the similarity threshold ζ.
while (obj^{t+1} − obj^t)/obj^t ≥ σ do
    K_{β^(t)} = Σ_{p=1}^{m} β_p^{(t−1)} K_p.
    Optimize H^(t+1) with J^(t) and β^(t) via Eq. (20).
    Optimize J^(t+1) with β^(t) and H^(t+1) via Eq. (22).
    Optimize β^(t+1) with J^(t+1) and H^(t+1) via Eq. (24).
    t = t + 1.
end while
return H^(t) and β^(t)
```

Obviously, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{M}^{(i)} + \frac{\rho}{2}\mathbf{M}$ can be proven positive semi-definite, for it is a linear positive combination of positive semi-definite matrices, i.e. $\mathbf{M}$ and $\{\mathbf{M}^{(i)}\}_{i=1}^{n}$, as seen in ***Proposition 2***. Therefore, Eq. (23) is a quadratic programming with linear constraints and can be sufficiently solved by off-the-shelf packages.

Proposition 2. Given a set of positive semi-definite matrices, $\mathcal{K} = \{\mathbf{K}\}_{i=1}^{m}$, the matrix, $\mathbf{M} \in \mathbb{R}^{m\times m}$, in which $\{\mathbf{M}_{pq} = \mathrm{Tr}(\mathbf{K}_p\mathbf{K}_q)\}_{p,q=1}^{m}$, is positive semi-definite.

Proof. *For any vector* $\mathbf{x} \in \mathbb{R}^{m\times 1}$, $\mathbf{x}^\top\mathbf{M}\mathbf{x} = \mathrm{Tr}(\mathbf{K}_{sum}^\top\mathbf{K}_{sum}) = \left\|\mathbf{K}_{sum}\right\|_f^2 \geq 0$, *where* $\mathbf{K}_{sum} = x_1\mathbf{K}_1 + x_2\mathbf{K}_2 + \cdots + x_m\mathbf{K}_m$, *illustrating matrix* $\mathbf{M}$ *is positive semi-definite.*

In summary, the proposed iterative algorithm to solve the resultant optimization problem in Eq. (18) is outlined in Algorithm 1. ρ is the trade-off between the clustering process and weight assigning process, while ζ controls the similarities among sample pairs in local kernels. The two parameters are supposed to be specified in advance. Nevertheless, σ is the stopping gap and should be set to a relatively small value.

3.3 Convergence and complexity

Convergence of the proposed iterative algorithm is theoretically guaranteed and clarified as following. For a brief expression, we represent the objective in Eq. (18) with $Obj(\mathbf{H}, \boldsymbol{\beta}, \mathbf{J})$. When the initialization is done at the beginning of the algorithm, the value of $Obj(\mathbf{H}, \boldsymbol{\beta}, \mathbf{J})$ is set to a definite value. Let $\mathbf{H}^{(t)}$, $\boldsymbol{\beta}^{(t)}$ and $\mathbf{J}^{(t)}$ be the solution at the t-th iteration.

i) **Optimizing H with fixed J and $\boldsymbol{\beta}$**. With the obtained solution written as $\mathbf{H}^{(t+1)}$, we have

$$Obj(\mathbf{H}^{(t+1)}, \boldsymbol{\beta}^{(t)}, \mathbf{J}^{(t)}) \ \leq \ Obj(\mathbf{H}^{(t)}, \boldsymbol{\beta}^{(t)}, \mathbf{J}^{(t)}). \tag{25}$$

ii) **Optimizing J with fixed $\boldsymbol{\beta}$ and H**. With the obtained solution written as $\mathbf{J}^{(t+1)}$, we have

$$Obj(\mathbf{H}^{(t+1)}, \boldsymbol{\beta}^{(t)}, \mathbf{J}^{(t+1)}) \ \leq \ Obj(\mathbf{H}^{(t+1)}, \boldsymbol{\beta}^{(t)}, \mathbf{J}^{(t)}). \tag{26}$$

iii) **Optimizing $\boldsymbol{\beta}$ with fixed $\mathbf{J}$ and $\mathbf{H}$**. With the obtained solution written as $\boldsymbol{\beta}^{(t+1)}$, we have

$$Obj(\mathbf{H}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \mathbf{J}^{(t+1)}) \ \leq \ Obj(\mathbf{H}^{(t+1)}, \boldsymbol{\beta}^{(t)}, \mathbf{J}^{(t+1)}). \tag{27}$$

Together with Eq. (25), (26) and (27), we have

$$Obj(\mathbf{H}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \mathbf{J}^{(t+1)}) \ \leq \ Obj(\mathbf{H}^{(t)}, \boldsymbol{\beta}^{(t)}, \mathbf{J}^{(t)}). \tag{28}$$

which indicates Eq. (18) monotonically decreases with iterations. At the same time, Eq. (18) is lower bounded by zero. Therefore, the iterative algorithm is theoretically guaranteed to converge.

While optimizing $\mathbf{H}$ in Eq. (20), $\mathbf{A}^{(i)}$ is a diagonal matrix sized n with $\mu^{(i)}$ ones, resulting in that $\sum_{i=1}^{n} \mathbf{A}^{(i)}\mathbf{J}\mathbf{A}^{(i)} = (\sum_{i=1}^{n} \mathbf{A}^{(i)}) \odot \mathbf{J}$ in which $\odot$ is the Hadamard product. Its complexity is $\mathcal{O}(n^2)$, and the complexity of standard kernel k-means is $\mathcal{O}(n^3)$, with the overall complexity $\mathcal{O}(n^3 + n^2)$. Similarly, computing $\mathbf{B}$ in Eq. (22) needs $\mathcal{O}(n^2)$. The SVD decomposition needs $\mathcal{O}(n^3)$. In sum, $\mathcal{O}(n^3 + n^2)$ is required to solve $\mathbf{J}$. In Eq. (24), the calculation of $\mathbf{M}$ needs $\mathcal{O}(m^2n^2)$, while $\mathcal{O}(m^2{\mu^{(i)}}^2)$ is required for $\mathbf{M}^{(i)}$. The resultant quadratic programming problem needs $\mathcal{O}(m^3)$. Overall, the complexity of solving $\boldsymbol{\beta}$ is $\mathcal{O}(m^2n^2 + m^2\sum_{i=1}^{n}{\mu^{(i)}}^2 + m^3)$. In our implementation, $\mathbf{M}$ and $\{\mathbf{M}^{(i)}\}_{i=1}^{n}$ are calculated in advance and keep the same during the whole optimization process. Therefore, the whole computation complexity is $\mathcal{O}(n^3)$. The proposed algorithm shares the same computation complexity with the comparative methods [7], [9], [10], [13], [17], [18], [18], [33], but achieves state-of-the-art performances as shown in Table 2, verifying its superiority.

4 Generalization Bound

In this section, we analyze the generalization bound of the proposed algorithm, and show how our objective contributes to a relatively lower bound. The proof details are provided in the supplementary material.

Generalization bound for k-means algorithms indicates how well the clustering centroids obtained in learning process perform in the test stage [34], [35]. With the combination weights $\boldsymbol{\beta}$ learned from the proposed model, the clustering centroids $\mathbf{C} \in \mathcal{H}^k$ can be found in the corresponding Hilbert space, where sample x of m views is mapped into $\phi_{\boldsymbol{\beta}}(x) = [\sqrt{\beta_1}\phi_1^\top(x^{(1)}), \sqrt{\beta_2}\phi_2^\top(x^{(2)}), \cdots, \sqrt{\beta_m}\phi_m^\top(x^{(m)})]^\top$. If data samples are given in batches, i.e. $\mathbf{x} = \{x_i\}_{i=1}^{l}$, the reconstruction error is supposed to be

$$\mathbb{E}\left[\frac{1}{l}\sum_{i=1}^{l}\min_{\mathbf{y}\in\{e_1,\ldots,e_k\}} a_i \left\|\phi_{\boldsymbol{\beta}}(x_i) - \mathbf{C}\mathbf{y}\right\|_{\mathcal{H}}^2\right], \tag{29}$$

in which $\{\mathbf{e}_1, \cdots, \mathbf{e}_k\}$ are the orthogonal bases of $\mathbb{R}^k$. a_i equals to the neighbor number of i-th sample. In other words, $a_i = |\mathbf{N}_i|$, where $\mathbf{N}_i = \{j \mid \phi_{\boldsymbol{\beta}}(x_i)^\top\phi_{\boldsymbol{\beta}}(x_j) \geq \zeta\}$. Eq. (29) is also compatible with testing on single sample via setting $l = 1$, reducing to

$$\mathbb{E}\left[\min_{\mathbf{y}\in\{e_1,\ldots,e_k\}} \left\|\phi_{\boldsymbol{\beta}}(x) - \mathbf{C}\mathbf{y}\right\|_{\mathcal{H}}^2\right]. \tag{30}$$

With removing the constraint, $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k$, in Eq. (18) of the manuscript, we can define a function class

$$\begin{aligned}\mathcal{F} = \{f : \mathbf{x} \mapsto \frac{1}{l}\sum_{i=1}^{l}\min_{\mathbf{y}\in\{e_1,\ldots,e_k\}} a_i \left\|\phi_{\boldsymbol{\beta}}(x_i) - \mathbf{C}\mathbf{y}\right\|_{\mathcal{H}}^2 \ \mid \\ \boldsymbol{\beta}^\top\mathbf{1} = 1, \beta_p \geq 0, \phi_p^\top(x_i^{(p)})\phi_p(x_j^{(p)}) \leq b, \\ \forall p \in \{1, \cdots, m\}, \forall x_i, x_j \in \mathcal{X}, \mathbf{C} \in \mathcal{H}^k\}.\end{aligned} \tag{31}$$

Theorem 1. *For the given samples of n batches and any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f \in \mathcal{F}$:*

$$\begin{aligned}\mathbb{E}[f(\mathbf{x})] \leq \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i) + \frac{2b\sqrt{2\pi}}{n\sqrt{l}}\left(\mathcal{G}_{1n} + \sqrt{2}\mathcal{G}_{2n} + \mathcal{G}_{3n}\right) \\ +4bN_{max}\sqrt{\frac{\ln 1/\delta}{2n}}.\end{aligned} \tag{32}$$

where

$$\begin{aligned}\mathcal{G}_{1n} &= \mathbb{E}_\gamma\left[\sum_{j=1}^{n}\sum_{i=1}^{l}\gamma_{ji}N_{max}\right], \\ \mathcal{G}_{2n} &= \mathbb{E}_\gamma\left[\sum_{j=1}^{n}\sum_{i=1}^{l}\sum_{r=1}^{k}\gamma_{jir}N_{max}\right], \\ \mathcal{G}_{3n} &= \mathbb{E}_\gamma\left[\sum_{j=1}^{n}\sum_{i=1}^{l}\sum_{r,s=1}^{k}\gamma_{jirs}N_{max}\right]\end{aligned} \tag{33}$$

and $N_{min} = \min_{j,i=1}^{n,l} a_i^{(j)}$, $N_{max} = \max_{j,i=1}^{n,l} a_i^{(j)}$, $\gamma_{ji}, \gamma_{jir}, \gamma_{jirs}, j \in \{1, \cdots, n\}, i \in \{1, \cdots, l\}, r, s \in \{1, \cdots, k\}$, are i.i.d. Gaussian random variables with zero mean and unit standard deviation.

The detailed proof of Theorem 1 can be found in the appendix. With further relaxation on Eq. (33), we have $\mathcal{G}_{1n} \leq N_{max}l\sqrt{n}$, $\mathcal{G}_{2n} \leq N_{max}lk\sqrt{n}$ and $\mathcal{G}_{3n} \leq N_{max}lk^2\sqrt{n}$, which implies the proposed algorithm have generalization bound of $\mathcal{O}(\sqrt{1/n})$. From the definition of $f(\mathbf{x})$ in Eq. (31), we have

$$\mathbb{E}\left[f(\mathbf{x})\right] = \mathbb{E}\left[\frac{1}{l}\sum_{i=1}^{l}\min_{\mathbf{y}\in\{e_1,\ldots,e_k\}} a_i \left\|\phi_{\boldsymbol{\beta}}(x_i) - \mathbf{C}\mathbf{y}\right\|_{\mathcal{H}}^2\right] \tag{34}$$

which is the reconstruction error expectation as shown in Eq. (29). According to **Theorem 1**, $\frac{1}{n}f(\mathbf{x}_i)$ is supposed to be as small as possible. For given samples, $\{x\}_{i=1}^{l}$, the following inequality holds

$$f(\mathbf{x}) \leq \min_{\mathbf{H},\,\boldsymbol{\beta}} \frac{1}{l}\sum_{i=1}^{l}\left[\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}^{(i)}(\mathbf{I}_{\mu^{(i)}} - \mathbf{H}^{(i)}\mathbf{H}^{(i)\top}))\right] \tag{35}$$

because the proposed algorithm takes a extra constraint, $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k$, resulting that the learned $\mathbf{C}$ and $\boldsymbol{\beta}$ are not optimal for $f(\mathbf{x})$. Therefore, minimizing the objective provides a small bound for $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$ and is advantageous to model generation.

5 EXPERIMENT

In this section, we conduct extensive experiments on various benchmark datasets to evaluate the proposed algorithm and compare its performances with typical MKC algorithms in recent literatures. In addition, we experimentally show that the performance is boosted from two aspects simultaneously, i.e. *adaptive local kernel* and *optimal neighborhood kernel*. Furthermore, parameter sensibility and convergence of the proposed algorithm is validated. Finally, we evaluate its performances on additional datasets with a new criterion, verifying its effectiveness sufficiently.

5.1 Datasets and comparative algorithms

Six benchmark datasets of various categories are employed to evaluate the effectiveness of the proposed algorithm. They are *Flower102*[1] [36], *Digital*[2] [37], *Caltech101*[3] [38], *Protein Fold*[4] [39], *Cornell*[5] [40] and *AR10P*[6] [41]. The corresponding kernel matrices are selected from recent multiple kernel clustering literatures [18], [33], [42], instead of generating by ourselves. This prevents from generating the kernel matrices, on which only our algorithm achieves good performances, in purpose. In specific, kernel matrices of *Protein Fold* are generated by following the technique in [43], in which the second order polynomial kernel and inner product (cosine) kernel are applied to the first ten feature sets and the last two feature sets, respectively. Meanwhile, the others are generated by applying RBF kernel on every feature. Furthermore, the detail information, including kernel size, kernel number and class number, is listed in Table 1.

TABLE 1: Specifications of used datasets.

Dataset	Number of			ML tasks
	Samples	Kernels	Clusters	
Flower102	8189	4	102	image clustering
Digital	2000	3	10	image clustering
Caltech101	1530	25	102	object detection
Protein Fold	694	12	27	medical research
Cornell	195	2	5	linguistics
AR10P	130	6	10	face recognition

Along with the proposed algorithm, we ran another ten typical MKC algorithms in recent literatures, i.e. Average Multiple Kernel k-means (AMKKM) by assigning pre-specified kernels with same weights uniformly, Single Best Kernel k-means (SBKKM), Multiple Kernel k-means (MKKM) [13], Robust Multiple Kernel k-means (RMKKM) [17], Co-regularized Spectral Clustering (CRSC) [7], Robust Multi-view Spectral Clustering (RMSC) [9], Robust Multiple Kernel Clustering (RMKC) [10], Multiple Kernel k-Means Clustering with Matrix-Induced Regularization (MKCMR) [18], Multiple kernel clustering with local kernel alignment maximization (LKAM) [26] and Optimal Neighborhood Kernel Clustering with Multiple Kernels (ONKC) [33]. We implement the codes of AMKKM, SBKKM and MKKM, while codes of the others are publicly available in authors' websites, and we adopt them directly.

1. http://www.robots.ox.ac.uk/~vgg/data/flowers/102/
2. http://ss.sysu.edu.cn/~py/
3. http://files.is.tue.mpg.de/pgehler/projects/iccv09/
4. http://mkl.ucsd.edu/dataset/protein-fold-prediction
5. http://lamda.nju.edu.cn/code_PVC.ashx
6. http://featureselection.asu.edu/

5.2 Experiment settings

At the initialization stage, the pre-specified kernel matrices are centered, for better performance can be achieved by using centered ones comparing with original ones, as claimed in [24]. Next, we perform normalization on them so as to better specify the range of similarity values between sample pairs into $[-1, 1]$. Kernel matrices and class numbers are assumed known in advance. Three well-established measurements, i.e. accuracy(ACC), normalized mutual information(NMI) and purity, are computed to evaluate the clustering performance.

There are two hyper-parameters, ρ and ζ. ρ is the trade-off between the *adaptive local kernel* and the *optimal neighborhood kernel*, and indicates which one is more important than the other one or they are equally weighted (the special case when $\rho = 2$). At the same time, ζ controls the similarity values between sample pairs in local kernel matrices by selecting closer sample vertices around i-th sample. Grid search technique is performed to select the two parameters, where ρ and ζ varies in $2.\hat{}[-15,\ -14 \cdots,\ 15]$ and $[-0.5,\ -0.4 \cdots,\ 0.5]$, respectively.

5.3 Experiment results

In the beginning, the proposed ON-ALK algorithm and the comparative ones are tested on six benchmark datasets. The ACC, NMI and purity are calculated and reported in Table 2, where the best results are marked in bold. We have the following observations:

1) The proposed algorithm holds the best results among the eleven algorithms in three measurements. Specifically in ACC, it exceeds the second-best algorithm by 3.88% on *Flower102*, 0.25% on *Digital*, 2.13% on *Caltech101*, 1.35% on *Protein Fold*, 3.59% on *Cornell* and 1.54% on *AR10P*. In NMI, it outperforms the second-best by 1.09% on *Flower102*, 0.6% on *Digital*, 1.28% on *Caltech101*, 0.88% on *Protein Fold*, 3.91% on *Cornell*, while drops back to the second with only 0.43% lower than SBKKM. In purity, it performs better than the other algorithms by 3.42% on *Flower102*, 0.25% on *Digital*, 2.81% on *Caltech101*, 3.02% on *Protein Fold*, 0.51% on *Cornell* and 1.54% on *AR10P*, respectively.
2) AMKKM and SBKKM, the baselines of multiple kernel clustering, outperform some other recently proposed algorithm. However, the proposed algorithm has a consistently and notably better performance over the two algorithms. For instance, it exceeds AMKKM by 18.15%, 2.48%, 7.55%, 9.94%, 8.2% and 6.16% on six datasets in ACC.

We also investigate the effect of class number on clustering performance via conducting experiments with big datasets, such as *Flower102*. Figure 2 presents the ACC,

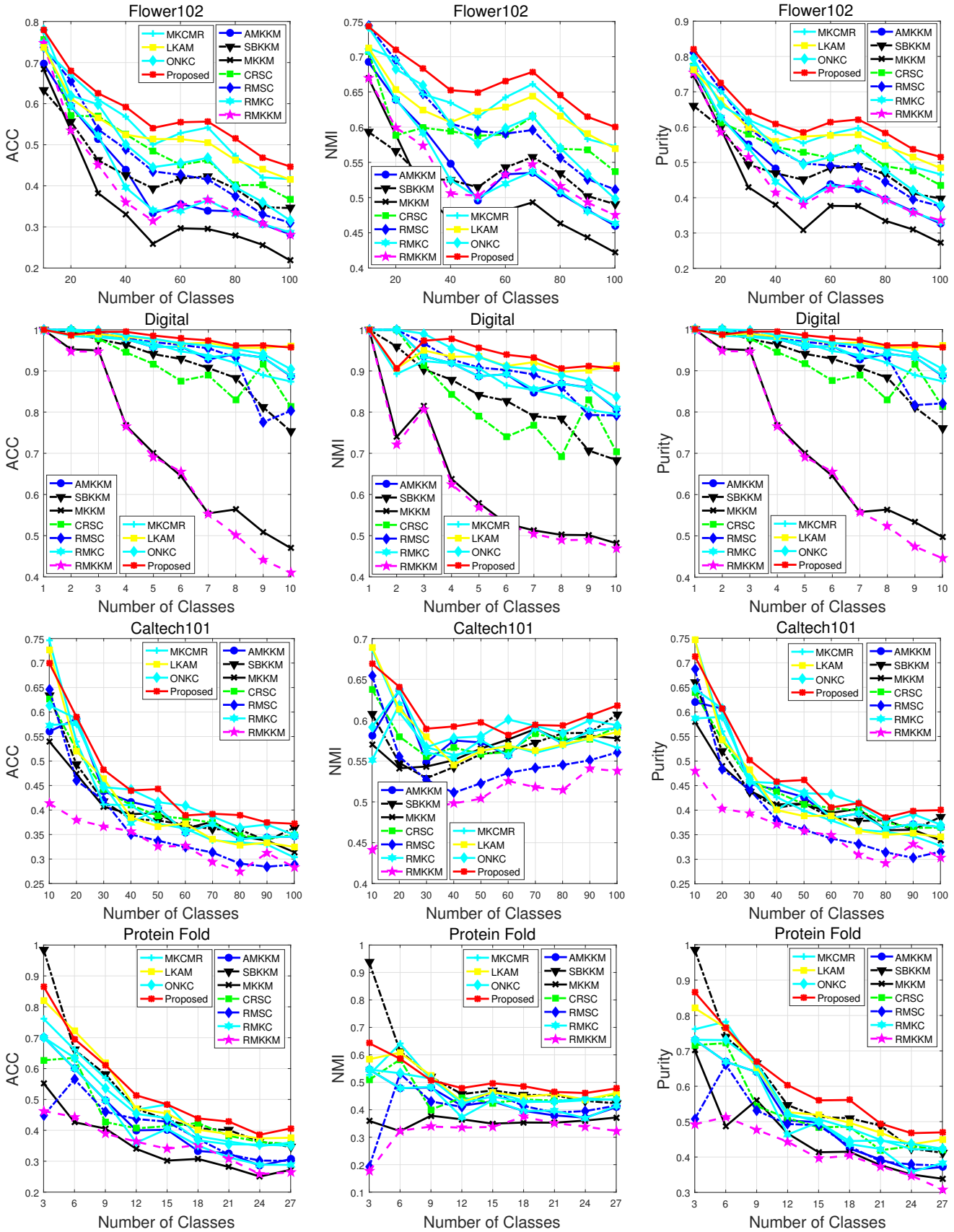

Fig. 2: ACC, NMI and purity variation with number of classes on four datasets, i.e. *Flower102*, *Digital*, *Caltech101* and *Protein Fold*. The other two datasets, *Cornell* and *AR10P*, are not reported for they are composed of a small number of classes and samples. The comparative algorithms are consistent to Table 2, including AMKKM, SBKKM, MKKM [13], RMKKM [17], CRSC [7], RMSC [9], RMKC [10], MKCMR [18], LKAM [26] and ONKC [33].

TABLE 2: ACC, NMI and purity of eleven algorithms on six benchmark datasets.

Dataset	AMKKM	SBKKM	MKKM [13]	CRSC [7]	RMSC [9]	RMKC [10]	RMKKM [17]	MKCMR [18]	LKAM [26]	ONKC [33]	Proposed
ACC											
Flower102	27.29	33.13	21.96	36.79	30.66	33.54	28.17	39.91	40.84	41.56	**45.44**
Digital	88.75	75.40	47.00	85.60	80.30	88.90	41.05	87.45	96.05	91.00	**96.30**
Caltech101	35.56	33.14	34.77	33.33	31.5	35.56	29.67	34.84	33.58	35.91	**38.04**
Protein Fold	30.69	34.58	27.23	35.59	30.12	28.82	26.37	36.02	39.34	39.19	**40.63**
Cornell	52.31	53.85	41.54	48.21	45.13	52.31	43.59	53.85	56.92	53.33	**60.51**
AR10P	38.46	43.08	40.00	36.15	30.00	38.46	31.54	40.77	32.31	41.54	**44.62**
NMI											
Flower102	46.32	48.99	42.30	53.44	50.90	49.73	48.17	57.27	57.60	59.13	**60.22**
Digital	80.59	68.38	48.16	74.95	79.20	80.88	46.85	79.51	91.27	83.95	**91.87**
Caltech101	59.90	59.07	59.64	58.20	58.40	59.90	55.86	60.38	58.78	59.40	**61.66**
Protein Fold	40.96	42.33	37.16	45.66	41.49	41.39	32.30	43.85	46.88	45.78	**47.76**
Cornell	36.60	35.53	4.72	19.64	24.01	36.60	9.22	37.55	39.52	37.55	**43.43**
AR10P	37.27	**42.61**	39.53	36.76	27.40	37.27	26.15	37.35	28.76	37.79	42.18
Purity											
Flower102	32.28	38.78	27.61	42.83	36.62	38.87	33.86	46.39	48.21	47.64	**51.63**
Digital	88.75	76.10	49.70	85.60	82.10	88.90	44.60	87.45	96.05	91.00	**96.30**
Caltech101	37.12	35.10	37.25	35.75	33.27	37.12	31.70	37.19	35.35	37.14	**40.00**
Protein Fold	37.18	41.21	33.86	42.07	37.61	38.33	30.84	42.07	46.11	43.95	**46.97**
Cornell	67.69	66.15	45.64	53.33	60.51	67.69	47.18	68.21	69.23	68.21	**69.74**
AR10P	39.23	43.08	40.00	36.92	30.77	39.23	33.08	42.31	33.08	42.31	**44.62**

TABLE 3: ACC, NMI and purity of LKAM, ON-LK and ON-ALK algorithms on six datasets.

		Flower102	Digital	Caltech101	Protein Fold	Cornell	AR10P
ACC	LKAM [26]	40.84	96.05	33.58	39.34	56.92	32.31
	ON-LK	**43.44**	**96.30**	**37.22**	**41.5**	**57.44**	**38.46**
	Proposed	**45.44**	**96.30**	**38.04**	**40.63**	**60.51**	**44.62**
NMI	LKAM [26]	57.6	91.27	58.78	46.88	39.52	28.76
	ON-LK	**59.95**	**91.65**	**61.52**	**48.85**	**40.82**	**31.8**
	Proposed	**60.22**	**91.87**	**61.66**	**47.76**	**43.43**	**42.18**
Purity	LKAM [26]	48.21	96.05	35.35	46.11	69.23	33.08
	ON-LK	**51.18**	**96.30**	**39.62**	**47.69**	**70.26**	**38.46**
	Proposed	**51.63**	**96.30**	**40.00**	**46.97**	**69.74**	**44.62**

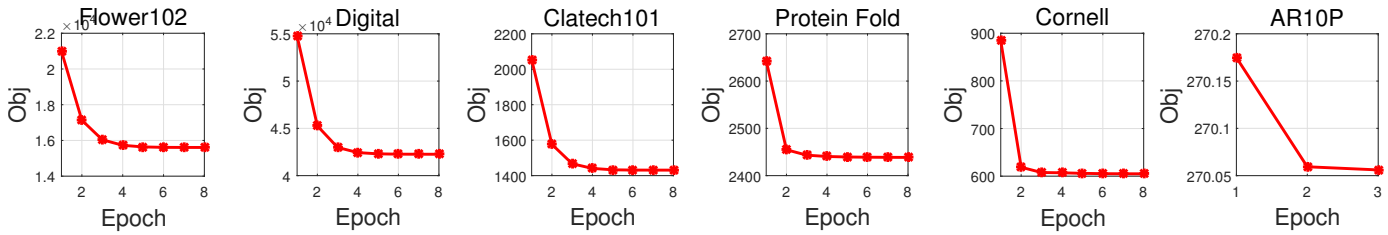

Fig. 3: Objective values at each iteration with $\rho = 2.\hat{}-1$ and $\zeta = 0$ on six datasets, i.e. *Flower102*, *Caltech101*, *Digital*, *Protein Fold*, *Cornell* and *AR10P*. It can be seen that the objectives monotonically decrease and converge to minimums with a small number of iterations.

NMI and purity variation with number of classes. *Flower102* and *Caltech101* both have 102 classes, therefore, we calculate the three measurements when increasing the class number by 10. For *Digital* and *Protein Fold*, there are 10 and 27

classes each, so we gradually increase the class number by 1 and 3, respectively. The results show the proposed algorithm, together with the others, decreases gradually with increasing the class number, except the ones on *Flower102*. On *Flower102*, the proposed algorithm also shows a similar trend with the others, in which the three measurements firstly drop to minimums and further increase. But, generally, the proposed algorithm has a better clustering performance over the others in recent literatures not only on datasets of few classes but also on those of large class number. The proposed algorithm shows relatively weaker ACC, NMI and purity than SBKKM on *Protein Fold* with setting the class number to 3. We think the small datasets introduce more randomness and not rich enough to describe the sample distribution. At the same time, the proposed algorithm still outperforms the most ones, verifying its effectiveness.

The proposed ON-ALK algorithm improves the performance of MKC algorithm from two aspect, i.e. the *adaptive local kernel* and the *optimal neighborhood kernel*. For the sake of sufficiently illustrate the effectiveness of these regularizations, we make a minor change on the proposed algorithm by keeping sizes of all local kernels the same and name it as ON-LK. In this case, whether locating the optimal kernel in neighborhood areas of linear combinations becomes the only difference between ON-LK and LKAM proposed in [26]. At the same time, whether employing the adaptive local kernels is also the only distinction between ON-LK and ON-ALK. Experiments are carried out on the aforementioned three algorithms and the three clustering measurements, including ACC, NMI and purity, are gathered in Table 3. We mark the biggest values in red and the middle ones in bold, while leaving the minimums unmarked.

From the first perspective, we compare the performances of LKAM and ON-LK algorithm in Table 3. The ON-LK algorithm significantly outperforms LKAM on six datasets in ACC, i.e. 2.60% on *Flower102*, 3.64% on *Caltech101*, 0.25% on *Digital*, 2.16% on *Protein Fold*, 0.52% on *Cornell* and 6.15% on *AR10P*. Similar results are achieved in NMI and purity, demonstrating the effectiveness of *optimal neighborhood kernel*. By extending the domain of optimal kernel from weighted combinations of pre-specified kernels to their neighborhood areas, the proposed algorithm enlarges the kernel search range, resulting in the improvement of clustering performance.

From the second perspective, although the results on *Protein Fold* of the proposed algorithm are slightly weaker, i.e. 0.87% lower than ON-LK in ACC, it achieves better performances over the ON-LK algorithm in most cases, specifically, by 2.00% on *Flower102*, 3.07% on *Cornell* and 6.16% on *AR10P* in ACC, illustrating the superiority of the *adaptive local kernel*. The proposed algorithm allows the sample numbers of local kernels vary along with the density around samples. This improvement removes the farther sample pairs from local kernels, decreasing the unreliability of aligning father sample pairs, therefore, obtains better performances.

From the above analysis, it can be concluded that the adaptive local kernel and relaxing the optimal kernel around the linearly combined kernel can both improve the performance of MKC algorithm.

5.4 Parameter study and convergence

We conduct parameter study on the two parameters in the proposed algorithm. Fig. (4) reports the clustering performances on *Flower102* when varying one parameter with the other fixed. It can be observed that ACC, NMI and purity increase dramatically, reach the top near 2^{-1}, then slightly decrease while ρ gradually increases from 2^{-15} to 2^{15} with $\zeta = 0$. With fixing ρ at 2^{-1}, the three metrics keep steady when $\zeta \leq -0.1$, reach the top at $\zeta = 0$, then drop dramatically. At the same time, we also present the results on *Digital*, as shown Fig. (5). The same trend can be obtained when varying ρ, while the curves on ζ show a slight difference, that the results increase at first. By observing ACC, NMI and purity keep relatively stable at $\rho \in [2^{-5}, 2^{5}]$ and $\zeta \in [-0.2, 0.1]$ on *Flower102*, *Digital* and the others (not presented for limit of space), we recommend to set the two parameters in these ranges. This parameter recommendation setting is also widely adopted in current literatures, such as [18], [33], [42]. In addition, we also explore a new approach to choose the parameters on unlabeled datasets. Davies Bouldin (DB) [44] index is an internal metric for evaluating clustering, which requires no ground-truth class labels. The smaller it is, the better clustering result is obtained. Therefore, we plot the -DB on $\rho \in 2.\hat{}[-15, -14, \cdots, 15]$ and $\zeta \in [-0.5, -0.4, \cdots, 0.5]$, as shown in Fig. (4) and (5). It can be observed that -DB curves share the same trend with ACC, NMI and purity, respectively. This indicates that users can also select ideal parameters according to the internal metric DB in their applications.

We also investigate the objective values of the proposed algorithm at each iteration with $\rho = 2^{-1}$ and $\zeta = 0$, as presented in Fig. (3). It shows that the objective value monotonically and rapidly decreases along with the clustering process, indicating the convergence of the proposed algorithm. In most cases, the proposed algorithm converges with fewer than twenty iterations.

5.5 Evaluation on additional datasets

In order to verify the effectiveness and generalization ability on datasets of the proposed algorithm more sufficiently, we conduct additional experiments on six new datasets [45], including *A* [46], *Birch* [47], *DIM* [48], *G2* [49], *S* [50] and *Unbalance* [51], which are described in Table 5. Note that there are too many subsets of *G2*, and the subset with 2 centroids and 1024 dimensions are used. Additionally, we generate the corresponding kernel matrices with six kernel mappings, such as linear, gaussian, etc. We also adopt a new evaluation criterion, i.e. Centroid index (CI) [52], which evaluates the difference between predicted clustering centroids and the ground truth, reflecting the quality of the clustering model. It counts at the cluster level how many clusters are incorrectly solved, and CI=0 indicates perfect cluster-level solution.

We report the performances in terms of CI in Table 4, while the corresponding ACC, NMI and purity can be found in Table 1 of the *Appendix*. Besides, if the proposed algorithm achieves the best results, corresponding values are marked in bold, or the best results are marked. Three observations can be concluded as:

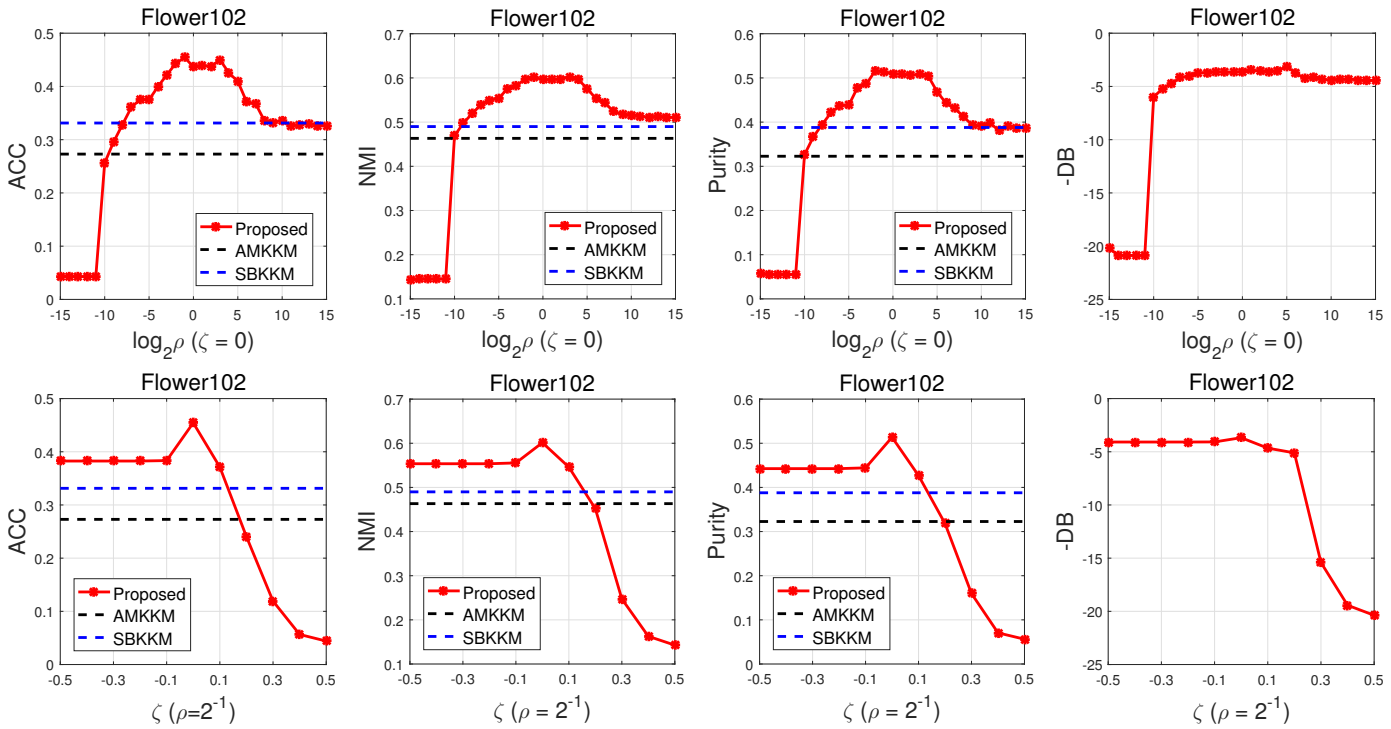

Fig. 4: Parameter sensibility study on *Flower102*. (a) ACC, NMI and Purity when fixing ζ as 0 and varying ρ in $2.\hat{}[-15, -14, \cdots, 15]$; (b) ACC, NMI and Purity when fixing ρ as 2^{-1} and varying ζ in $[-0.5, -0.4, \cdots, 0.5]$. The comparative algorithms, AMKKM and SBKKM, are free of parameters, so horizontal lines are presented.

.

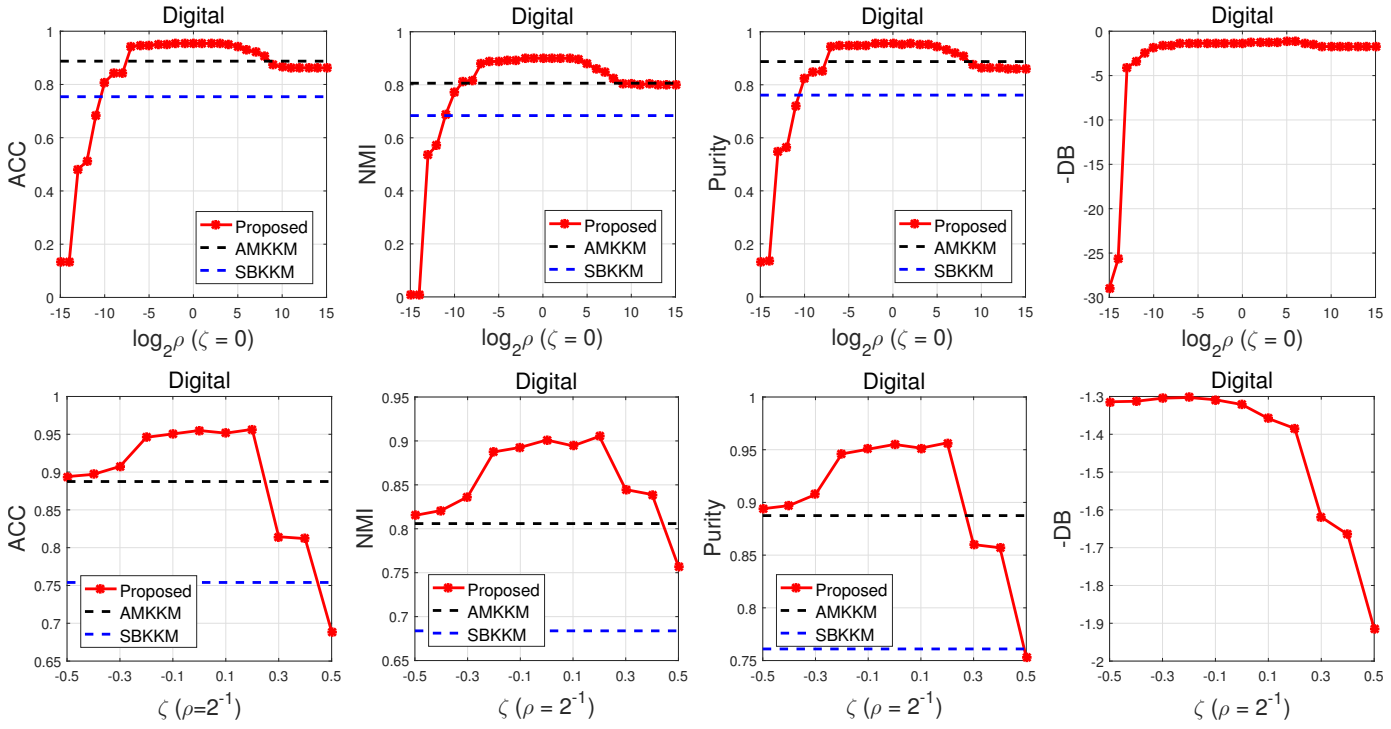

Fig. 5: Parameter sensibility study on *Digital*. (a) ACC, NMI and Purity when fixing ζ as 0 and varying ρ in $2.\hat{}[-15, -14, \cdots, 15]$; (b) ACC, NMI and Purity when fixing ρ as 2^{-1} and varying ζ in $[-0.5, -0.4, \cdots, 0.5]$. The comparative algorithms, AMKKM and SBKKM, are free of parameters, so horizontal lines are presented.

.

TABLE 4: Centroid Index of eleven algorithms on additional datasets [45].

Dataset	AMKKM	SBKKM	MKKM [13]	CRSC [7]	RMSC [9]	RMKC [10]	RMKKM [17]	MKCMR [18]	LKAM [26]	ONKC [33]	Proposed
A1	0	0	10	0	9	0	5	2	0	0	**0**
A2	1	1	17	1	22	1	5	1	1	1	**1**
A3	4	3	38	3	28	4	6	4	3	4	**3**
Birch1	16	**8**	73	11	47	16	19	17	16	15	12
Birch2	15	15	64	12	29	15	19	12	15	12	**10**
DIM032	0	0	0	0	1	0	2	1	0	0	**0**
DIM064	1	0	1	0	2	0	2	0	0	0	**0**
DIM128	1	0	1	0	2	0	2	1	0	1	**0**
DIM256	2	0	0	0	2	1	2	2	0	2	**0**
DIM512	2	1	1	1	1	1	3	2	0	2	**0**
DIM1024	2	1	0	0	4	2	3	1	0	2	**0**
G2	0	0	0	0	0	0	0	0	0	0	**0**
S1	0	0	9	0	6	0	2	0	0	0	**0**
S2	0	0	9	0	8	0	1	0	0	0	**0**
S3	0	0	7	0	6	0	2	0	0	0	**0**
S4	0	0	5	0	8	0	2	0	0	0	**0**
Unbalance	4	4	4	**3**	4	4	4	4	4	4	4
Average	2.82	1.94	14.06	1.82	10.53	2.59	4.65	2.76	2.29	2.53	**1.76**

* Note: *Average* column is computed with the results of each algrithm on all datasets.

TABLE 5: Specifications of additional datasets [45].

Dataset	Subset	Number of		
		Samples	Clusters	Dimensions
A	A1	3000	20	2
	A2	5250	35	2
	A3	7500	50	2
Birch	Birch1	10000	100	2
	Birch2	10000	100	2
DIM	DIM032	1024	16	32
	DIM064	1024	16	64
	DIM128	1024	16	128
	DIM256	1024	16	256
	DIM512	1024	16	512
	DIM1024	1024	16	1024
G2	G2	2048	2	1024
S	S1	5000	15	2
	S2	5000	15	2
	S3	5000	15	2
	S4	5000	15	2
Unbalance	Unbalance	6500	8	2

1) The proposed algorithm consistently outperforms the other methods in CI, showing its effectiveness and superiority.
2) The proposed algorithm achieves over 90% ACC, NMI and Purity in most datasets, even 100% sometimes, verifying its effectiveness.
3) Although some comparative methods get better performances in some datasets, the gaps between them are so small, near 0.1%. This may result from the simplicity of datasets. At the same time, the proposed algorithm is much more stable than the others, showing a better performance on *Average* over all datasets.

6 Discussion and Future Work

This paper improves the performance of MKC algorithm via constructing the adaptive local kernel acoording to sample density information. While, Martin et al. also utilize the density information but use it to enhance the original kernel matrices [53]. We are going to explore its merits in the further work. Nevertheless, It is a widely used approach to construct a consensus kernel for clustering by linearly combining a set of base kernels. With a step further, our method locates the optimal one around the kernel combinations. Meanwhile, some researchers claim that any dot-product kernel can be defined as a linear combination of polynomial kernels [54], [55], as shown in Theorem 2 in [56]. We are going to explore the properties of dot-product kernel and construct the optimal kernel with a set of polynomial kernels in the future work.

7 Conclusion

While the recently proposed MKC algorithms are able to handle multiple kernel clustering, they usually do not sufficiently consider the local density around individual data samples and excessively limit the representation capacity of the learned optimal kernel, leading to unsatisfying performance. This paper proposes an algorithm, named optimal neighborhood multiple kernel clustering with adaptive local kernels, to address these issues. The proposed algorithm is elegantly solved and its effectiveness and superiority are well demonstrated via conducting comprehensive experiments on benchmark datasets.

Acknowledgements

This work was supported by the Natural Science Foundation of China (project no. 61773392 and 61922088) and Education Ministry-China Mobile Research Funding (project no. MCM20170404).

REFERENCES

[1] K. Krishna and N. M. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems Man And Cybernetics-Part B: Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999.

[2] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[3] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[5] Z. Zivkovic *et al.*, "Improved adaptive gaussian mixture model for background subtraction." in *ICPR (2)*. Citeseer, 2004, pp. 28–31.

[6] A. Trivedi, P. Rai, H. Daumé III, and S. L. DuVall, "Multiview clustering with incomplete views," in *NIPS Workshop*, vol. 224, 2010.

[7] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 393–400.

[8] W. Shao, X. Shi, and S. Y. Philip, "Clustering on multiple incomplete datasets via collective kernel learning," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 1181–1186.

[9] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[10] P. Zhou, L. Du, L. Shi, H. Wang, and Y.-D. Shen, "Recovery of corrupted multiple kernels for clustering," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[11] H. Wang, Y. Yang, and B. Liu, "Gmc: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2020.

[12] 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*, 2018.

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

[14] 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.

[15] D. Guo, J. Zhang, X. Liu, Y. Cui, and C. Zhao, "Multiple kernel learning based multi-view spectral clustering," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 3774–3779.

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

[17] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means using l21-norm," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

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

[19] Y. Zhao, Y. Dou, X. Liu, and T. Li, "A novel multi-view clustering method via low-rank and matrix-induced regularization," *Neurocomputing*, vol. 216, pp. 342–350, 2016.

[20] 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*, 2019.

[21] P. Zhang, Y. Yang, B. Peng, and M. He, "Multi-view clustering algorithm based on variable weight and mkl," in *International Joint Conference on Rough Sets*. Springer, 2017, pp. 599–610.

[22] 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*, 2019.

[23] G. F. Tzortzis and A. C. Likas, "Multiple view clustering using a weighted combination of exemplar-based mixture models," *IEEE Transactions on neural networks*, vol. 21, no. 12, pp. 1925–1938, 2010.

[24] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 795–828, 2012.

[25] 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.

[26] M. Li, X. Liu, W. Lei, D. Yong, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *International Joint Conference on Artificial Intelligence*, 2016.

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

[28] Q. Wang, Y. Dou, X. Liu, F. Xia, Q. Lv, and K. Yang, "Local kernel alignment based multi-view clustering using extreme learning machine," *Neurocomputing*, vol. 275, pp. 1099–1111, 2018.

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

[30] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "Lp-norm multiple kernel learning," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 953–997, 2011.

[31] C. Cortes, M. Mohri, and A. Rostamizadeh, "L 2 regularization for learning kernels," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 109–116.

[32] I. Lauriola, M. Polato, and F. Aiolli, "The minimum effort maximum output principle applied to multiple kernel learning," in *26th European Symposium on Artificial Neural Networks, ESANN 2018, Bruges, Belgium, April 25-27, 2018*, 2018. [Online]. Available: http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-181.pdf

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

[34] T. Liu, D. Tao, and D. Xu, "Dimensionality-dependent generalization bounds for k-dimensional coding schemes," *Neural Comput.*, vol. 28, no. 10, pp. 2213–2249, Oct. 2016. [Online]. Available: https://doi.org/10.1162/NECO_a_00872

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

[36] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 722–729.

[37] M. van Breukelen, R. P. W. Duin, D. M. J. Tax, and J. E. den Hartog, "Handwritten digit recognition by combined classifiers," *Kybernetika*, vol. 34, no. 4, pp. 381–386, 1998. [Online]. Available: http://www.kybernetika.cz/content/1998/4/381

[38] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2004, Washington, DC, USA, June 27 - July 2, 2004*. IEEE Computer Society, 2004, p. 178. [Online]. Available: https://doi.org/10.1109/CVPR.2004.383

[39] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinform.*, vol. 17, no. 4, pp. 349–358, 2001. [Online]. Available: https://doi.org/10.1093/bioinformatics/17.4.349

[40] M. Craven and S. Slattery, "Relational learning with statistical predicate invention: Better models for hypertext," *Machine Learning*, vol. 43, no. 1-2, pp. 97–119, 2001.

[41] L. Ding and A. M. Martínez, "Features versus context: An approach for precise and detailed detection and delineation of faces and facial features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2022–2038, 2010. [Online]. Available: https://doi.org/10.1109/TPAMI.2010.28

[42] T. Wang, D. Zhao, and S. Tian, "An overview of kernel alignment and its applications," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 179–192, 2015.

[43] T. Damoulas and M. A. Girolami, "Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection," *Bioinformatics*, vol. 24, no. 10, pp. 1264–1270, 2008.

[44] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[45] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," *Applied Intelligence*, vol. 48, no. 12, pp. 4743–4759, 2018.

[46] I. Kärkkäinen and P. Fränti, *Dynamic local search algorithm for the clustering problem*. University of Joensuu, 2002.
[47] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications," *Data Min. Knowl. Discov.*, vol. 1, no. 2, pp. 141–182, 1997.
[48] P. Fränti, O. Virmajoki, and V. Hautamäki, "Fast agglomerative clustering using a k-nearest neighbor graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1875–1881, 2006.
[49] P. Fränti, R. Mariescu-Istodor, and C. Zhong, "XNN graph," in *Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshop, S+SSPR 2016, Mérida, Mexico, November 29 - December 2, 2016, Proceedings*, 2016, pp. 207–217.
[50] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognition*, vol. 39, no. 5, pp. 761–775, 2006.
[51] M. Rezaei and P. Fränti, "Set matching measures for external cluster validity," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2173–2186, 2016.
[52] P. Fränti, M. Rezaei, and Q. Zhao, "Centroid index: cluster level similarity measure," *Pattern Recognition*, vol. 47, no. 9, pp. 3034–3045, 2014.
[53] D. Marin, M. Tang, I. B. Ayed, and Y. Boykov, "Kernel clustering: Density biases and solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 136–147, 2019. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2780166
[54] M. Donini and F. Aiolli, "Learning deep kernels in the space of dot product polynomials," *Mach. Learn.*, vol. 106, no. 9-10, pp. 1245–1269, 2017. [Online]. Available: https://doi.org/10.1007/s10994-016-5590-8
[55] I. Lauriola, M. Polato, and F. Aiolli, "Radius-margin ratio optimization for dot-product boolean kernel learning," in *Artificial Neural Networks and Machine Learning - ICANN 2017 - 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11-14, 2017, Proceedings, Part II*, ser. Lecture Notes in Computer Science, A. Lintas, S. Rovetta, P. F. M. J. Verschure, and A. E. P. Villa, Eds., vol. 10614. Springer, 2017, pp. 183–191. [Online]. Available: https://doi.org/10.1007/978-3-319-68612-7_21
[56] P. Kar and H. Karnick, "Random feature maps for dot product kernels," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21-23, 2012*, ser. JMLR Proceedings, N. D. Lawrence and M. A. Girolami, Eds., vol. 22. JMLR.org, 2012, pp. 583–591. [Online]. Available: http://proceedings.mlr.press/v22/kar12.html

Jiyuan Liu is a PhD student in National University of Defense Technology (NUDT), China. His current research interests include unsupervised multiple-view clustering, deep one class classification, deep clustering.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Assistant Researcher of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ 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, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc. He serves as the associated editor of Information Fusion Journal. More information can be found at https://xinwangliu.github.io/.

Jian Xiong received the B.S. degree in engineering, and the M.S. and Ph.D. degrees in management from National University of Defense Technology, Changsha, China, in 2005, 2007, and 2012, respectively. He is an Associate Professor with the School of Business Administration, Southwestern University of Finance and Economics. His research interests include data mining, multiobjective evolutionary optimization, multiobjective decision making, project planning, and scheduling.

Qing Liao received her Ph.D. degree in computer science and engineering in 2016 supervised by Prof. Qian Zhang from the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. She is currently an assistant professor with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include artificial intelligence and data mining.

Sihang Zhou is a Ph.D student of National University of Defense Technology (NUDT), Changsha, PR China. He received M.S. degree in Computer Science from the same school in 2014 and his bachelor's degree in information and Computing Science from the University of Electronic Science and Technology of China (UESTC) in 2012. His current research interests include machine learning, pattern recognition, and medical image analysis.

Siwei Wang is a graduate student in National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning,scalable kernel k-means and deep neutral network.

Yuexiang Yang received the B.S. degree in Mathematics from Xiangtan University, Xiangtan, China, in 1986, the M.S. degree in Computer Application and the PHD degree in Computer Science and Technology from National University of Defense Technology, Changsha, China, in 1989 and 2008, respectively. His research interests include information retrieval, network security and data analysis. He is the executive director of the Informatization Branch of China Higher Education Association. He has co-authored more than 100 papers in international journals and conference or workshop proceedings. He has been serving as reviewer and program committee member of various conferences and journals.