Multiple Kernel Clustering With Neighbor-Kernel Subspace Segmentation

Sihang Zhou, Xinwang Liu, Miaomiao Li, En Zhu, Li Liu, Changwang Zhang, and Jianping Yin

***Abstract*— Multiple kernel clustering (MKC) has been intensively studied during the last few decades. Even though they demonstrate promising clustering performance in various applications, existing MKC algorithms do not sufficiently consider the intrinsic neighborhood structure among base kernels, which could adversely affect the clustering performance. In this paper, we propose a simple yet effective neighbor-kernel-based MKC algorithm to address this issue. Specifically, we first define a neighbor kernel, which can be utilized to preserve the block diagonal structure and strengthen the robustness against noise and outliers among base kernels. After that, we linearly combine these base neighbor kernels to extract a consensus affinity matrix through an exact-rank-constrained subspace segmentation. The naturally possessed block diagonal structure of neighbor kernels better serves the subsequent subspace segmentation, and in turn, the extracted shared structure is further refined through subspace segmentation based on the combined neighbor kernels. In this manner, the above two learning processes can be seamlessly coupled and negotiate with each other to achieve better clustering. Furthermore, we carefully design an efficient iterative optimization algorithm with proven convergence to address the resultant optimization problem. As a by-product, we reveal an interesting insight into the exact-rank constraint in ridge regression by careful theoretical analysis: it back-projects the solution of the unconstrained counterpart to its principal components. Comprehensive experiments have been conducted on several benchmark data sets, and the results demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*— Kernel method, multiple kernel learning, neighbor kernel, subspace segmentation.**

I. Introduction

MULTIPLE kernel clustering (MKC) [1]–[6] provides an elegant framework to group samples into clusters by extracting and enhancing common structure from complementary information sources (base kernels). Based on the criteria that are utilized to guide the common cluster structure extraction, existing methods can be roughly categorized into four branches, i.e., margin-based [1], [7], spectral clustering-based [4], [8], [9], kernel k-means-based [3], [10]–[15], and kernel decomposition-based [2], [16]–[21] algorithms. Among these algorithms, margin-based methods extend the maximum margin classification formulation [22], [23] into the field of unsupervised learning by simultaneously assigning labels and maximizing the margin between different clusters [1], [7]. Spectral clustering-based methods adopt cotraining and coregularization mechanisms to look for clusters that are consistent across views [4], [8]. Kernel k-means-based algorithms construct an optimal kernel that integrates intrinsic information from all views for k-means clustering [3], [12], [24]. Comparatively, kernel decomposition-based methods factorize the prespecified base kernels in various fashions to filter the noisy information while extracting the shared discriminative structure [18], [19], [21]. Given their inherent antinoise capability, kernel decomposition-based MKC algorithms tend to provide more robust and promising performance in various applications [2], [16], [17] and thus remain a hotspot of research activity in the field. Our proposed algorithm in this paper belongs to this category.

The goal of kernel decomposition-based MKC is to find an effective method of kernel factorization that can best eliminate the adverse effect of noise and outliers among base kernels and extract complementary discriminative information for clustering. Under the guidance of this roadmap, the work in [2] models this problem as a multiple undirected graph mining task. They propose a novel linked matrix factorization (LMF) algorithm to extract common information from multiple graphs and filter out irrelevant information. In [17] and [21], the base kernels are reconstructed as a combination of a shared low-rank matrix and different sparse matrices. In this formulation, the low-rank matrix stands for the common cluster structure, while sparse matrices stand for different noises within base kernels. To more appropriately model the kernel noise and add better regularization to kernel decomposition, an $\ell_{2,1}$-norm and a positive semidefinite (PSD) constraint are introduced to the objective function for the noise matrices and the low-rank matrix in [19].

The aforementioned methods share a common assumption that samples are approximately drawn from a common low-dimensional space. However, in practical applications, it is not uncommon that a given data set cannot be appropriately represented by a single subspace. A more reasonable remedy

Manuscript received March 18, 2018; revised October 31, 2018 and May 24, 2019; accepted May 27, 2019. This work was supported by the National Key R&D Program of China 2018YFB1003203 and the National Natural Science Foundation of China (project no. 61773392 and 61672528). *(Corresponding author: Xinwang Liu.)*

S. Zhou, X. Liu, and E. Zhu are with the School of Computer Science, National University of Defense Technology, Changsha 410073, China (e-mail: sihangjoe@gmail.com; xinwangliu@nudt.edu.cn; enzhu@nudt.edu.cn).

M. Li is with the College of Changsha, Changsha 410073, China (e-mail: miaomiaolinudt@gmail.com).

L. Liu is with the College of System Engineering, National University of Defense Technology, Changsha 410073, China, and also with the Department of Machine Vision and Signal Analysis, University of Oulu, 90014 Oulu, Finland (e-mail: li.liu@oulu.fi).

C. Zhang is with the Technology and Engineering Group, Tencent Technology (Shenzhen) Co., Ltd., Shenzhen 518064, China (e-mail: changwzhang@tencent.com).

J. Yin is with the School of Cyberspace Science, Dongguan University of Technology, Guangdong 523808, China (e-mail: jpyin@dgut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2919900

is to assume that samples are polluted observations drawn from a mixture of several independent subspaces [25]. To this end, based on the multisubspace hypothesis, the work in [20] proposed a subspace segmentation-based MKC algorithm. In this method, a common consensus sparse reconstruction matrix was optimized to reveal the intrinsic subspaces shared by base kernels. To encourage the reconstruction matrix to integrate more diverse information from the base kernels, the Hilbert Schmidt independence criterion (HSIC) is utilized as a diversity measuring term to encourage exploring the complementarity of multiple representations in [18]. These methods launched another substantial step to improve the performance of MKC algorithms. However, the mentioned models are designed globally, which implies all relationships between any of the two components should be finely and equally considered. Nevertheless, this setting neglects a well-established problem in which the similarity evaluated for two distant samples in a high-dimensional space is less reliable due to the presence of the underlying manifold structure [26]–[31]. Furthermore, as pointed out in [32], for unsupervised tasks, preserving the local geometric structure of data is much more effective than preserving pairwise similarity. To this end, several algorithms have been proposed. The work in [18] introduces graph regularization to encourage local geometry preservation in MKC subspace clustering methods. However, the resultant extra hyperparameter that balances the importance of this term is not preferable in unsupervised learning scenarios. A novel local kernel alignment-based method that only focuses on aligning the local elements within base kernels is proposed in [33]. However, since the base kernels can be noisy [9], the lack of a noise elimination mechanism limits the performance of this algorithm. In [20], by minimizing both the ℓ_1-norm and the ℓ_2-norm of the reconstruction matrix, the proposed formulation intrinsically enforces the method to represent samples with nearby counterparts. However, in this paper, the authors drew help from human experts and set the weights of kernels manually, which limits the applicability of the proposed algorithm.

To solve the aforementioned issues, we propose a novel neighbor-kernel-based MKC algorithm in the framework of subspace segmentation. Although preserving the neighborhood relation among samples may have been presented in solving other tasks, our work is distinguished by the following.

1) Identifying an important issue for the first time, which has been overlooked in multiple kernel subspace segmentation, and proposing an effective solution.
2) Designing a novel neighbor-kernel algorithm to solve the resultant optimization problem and theoretically analyzing its convergence and computational complexity. Through careful mathematical analysis, we discover that our solution to the exact-rank-constrained least-square regression problem is identical to the optimal solution in [24], thus also finds the global minimizer. The result also sheds light on the intrinsic mechanism of exact-rank-constrained ridge regression by revealing its relationship with the unconstrained counterpart.
3) Performing extensive experiments on both synthetic and popular benchmark data sets that validate our identification of the issues and the effectiveness of our solution. Moreover, experimental results also proved the effectiveness of our proposed neighbor kernels on improving the performance of the existing state-of-the-art methods.
4) The proposed algorithm is readily extended to other related multiview topics, such as multiview clustering and subspace learning.

II. Method

A. Construction of the Neighbor Kernel

It is common for existing kernel-based methods to reveal the underlying structure of data by calculating the pairwise similarity between samples. However, in many successful machine learning algorithms, such as dimensionality reduction [34], [35], clustering [16], [36], and recent feature selection algorithms [32], [37], [38], researchers find that it is beneficial to preserve only the reliable local geometry as a representation of the data structure. There are two main underlying reasons. On the one hand, since the global nonlinear high-dimensional structure can be finely reserved by hooking the local geometry patches, preserving only the local similarity among data will not degrade the capacity of the corresponding algorithms to reveal the global data structure [34]. On the other hand, as pointed out in [26], the similarity estimation between relatively long-distance samples may be inaccurate since the ambient geometry in the high-dimensional input space may be highly folded, twisted, or curved. Moreover, even worse are the disturbances caused by noise and outliers within data, which can further undermine the structure of the underlying manifold, making the long-distance similarity more unreliable. As a consequence, in the unsupervised kernel learning scenario, without the discriminative guidance of labels, it is a reasonable and practical strategy to preserve only the high-confidence local similarities for learning the intrinsic global manifold of data.

Based on the idea of representing the global intrinsic manifold in kernels with local structure patches, we first show how to construct the neighbor kernel. Its construction includes three steps, i.e., neighbor searching, kernel construction, and normalization. In the first step, the neighbors of samples are searched by finding the nearest k samples in the average kernel space. It is worth noting that this operation only requires that most of the base kernels are informative and are complementary to each other, and it is much weaker than the requirement of the cotraining-based methods that require all the base kernels to be informative for clustering [4], [8]. By taking this approach, we integrate complementary information from different base kernels to help robustly reveal the correct neighborhood relationships among samples. Denote the neighbor indicating matrix for a sample j as $\mathbf{N}^{(j)} \in \{0, 1\}^{S\times S}$, where S is the sample number of the data set, $\mathbf{N}^{(j)}(a, b) = 1$ if sample a, b are both neighbors of sample j according to the average kernel space metric, and $\mathbf{N}^{(j)}(a, b) = 0$ if not. Then, given a set of base kernels $\{\mathbf{K}_i^o\}(i = 1, \cdots, p)$,

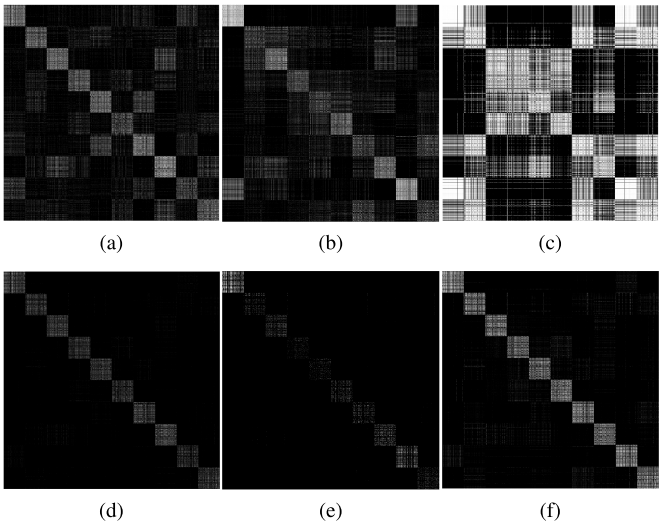


Fig. 1. Illustration of (a)–(c) original base kernels and (d)–(f) corresponding neighbor kernels on UCI-Digit data set.

the corresponding neighbor kernel $\mathbf{K}_i$ of a base kernel $\mathbf{K}_i^o$ can be formulated as

$$\mathbf{K}_i = \sum_{j=1}^{S} \mathbf{N}^{(j)} \circ \mathbf{K}_i^o$$

where $\circ$ is the Hadamard product. Through the formulation, we can see that a neighbor kernel is constructed by extracting and summing the neighbor elements of each sample in the original kernel space. It is easy to ascertain that for each index j, the neighbor indexing matrix $\mathbf{N}^{(j)}$ is PSD, so the matrix $\sum_{j=1}^{S} \mathbf{N}^{(j)}$ is also PSD. As a consequence, the constructed matrix $\mathbf{K}_i$ is also a kernel matrix. Finally, we normalize the generated neighbor kernels to unit trace.

In Fig. 1, we illustrate original base kernels and the corresponding neighbor kernels on the UCI-Digit data, which is a widely used benchmark in MKC. For the neighbor kernels, the number of neighbors is fixed as $0.01 * S$. For better illustration, we permute the order of the matrices to illustrate assembling samples from the same category. As seen from Fig. 1, the constructed neighbor kernels are more discriminative (with a better block diagonal structure). Additionally, the noise within kernels is largely suppressed. This phenomenon is more obvious in the third kernel [see Fig. 1(c) and (f)]. Two reasons contribute to the merits of neighbor kernels. First, since the neighbor samples are more likely to lie in the same cluster with each other, keeping only the neighborhood similarities may help to maintain the essential connection while cutting off the weak ones. Second, the complementary information extracted from different views remedies the missing information of each other. Since the neighbor kernels can better reveal the intrinsic cluster structure of data sets, they can well meet the subspace independent assumption and are thus appealing inputs to the subspace segmentation algorithms.

B. Subspace Segmentation

Subspace segmentation, also known as subspace clustering, is a family of methods that models a collection of data points as the integration of noise and a union of subspaces. The goal of these methods is to group data into clusters, with each cluster corresponding to a subspace. In the literature of this field, a commonly adopted formulation is

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{Z}\|_\dagger + \lambda \|\mathbf{E}\|_\ddagger, \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E} \tag{1}$$

where $\mathbf{X}, \mathbf{E} \in \mathbb{R}^{d \times S}$, and $\mathbf{Z} \in \mathbb{R}^{S \times S}$ are the data matrix, the noise representation matrix, and the reconstruction matrix, respectively. Here, d is the dimensionality of the input feature space. $\|\cdot\|_\dagger$ and $\|\cdot\|_\ddagger$ indicate different norms, such as the ℓ_1-norm, ℓ_2-norm, nuclear norm, and so on. Specifically, in [39], the ℓ_1-norm of both the reconstruction matrix and the noise representation matrix are minimized to extract a sparse representation and filter noise within samples. In [40], researchers efficiently seek the block diagonal structure of data by minimizing the Frobenius norm of both matrices under the assumption of subspace independence. In [41], $\|\mathbf{Z}\|_*$ and $\|\mathbf{E}\|_{2,1}$ are minimized to categorize samples to their respective subspaces and remove the possible outliers.

Generally, in many of the popular methods in this branch, a common target is to filter noise and reveal the intrinsic block diagonal structure of data [40]. Since the neighbor kernel introduced in Section II-A possesses better block diagonal structure and robustness against noise and outliers, using these kernels as input largely decreases the difficulty of data reconstruction and noise modeling. In turn, with the optimization of subspace segmentation, the remaining noise within neighbor kernels can be further filtered, and the cluster structure can be refined. To make full use of the two techniques to better serve MKC, in Section II-C, we combine them into one framework and propose a multiple neighbor-kernel subspace segmentation algorithm.

C. Multiple Neighbor-Kernel Subspace Segmentation

In this section, we integrate multiple neighbor-kernels $\mathbf{K}_i (i = 1, \cdots, p)$ with exact low-rank subspace segmentation to extract complementary information from different base kernels and achieve better clustering performance. The formulation of our algorithm is as follows:

$$\begin{aligned} &\min_{\mathbf{Z},\boldsymbol{\mu}} \|\mathbf{K}_\mu - \mathbf{K}_\mu \mathbf{Z}\|_{\mathrm{F}}^2 + \alpha \|\mathbf{Z}\|_{\mathrm{F}}^2 + \beta \boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu} \\ &\text{s.t. } \operatorname{rank}(\mathbf{Z}) = l, \quad \mathbf{K}_\mu = \sum_{i=1}^{p} \boldsymbol{\mu}_i \mathbf{K}_i, \\ &\qquad \boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_1 = 1 \end{aligned} \tag{2}$$

where p is the number of base kernels, l is the expected rank of $\mathbf{Z}$, $\mathbf{M} \in \mathbb{R}^{p \times p}$ is the centered kernel alignment-based kernel correlation matrix [42], and $\boldsymbol{\mu} \in \mathbb{R}^p$ is the weight vector for linear kernel combination. The definition of $\mathbf{M}$ is: $\mathbf{M}_{a,b} = \operatorname{Tr}(\mathbf{K}_a, \mathbf{K}_b)/(\|\mathbf{K}_a\|_{\mathrm{F}} \|\mathbf{K}_b\|_{\mathrm{F}})$. Here, $\operatorname{Tr}(\mathbf{K}_a^\top \mathbf{K}_b)$ calculates the trace of $\mathbf{K}_a^\top \mathbf{K}_b$. $\|\mathbf{K}_a\|_{\mathrm{F}}$ is the Frobenius norm of $\mathbf{K}_a$.

Specifically, in (2), the first term of the target function indicates the self-reconstruction error of the combined kernel $\mathbf{K}_\mu$. In this setting, each column of $\mathbf{K}_\mu$ is treated as a sample. The second term is a noise simulation term. It is utilized to improve the robustness of $\mathbf{Z}$ against Gaussian noise [43].

It can also improve the quality of the condition number for better calculation of the matrix inverse. The third term is the diversity-inducing term. As discussed in [12], imposing smaller weights on the redundant kernels and improving the variety of information sources are crucial for MKC. To achieve this intuition, we model the kernel correlation through matrix $\mathbf{M}$ and encourage the formulation to impose higher weights on the kernels that have a smaller correlation with others and impose smaller weights on the highly correlated ones. Moreover, to enforce the samples from the same clusters to reconstruct themselves, an exact-rank constraint is introduced (the first constraint). As a consequence, the kernel reconstruction matrix $\mathbf{Z}$, which has integrated the information from multiple base kernels, is forced to be block diagonal. The unit constraint on the kernel combination coefficient $\boldsymbol{\mu}$ is introduced to discard trivial solutions. α and β are the hyperparameters that balance the importance of the kernel reconstruction term and the two generalization terms. In summary, through this formulation, we: 1) extract and fine-tune the intrinsic cluster structure by calculating a common block diagonal kernel reconstruction matrix $\mathbf{Z}$ through an exact-rank-constrained subspace segmentation and 2) find the optimal linear combination by minimizing the sample reconstruction error and the kernel correlation.

D. Optimization Algorithm

Because the rank constraint is nonconvex and discrete, the optimization of our proposed algorithm is difficult. In many of the existing algorithms, for the sake of optimization simplicity, a nuclear norm [44] is adopted to replace the rank constraint as an approximation. However, as discussed in [45], the performance will be adversely influenced due to the inaccurate estimation. In this section, to solve the difficult optimization problem, we design a two-step iterative optimization algorithm with proven convergence to solve the resulting problem in (2). In each iteration, an exact-rank-constrained ridge regression problem and a quadratic programming (QP) problem is solved in turn.

Update $\mathbf{Z}$ *With Fixed* $\boldsymbol{\mu}$*:* With $\boldsymbol{\mu}$ fixed, the optimization problem can be simplified as

$$\min_{\mathbf{Z}} \|\mathbf{K}_\mu - \mathbf{K}_\mu \mathbf{Z}\|_{\mathrm{F}}^2 + \alpha \|\mathbf{Z}\|_{\mathrm{F}}^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{Z}) = l. \tag{3}$$

Equation (3) can be written as

$$\min_{\mathbf{Z}} \quad \operatorname{Tr}\left(\mathbf{A}\mathbf{Z}\mathbf{Z}^\top - 2\mathbf{K}_\mu^2 \mathbf{Z}\right), \quad \text{s.t. } \operatorname{rank}(\mathbf{Z}) = l \tag{4}$$

where $\mathbf{A} = {\mathbf{K}_\mu}^2 + \alpha \mathbf{I}_S$ and $\mathbf{I}_S$ is an S-order identity matrix.

To eliminate the discrete and nonconvex rank constraint, we take advantage of the exact-rank constraint and introduce two matrices, i.e., $\mathbf{G}, \mathbf{H} \in \mathbb{R}^{S\times l}$, to replace $\mathbf{Z}$ as $\mathbf{G}\mathbf{H}^\top$. In these matrices, $\mathbf{H}$ is an orthogonal matrix, i.e., $\mathbf{H}^\top\mathbf{H} = \mathbf{I}$. Then, substituting $\mathbf{Z}$ in (4) with $\mathbf{G}\mathbf{H}^\top$, we eliminate the rank constraint and obtain the following formulation:

$$\begin{aligned} &\min_{\mathbf{G},\mathbf{H}} \operatorname{Tr}(\mathbf{A}\mathbf{G}\mathbf{G}^\top - 2{\mathbf{K}_\mu}^2 \mathbf{G}\mathbf{H}^\top) \\ &\text{s.t.} \mathbf{G}, \mathbf{H} \in \mathbb{R}^{S\times l}, \mathbf{H}^\top\mathbf{H} = \mathbf{I}. \end{aligned} \tag{5}$$

Algorithm 1 MKC With Neighbor-Kernel Subspace Segmentation

Input:

Base kernel set $\{\mathbf{K}_i^o\}_{i=1}^p$. Hyperparameters α, β. The number of nearest neighbors and the expected rank of $\mathbf{Z}$.

Output:

Kernel combination weight $\boldsymbol{\mu}$ and the reconstruction matrix $\mathbf{Z}$;

Generate the corresponding neighbor-kernel set $\{\mathbf{K}_i\}_{i=1}^p$ and set $t = 1$;
repeat
  Calculate $\mathbf{K}^{(t)} = \sum_{i=1}^p \boldsymbol{\mu}_i^{(t)} \mathbf{K}_i$
  Calculate $\mathbf{H}^{(t)}$ by optimizing Eq. (6);
  Calculate $\mathbf{Z}^{(t)} = \mathbf{A}^{-1}{\mathbf{K}^{(t)}}^2 \mathbf{H}^{(t)} {\mathbf{H}^{(t)}}^\top$;
  Calculate $\boldsymbol{\mu}^{(t)}$ by solving the QP problem in Eq. (9);
  $t = t + 1$.
until $|\mathrm{Obj}^{(t)} - \mathrm{Obj}^{(t-1)}| < 10^{-4} \times |\mathrm{Obj}^{(t)}|$.

Setting the derivation of formula (5) on $\mathbf{G}$ to zero, we have $\mathbf{G}_* = \mathbf{A}^{-1}\mathbf{K}_\mu^2\mathbf{H}$. Substituting $\mathbf{G}_*$ into (5), we have

$$\begin{aligned} &\max_{\mathbf{H}} \operatorname{Tr}(\mathbf{H}^\top \mathbf{K}_\mu^2 \mathbf{A}^{-1} \mathbf{K}_\mu^2 \mathbf{H}) \\ &\text{s.t.} \mathbf{H} \in \mathbb{R}^{S\times l}, \mathbf{H}^\top\mathbf{H} = \mathbf{I}. \end{aligned} \tag{6}$$

To calculate the optimal $\mathbf{H}$, denoted as $\mathbf{H}_*$, we perform KPCA on $\mathbf{K}_\mu^2\mathbf{A}^{-1}\mathbf{K}_\mu^2$ and extract the eigenvectors according to the largest l eigenvalues of the matrix. Then, substituting $\mathbf{G}_*$ and $\mathbf{H}_*$ back into $\mathbf{Z}$, the solution of (4) can be obtained by calculating

$$\mathbf{Z}_* = \mathbf{G}_*\mathbf{H}_*^\top = \mathbf{A}^{-1}\mathbf{K}_\mu^2 \mathbf{H}_*\mathbf{H}_*^\top. \tag{7}$$

Update $\boldsymbol{\mu}$ *With Fixed* $\mathbf{Z}$*:* Given $\mathbf{Z}$, the original optimization problem in (2) can be simplified as

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \|\mathbf{K}_\mu - \mathbf{K}_\mu \mathbf{Z}\|_{\mathrm{F}}^2 + \beta \boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu} \\ &\text{s.t.} \boldsymbol{\mu} \geq 0, \ \|\boldsymbol{\mu}\|_1 = 1, \quad \mathbf{K}_\mu = \sum_{i=1}^p \mathbf{K}_i \cdot \boldsymbol{\mu}_i. \end{aligned} \tag{8}$$

Equation (8) can be rewritten as

$$\min_{\boldsymbol{\mu}} \boldsymbol{\mu}^\top (\beta\mathbf{M} + \mathbf{M}^*)\boldsymbol{\mu}, \quad \text{s.t. } \boldsymbol{\mu} \geq 0, \ \|\boldsymbol{\mu}\|_1 = 1 \tag{9}$$

where $\mathbf{M}_{ab}^* = \operatorname{Tr}(\mathbf{K}_a(\mathbf{Z} - \mathbf{I}_S)(\mathbf{K}_b(\mathbf{Z} - \mathbf{I}_S))^\top)$. This is a typical QP problem with linear constraints. It can be easily solved with the optimization toolbox in MATLAB.

We summarize our optimization algorithm for (2) in Algorithm 1. Generally, in each iteration, the reconstruction matrix $\mathbf{Z}$ and the kernel weights $\boldsymbol{\mu}$ are iteratively optimized. The algorithm stops when the variation of the objective value of (2) (denoted as *Obj*) reaches a preset threshold (10^{-4}).

E. Discussion

In this section, we first compare the solution of ridge regression with and without the exact-rank constraint and shed light on the intrinsic meaning of the rank constraint in the formulation. Then, we analyze the optimality of our proposed

solution for the exact-rank-constrained ridge regression problem. Finally, we further analyze the convergence and computational complexity of the proposed iterative optimization algorithm.

1) Insight Into the Solution of Exact-Rank-Constrained Ridge Regression: We reveal the essence of the exact-rank constraint in ridge regression by comparing the solution with and without the constraint. Without the rank constraint, the formulation of (3) is simple

$$\min_{\mathbf{Z}} \|\mathbf{K}_\mu - \mathbf{K}_\mu \mathbf{Z}\|_{\mathrm{F}}^2 + \alpha \|\mathbf{Z}\|_{\mathrm{F}}^2. \tag{10}$$

Its global optimal solution can be quickly obtained by calculating $\mathbf{Z} = (\mathbf{K}_\mu^2 + \alpha \mathbf{I}_S)^{-1}\mathbf{K}_\mu^2 = \mathbf{A}^{-1}\mathbf{K}_\mu^2$. When the rank constraint is added, as discussed in Section II-D, the solution of (3) becomes $\mathbf{Z}_* = \mathbf{A}^{-1}\mathbf{K}_\mu^2\mathbf{H}_*\mathbf{H}_*^\top$. The only difference between these two solutions is the matrix $\mathbf{H}_*\mathbf{H}_*^\top$. In this term, the multiplication of $\mathbf{H}_*$ first projects the solution of ridge regression to the informative directions to keep the discriminative information for clustering and discard the within-cluster details that might confuse the algorithm during clustering. Then, since $\mathbf{H}_*$ is orthogonal, its transpose matrix inversely maps the samples back to its original feature space. For more detailed information about the mechanism of backprojection, please refer to [4]. In summary, the exact-rank constraint in ridge regression forces the unconstrained formulation to project its solution onto the more discriminative directions and then back-project it to the original feature space.

2) Solution Optimality: In this section, we prove that our solution for the exact-rank-constrained rigid regression problem is equivalent to the global minimizer of [45]. The deduction is straightforward. We simply compare the equivalence of these two solutions by subtracting one from the other. For the convenience of the following proof, we denote the solution of our proposed algorithm and the solution proposed in [45] as $\mathbf{Z}_1$ and $\mathbf{Z}_2$, respectively. Additionally, we decompose the symmetric positive definite matrix $\mathbf{A}$ as $\mathbf{C}\mathbf{C}^\top = (\mathbf{U}_A\mathbf{D}_A^{1/2})(\mathbf{U}_A\mathbf{D}_A^{1/2})^\top$, where $\mathbf{U}_A$ is an orthogonal matrix that contains the eigenvectors of $\mathbf{A}$, and $\mathbf{D}_A$ is a diagonal matrix whose diagonal values are the eigenvalues of $\mathbf{A}$. With these definitions, (4) can be rewritten as

$$\min_{\mathbf{Z}} \left\|\mathbf{C}^\top\mathbf{Z} - \mathbf{C}^{-1}\mathbf{K}_\mu^2\right\|_{\mathrm{F}}^2, \quad \text{s.t. } \mathrm{rank}(\mathbf{Z}) = l. \tag{11}$$

Denoting $\mathbf{B} = \mathbf{C}^{-1}\mathbf{K}_\mu^2$, according to the conclusion in [45], the global minimizer of (11) is: $\mathbf{Z}_2 = {\mathbf{C}^{-1}}^\top\mathbf{U}\mathbf{D}_l\mathbf{V}^\top$, where $\mathbf{D}_l$ consists of the largest l singular values of $\mathbf{B}$ given the SVD of $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. Then, subtracting $\mathbf{Z}_1$ [defined in (7)] from $\mathbf{Z}_2$, we have

$$\mathbf{Z}_1 - \mathbf{Z}_2 = {\mathbf{C}^{-1}}^\top\mathbf{C}^{-1}\mathbf{K}_\mu^2\mathbf{H}_*\mathbf{H}_*^\top - {\mathbf{C}^{-1}}^\top\mathbf{U}\mathbf{D}_l\mathbf{V}^\top. \tag{12}$$

According to (6), $\mathbf{H}_*$ is the matrix that consists of the eigenvectors corresponding to the largest l eigenvalues of matrix $\mathbf{K}_\mu^2\mathbf{A}^{-1}\mathbf{K}_\mu^2 = \mathbf{B}^\top\mathbf{B}$. As a consequence, according to the definition of $\mathbf{V}$ and $\mathbf{H}_*$, we have $\mathbf{V}^l = \mathbf{H}_*$. Here, $\mathbf{V}^l \in \mathbb{R}^{S\times l}$ is composed of the right singular vectors that correspond to the largest l singular values of $\mathbf{B}$. Denoting $\mathbf{V}_*^l \in \mathbb{R}^{S\times S}$ as the concatenation of $\mathbf{V}^l$ and a zero-matrix: $[\mathbf{V}^{l\top}; \mathbf{0}_{(\mathrm{S}-l)\times \mathrm{S}}]$, $\mathbf{Z}_1 - \mathbf{Z}_2$ can be transformed as

$$\begin{aligned}\mathbf{Z}_1 - \mathbf{Z}_2 &= {\mathbf{C}^{-1}}^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{H}_*\mathbf{H}_*^\top - {\mathbf{C}^{-1}}^\top\mathbf{U}\mathbf{D}^l\mathbf{V}^\top \\ &= {\mathbf{C}^{-1}}^\top\mathbf{U}\mathbf{D}\mathbf{V}_*^{l\top} - {\mathbf{C}^{-1}}^\top\mathbf{U}\mathbf{D}^l\mathbf{V}^\top = \mathbf{0}_{S\times S}.\end{aligned}$$

Since the solutions of the two methods are equal, our proposed algorithm can also achieve the global optimal solution for (3).

3) Convergence Analysis: In this section, we prove the convergence of the proposed optimization algorithm. To clarify this point, we first define the objective function of the optimization problem as

$$\mathcal{J}(\mathbf{Z}, \boldsymbol{\mu}) = \Big\{ \min_{\mathbf{Z},\boldsymbol{\mu}} \|\mathbf{K}_\mu - \mathbf{K}_\mu\mathbf{Z}\|_{\mathrm{F}}^2 + \alpha\|\mathbf{Z}\|_{\mathrm{F}}^2 + \beta\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}, \quad s.t. \mathrm{rank}(\mathbf{Z}) = l,\ \boldsymbol{\mu} \geq \mathbf{0},\ \|\boldsymbol{\mu}\|_1 = 1\Big\}. \tag{13}$$

As seen from (13), jointly optimizing $\mathbf{Z}$ and $\boldsymbol{\mu}$ is difficult. Instead, in Section II-D, we developed a two-step alternative algorithm to solve it. During the optimization, we fix one variable and optimize the other one. Specifically, in the tth iteration, when $\boldsymbol{\mu}$ is fixed as $\boldsymbol{\mu}^{(t)}$, we have proven in Section II-E2 that our algorithm can achieve the global minimizer of $\mathcal{J}(\mathbf{Z}, \boldsymbol{\mu}^{(t)})$. As a consequence, we have

$$\mathcal{J}(\mathbf{Z}^{(t)}, \boldsymbol{\mu}^{(t)}) \geq \mathcal{J}(\mathbf{Z}^{(t+1)}, \boldsymbol{\mu}^{(t)}). \tag{14}$$

With fixed $\mathbf{Z}^{(t+1)}$, the optimization problem $\mathcal{J}(\mathbf{Z}^{(t+1)}, \boldsymbol{\mu})$ is a typical QP problem with a convex constraint [see (9)]. We can prove the convexity of this problem by proving that $\mathbf{M} + \mathbf{M}^*$ is PSD. The PSD property of the kernel correlation matrix $\mathbf{M}$ is proven in [42]. In Proposition 1, we will prove that the matrix $\mathbf{M}^*$ is also PSD.

Proposition 1: The symmetric matrix $\mathbf{M}^*$ in (9) is PSD.

Proof: For any vector $\mathbf{x} \in \mathbb{R}^p$

$$\begin{aligned}\mathbf{x}^\top\mathbf{M}^*\mathbf{x} &= \sum_{a,b=1}^{p} \mathbf{x}_a\mathbf{x}_b\mathbf{M}_{ab}^* \\ &= \mathrm{Tr}\left(\sum_{a,b=1}^{p} \mathbf{x}_a\mathbf{x}_b\mathbf{K}_a(\mathbf{Z} - \mathbf{I}_S)(\mathbf{Z} - \mathbf{I}_S)^\top\mathbf{K}_b\right) \\ &= \mathrm{Tr}\left(\left(\sum_{a=1}^{p} \mathbf{x}_a\mathbf{K}_a(\mathbf{Z} - \mathbf{I}_S)\right)\left(\sum_{b=1}^{p} \mathbf{x}_b\mathbf{K}_b(\mathbf{Z} - \mathbf{I}_S)\right)^\top\right) \\ &= \left\|\sum_{a=1}^{p} \mathbf{x}_a\mathbf{K}_a(\mathbf{Z} - \mathbf{I}_S)\right\|_{\mathrm{F}}^2 \geq 0.\end{aligned}$$

□

Since matrix $\mathbf{M} + \mathbf{M}^*$ is PSD, the corresponding QP problem is convex and has a global optimal solution. Denoting this solution as $\boldsymbol{\mu}^{(t+1)}$, we have

$$\mathcal{J}(\mathbf{Z}^{(t+1)}, \boldsymbol{\mu}^{(t)}) \geq \mathcal{J}(\mathbf{Z}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}). \tag{15}$$

By combining (14) and (15), we have

$$\mathcal{J}(\mathbf{Z}^{(t)}, \boldsymbol{\mu}^{(t)}) \geq \mathcal{J}(\mathbf{Z}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}) \tag{16}$$

which indicates that the objective function of our algorithm in (13) monotonically decreases with the increase of iterations.

Moreover, the objective function in (13) is lower bounded by zero. Therefore, we conclude that the proposed algorithm is theoretically guaranteed to converge to a local minimum.

4) Computational Complexity Analysis: In this section, we provide the computational complexity analysis of our proposed optimization algorithm. In each iteration, the cost of updating $\mathbf{H}_*$ with SVD is $\mathcal{O}(S^3)$. Calculating $\mathbf{Z}$ with (7) also has a complexity cost of $\mathcal{O}(S^3)$. To update $\boldsymbol{\mu}$, the time consumption of a convex linear-constrained QP problem is $\mathcal{O}(Lp^3)$, where L is the size of the problem encoded as binary and p is the number of base kernels [46]. In summary, because, in each iteration, a quadratic program, an SVD, a matrix inverse, and six matrix multiplication operations are conducted, the total computational complexity of our algorithm is $\mathcal{O}(t(Lp^3 + S^3))$, where t is the number of iterations. Through our empirical trials, we find that this value is usually smaller than 15 in most circumstances. Compared with kernel clustering algorithms with a single kernel or fixed kernel weights, the proposed algorithm is less efficient due to the extra time consumed to find appropriate kernel weights and to perform noise filtering. However, for the sake of the proposed simple optimization algorithm, it is still more efficient than other local sample adaptive MKC algorithms, such as localized multiple kernel k-means (LMKKM) [10], and comparable to those fast algorithms, such as robust multiple kernel k-means (RMKKM) (with a time complexity of $\mathcal{O}(S^2\,dp+(S^3+S^2+S)pt)$ [11]) and robust MKC (RMKC) [with a time complexity of $\mathcal{O}((S^2p + S^3)t)$]. Here, d is the feature dimension in the original feature space.

III. Experiments

In this section, to evaluate the effectiveness of our proposed MKC algorithm, especially the efficacy of the neighbor kernels, four experiments are designed. In the first experiment, we construct a synthetic data set to test the robustness against noise and outliers of the proposed neighbor kernel. Second, we compare our proposed algorithm with nine state-of-the-art MKC algorithms on real-world data sets to evaluate its performance. Then, we test the sensitivity of the algorithm against the main hyperparameters. Finally, we apply neighbor kernels to the existing MKC algorithms and test the capacity of the proposed kernel on enhancing the performance of these methods.

Following the settings in [42], we centralize each base kernel and then normalize it to keep the diagonal elements of these kernels as one. In our experiments, three widely used criteria, i.e., accuracy (ACC), normalized mutual information (NMI), and purity, are adopted to evaluate the performance of the compared MKC methods. For the methods that output a unified kernel matrix, we conduct kernel k-means to evaluate their performance. For the methods that output an affinity matrix or a reconstruction matrix $\mathbf{Z}$, spectral clustering with the input of $(|\mathbf{Z}| + |\mathbf{Z}^\top|)/2$ will be adopted to conduct clustering. For all algorithms, we repeat each experiment 50 times with random initialization to reduce the effect of randomness caused by k-means and report the best result.

A. Evaluation of the Effectiveness of Neighbor Kernels

In this section, a synthetic data set is constructed to evaluate the robustness of the proposed neighbor kernels against noise and outliers. The main idea of the experiment is to compare the performance variation of the original kernels and the corresponding neighbor kernels when noise within the data increases. The synthetic data are generated by three steps. First, we generate 600 unit samples evenly with the standard normal distribution from three independent subspaces, each of which is extended by four independent components. As a consequence, the original synthetic data set is a 12-D data set with 600 samples and three categories. We repeat the same operation three times to simulate three different views of the samples. After that, we randomly add white Gaussian noise $\mathbf{N}(0, 1)$ to 40% of randomly selected samples in each view. The energy level of the noise is increased from 0.05 to 0.5 to simulate both noise and outliers within data. Finally, we generate two kernels for each view. One is a linear kernel and the other is a Gaussian kernel with the bandwidth equal to the average distance among samples in the corresponding view. As a consequence, there are six original kernels in each data set. The synthetic neighbor kernels are constructed according to the description in Section II-A.

To compare the discriminative capacity and the robustness of original kernels and neighbor kernels, we report the performance of the single best kernels and the average kernels for comparison. Specifically, in the single best kernel selection mechanism, we conduct kernel k-means on each kernel alone and report the best performance. In the average kernel combination mechanism, the base kernels are combined linearly with equal weights to integrate information from different views for clustering. We repeat the experiment ten times to alleviate the influence of randomness, and the average result is reported in the experiment.

Fig. 2 shows the performance of the compared algorithms. From the variation curves, we can clearly find several consistent observations as follows.

1) The performance of all the compared methods decreases with the increase of noise magnitude. However, comparatively, methods using neighbor kernels perform consistently better than those using original kernels in both the single best kernel and the average kernel methods.
2) In the front part of the curves, which corresponds to experimental results with relatively low-level noise, the performance of the average neighbor kernel maintains a 20% advantage over the second-best method, indicating good robustness against noise.
3) In the latter part of the curves, the gaps between different methods decrease since the noise in samples gradually dominates the distribution, making some of them become outliers in the data set.

Nevertheless, even in this circumstance, the neighbor-kernel-based methods still outperform the original kernel counterparts, indicating good robustness against outliers.

In general, the proposed neighbor kernels are robust against noise and outliers because of the intrinsic weighting mechanism of the kernels. By keeping the more reliable similarities

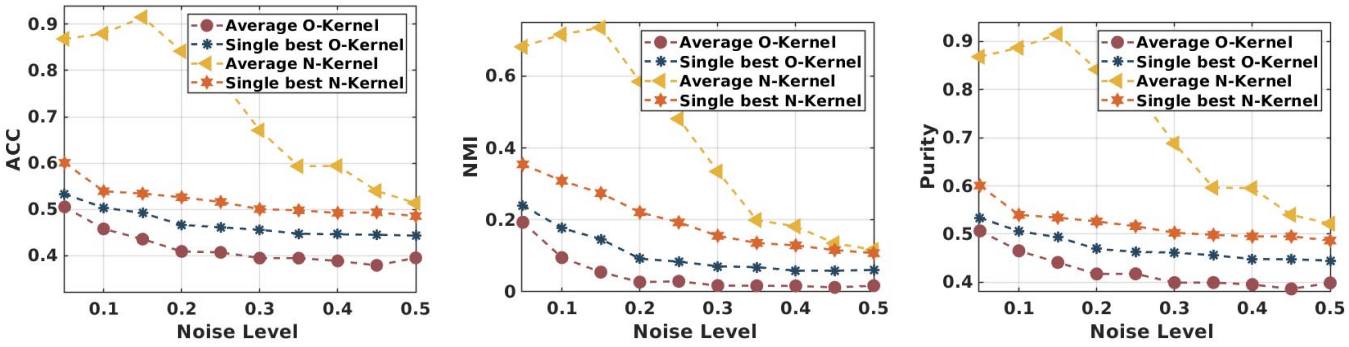


Fig. 2. Clustering performance comparison between original kernels and neighbor kernels against the variation of the noise level. In this figure, the yellow, orange, brown, and blue dotted lines indicate the performance of the average neighbor kernel, best single neighbor kernel, average original kernel, and best single original kernel, respectively.

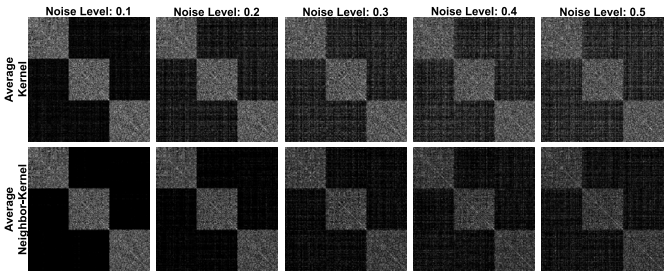


Fig. 3. Illustration of the variation of average kernels (first row) and average neighbor kernels (second row) against different magnitudes of noise. The noise level increases from 0.1 to 0.5.

TABLE I
BENCHMARK DATA SETS

Datasets	# Samples	# Kernels	# Clusters	# Data Type
BBCSport2	554	2	5	News article
CCV	6773	6	20	Video event
Plant	940	69	4	Protein sequence
ProteinFold	694	12	27	Protein sequence
PsortPos	541	69	4	Protein sequence
Nonpl	2732	69	3	Protein sequence
UCI-Digit	2000	3	10	Image
Mfeat	2000	12	10	Image
Flower102	8189	4	102	Image
Caltech101mit	1530	25	102	Image
Flower17-DL	1360	3	17	Image

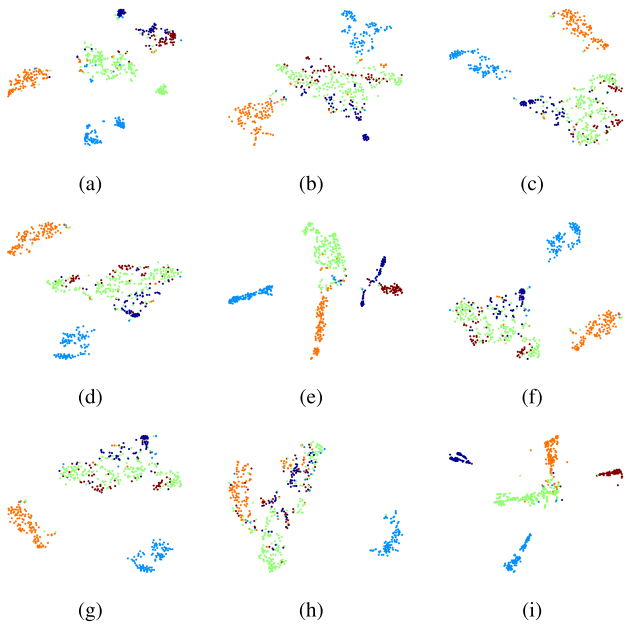


Fig. 4. Visualization of the revealed cluster structure of the compared algorithms with t-SNE [51] on the BBCSport data set. (a) SB-KKM. (b) MKKM. (c) RMKKM. (d) LMKKM. (e) RMSC. (f) RMKC. (g) MKKM-MR. (h) LAMKC. (i) Proposed.

among neighbors and abandoning those that go beyond the observation of samples (because of the sample distribution or noise, and so on), the neighbor kernels are able to keep the most reliable information and filter the less confident portion. To better illustrate this point of view, we further record the variation of the synthetic average kernel and the average neighbor kernel against the increase of the noise level. As seen from Fig. 3, the added noise quickly undermined the standard average kernel and corrupted the cluster structure of those kernels (starting from noise level 0.1). Comparatively, the proposed average neighbor kernel performs more robustly against the injected noise information, with a higher robustness bar and smaller influence.

B. Comparison With the State-of-the-Art Algorithms

In this section, to verify the effectiveness of our proposed algorithm on real-world data, we compare it with nine state-of-the-art MKC algorithms on 11 popular benchmark data sets. These data sets are collected from various applications, including natural language processing (BBCSports2)[1] protein function prediction (ProteinFold and PsortPos)[2] image recognition (Flower102[3] Caltech101mit)[4] and video analysis (CCV).[5] The sample, kernel, and cluster numbers of the data sets range from 554 to 8189, 2 to 69, and 3 to 102, respectively. All these data sets form abundant and comprehensive testing environments for the compared algorithms. It is worth noting that, in this paper, we even try to utilize our proposed algorithm to fuse the features generated by different

[1] http://mlg.ucd.ie/datasets/bbc.html
[2] http://www.raetschlab.org/suppl/protsubloc
[3] http://www.robots.ox.ac.uk/$\tilde{\text{v}}$gg/data/flowers/102/
[4] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[5] http://www.ee.columbia.edu/ln/dvmm/CCV/

TABLE II

ACC, NMI, AND PURITY COMPARISON OF DIFFERENT CLUSTERING ALGORITHMS ON 11 BENCHMARK DATA SETS. BOLDFACE: BEST PERFORMANCE AMONG ALL COMPARED ALGORITHMS. ITALICS: SECOND-BEST PERFORMANCE

Datasets	A-MKKM	SB-KKM	MKKM [50]	RMKKM [11]	LMKKM [10]	RMSC [17]	RMKC [19]	MKKM-MR [12]	LAMKC [33]	Proposed kernel	Proposed Z
ACC (%)											
BBCSport2view	66.18	76.65	66.18	63.79	66.18	*86.03*	66.18	66.18	60.48	64.71	**95.04**
CCV	19.71	23.82	19.95	17.57	21.84	16.11	19.71	*24.20*	19.46	22.80	**25.22**
Flower17-DL	74.34	70.81	75.51	71.99	*74.56*	58.75	74.34	73.90	66.18	68.90	**78.31**
MFea	71.95	80.20	59.60	65.30	71.35	*84.15*	72.00	83.20	80.90	91.35	**98.00**
Plant	*61.49*	51.60	56.38	55.53	58.30	53.62	*61.49*	52.45	47.34	51.49	**62.66**
ProteinFold	28.10	33.86	27.23	33.29	30.26	33.00	28.82	*36.31*	33.00	**37.61**	34.87
PsortPos	57.12	*69.13*	60.44	60.81	56.01	51.94	57.12	50.83	56.01	**69.87**	68.76
UCI-Digit	88.75	75.65	47.00	44.00	89.90	80.65	88.90	90.40	*95.50*	96.80	**96.95**
Flower102	27.29	33.22	21.96	27.05	26.98	32.97	34.03	*42.24*	42.01	42.61	**42.80**
Nonpl	49.38	57.50	49.30	*62.77*	-	60.65	49.38	56.59	55.78	**71.41**	62.59
Caltech101mit	35.29	32.09	34.77	32.03	34.44	29.67	35.29	**37.91**	31.90	*37.58*	36.01
NMI (%)											
BBCSport2view	53.93	58.96	53.93	39.62	54.22	*73.89*	53.83	53.93	45.57	48.12	**84.94**
CCV	17.50	19.05	15.41	13.79	18.43	14.54	17.50	*19.84*	17.42	20.25	**22.57**
Flower17DL	73.87	69.14	73.73	71.56	*73.97*	62.44	73.87	73.42	71.35	72.40	**78.75**
Meat	69.68	66.55	55.56	62.67	71.52	*81.69*	69.70	78.12	80.50	84.38	**95.27**
Plant	*26.57*	17.40	20.02	19.39	22.07	23.18	*26.57*	21.56	17.30	24.22	**33.49**
ProteinFold	38.53	42.03	37.16	40.17	40.26	43.91	39.46	*45.89*	41.25	**48.68**	46.93
psortPos	28.86	*42.54*	35.01	36.54	28.11	30.69	28.86	26.74	27.27	41.72	**48.82**
UCI-Digit	80.59	68.44	48.16	48.02	82.85	79.42	80.88	83.22	*90.08*	92.66	**92.91**
Flower102	46.32	49.08	42.30	46.99	45.92	53.36	49.80	57.57	*58.19*	58.64	**59.69**
Nonpl	16.55	15.29	14.94	17.34	-	*20.35*	16.55	15.51	11.53	26.60	**29.13**
Caltech101mit	59.93	58.30	59.64	56.21	58.92	57.09	59.93	*61.47*	58.19	**61.84**	60.69
Purity (%)											
BBCSport2view	77.21	79.41	77.21	67.83	77.39	*86.03*	77.21	77.21	70.77	71.51	**95.04**
CCV	24.45	25.29	24.58	21.32	25.79	20.49	24.45	*26.58*	24.13	27.02	**28.41**
Flower17-DL	74.56	70.96	75.51	73.68	*74.78*	61.69	74.56	74.19	69.19	70.66	**80.00**
MFeat	71.95	80.20	62.55	66.25	72.05	*84.10*	72.00	83.20	84.10	91.35	**98.00**
Plant	*61.49*	56.38	56.38	55.53	58.30	59.47	*61.49*	58.72	57.45	56.28	**65.21**
ProteinFold	36.17	41.21	33.86	37.61	37.18	42.36	36.46	*45.39*	37.90	**46.69**	44.38
PsortPos	60.81	*71.35*	66.54	64.33	60.26	58.60	60.81	60.81	58.04	**71.53**	**71.53**
UCI-Digit	88.75	76.30	49.70	47.20	89.90	82.90	88.90	90.40	*95.50*	96.80	**96.95**
Flower102	32.28	38.88	27.61	32.13	32.37	40.24	39.96	48.49	48.36	49.38	**50.04**
Nonpl	*72.18*	71.23	71.27	71.71	-	70.50	*72.18*	63.91	61.38	73.46	**77.34**
Caltech101mit	37.52	33.92	37.25	33.79	35.88	31.31	37.52	**39.74**	34.25	39.41	*39.54*

deep learning architectures to serve for clustering. In the experiment, we choose Flower 17 as a representative. Specifically, we extract the final fully connected layers of three pretrained deep convolutional neural networks, i.e., AlexNet [47], VGG [48], and GoogLeNet [49], and construct three linear kernels. The detailed information of all the data sets is listed in Table I.

The compared algorithms include average multiple kernel k-means (A-MKKM), single best kernel k-means (SB-KKM), multiple kernel k-means (MKKM) [50], RMKKM [11], LMKKM [10], robust multiview spectral clustering (RMSC) [17], RMKC [19], multiple kernel k-means clustering with matrix-induced regularization (MKKM-KR) [12], and MKC with local kernel alignment maximization (MKC-LKAM) [33]. All the MATLAB implementations of the compared algorithms are downloaded from web pages or acquired from the corresponding authors. The parameter settings of these algorithms also follow the suggestion of the corresponding literature. Regarding our method, the constrained rank value and the importance of the Frobenius term are fixed as $0.1S$ and $10^{(-4)}\|\mathbf{K}^*_{\text{avg}}\|_{\text{F}}$ in all the experiments, respectively. Here, $\mathbf{K}_{\text{avg}}$ is the average neighbor kernel. The other two parameters, i.e., the kernel diversity balancing term and the number of neighbors are set with grid search in a small range of $\{2^{-8}, 2^{-2}, 2^2, 2^6\}$, and $\{0.01, 0.03, 0.09, 0.11\}$, respectively. Note that the memory consumption of the LMKKM algorithm is directly proportional to $(S \times p)^2$ [10], where S and p are the sample number and the base kernel number, respectively. It is easy to run out of memory when the data possess a large number of samples and base kernels. As a consequence, the result of LMKKM is not provided on the Nonpl data set.

TABLE III

AVERAGE COMPUTATIONAL TIME CONSUMPTION OF THE COMPARED ALGORITHMS

Methods	AMKKM	SB-KKM	MKKM	RMKKM	LMKKM
Time (s)	2.05	12.06	5.69	613.2	10542
Methods	RMSC	RMKC	MKKM-MR	LAMKC	Proposed
Time (s)	1144	303	18.57	468.5	211.1

Results and Analysis: We summarize the clustering performance of the compared methods in items of three metrics, i.e., ACC, NMI, and purity, on the 11 data sets in Table II. For the computational time comparison, in Table III, we report the average time consumption of the compared algorithms of ten data sets on which the results of all algorithms are available.

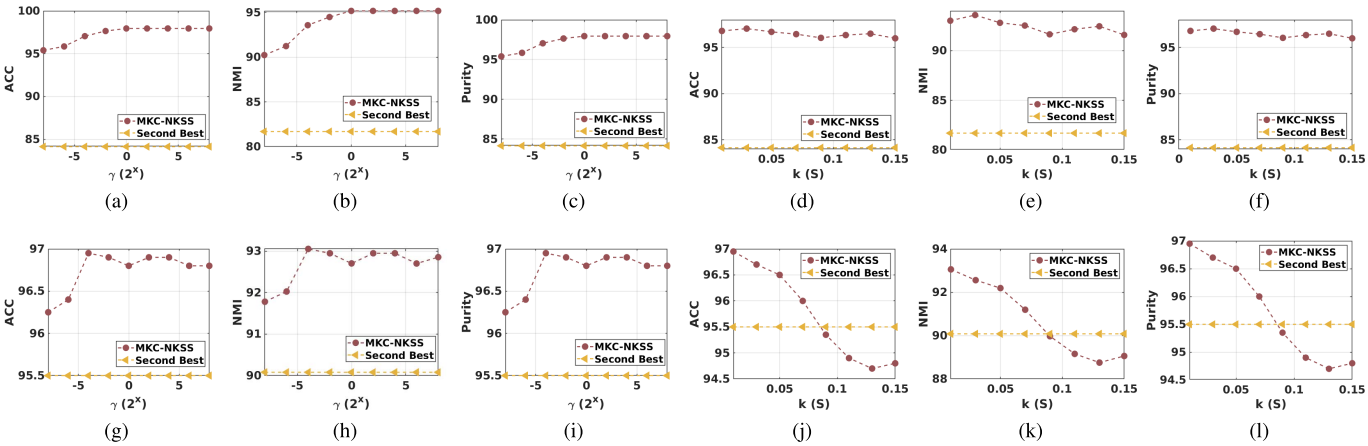


Fig. 5. Sensitivity testing. The sensitivity of the kernel diversity balancing term β (left three columns) and the neighbor number k (right three columns) are tested. The first and the second row correspond to the results on MultiFeature and UCI-Digit, respectively. The unit of k is S, i.e., the total sample number of a data set. (a) and (g) ACCU. (b) and (h) NMI. (c) and (i) Purity. (d) and (j) ACCU. (e) and (k) NMI. (f) and (l) Purity.

This testing is based on the Ubuntu 16.04 operating system with i7 8700K CPUs, 64-Gb memory, and the MATLAB 2018a environment. Moreover, to provide a more direct and concrete evaluation of the clustering results of different methods, we adopt t-SNE [51] to visualize the sample distribution generated by the compared algorithms on a representative data set (BBCSports). From the table and the figures, we have the following observations.

First, although many compared algorithms perform reasonably well, in most of the data sets, our proposed algorithm still outperforms the other state-of-the-art algorithms. It reflects the superior performance of the proposed algorithm in a variety of applications. Moreover, the good performance of MKKM-MR [12] and RMSC [17] indicates the importance of increasing the information diversity and enhancing the shared cluster structure with the low-rank constraint in MKC, respectively. Comparatively, generating sample-specific kernel weights tends to be more memory-consuming but has a limited performance improvement in the compared data sets. Second, in the proposed algorithm, the learned affinity matrix performs comparable to, if not better than, the learned kernel in most of the circumstances. This phenomenon indicates the effectiveness of the integrated subspace segmentation on fine-tuning the cluster structure of the linearly combined kernel. Third, the average kernel and the single best kernel provide two strong benchmark methods and they perform even better than many of the well-designed MKC algorithms in many data sets. This supports our intuition of selecting the average kernel as the metric to determine the neighbors of samples. Fourth, the trial of constructing base kernels with a deep neural network generated features that achieved a large performance enhancement against the competitors using manually designed base kernels (approximately 10% improvement on average on ACC against the results reported in [33]).

Regarding the computational consumption, the results reported in Table III are consistent with the analysis in Section II-E4. As can be seen, in addition to significantly improving the clustering performance of existing state-of-the-art algorithms, such as RMKKM [11], LMKKM [10], RMSC [17], RMKC [19], MKKM-MR [12], and LAMKC [33], the proposed algorithm does not significantly increase the computational cost. In Fig. 4, by observing the cluster structure revealed by different algorithms, we can find that the clusters generated by our algorithm are more compact and separable than those of the others. RMSC also provides a good performance, which is consistent with the result in Table II.

C. Convergence and Sensitivity

To test the sensitivity of our proposed algorithm against the hyperparameters, in this section, we report the performance variation curves of two parameters, i.e., the kernel diversity balancing term β and the number of neighbors k. The ACCU, NMI, and purity variational curves of these two parameters are compared with the second-best performance on the corresponding data sets. As seen in Fig. 5, our proposed algorithm is stable against the variation of parameters and remains better than the second-best algorithm in a large range, indicating the effectiveness and stability of our proposed algorithm.

In Fig. 5, two different tendencies are witnessed regarding the performance variation against the neighbor numbers. Specifically, equivalent or better performance was achieved when larger k is adopted on the MultiFeature data set. However, this tendency reverses on the UCI-Digit data set. Different properties of data sets cause this phenomenon. Generally, if including more neighbors can provide more useful information than indiscriminative information, the performance will increase. However, if more noise is included, the performance will decrease.

Fig. 6 shows the convergence of the proposed algorithm by plotting the objective value in each iteration. As observed, this value is monotonically decreased, and the algorithm usually converges in less than 15 iterations.

D. Applying Neighbor Kernels on Other Methods

Previous experiments have verified that the proposed neighbor kernel preserves a better block diagonal structure and is more robust to noise and outliers. In this section, we show

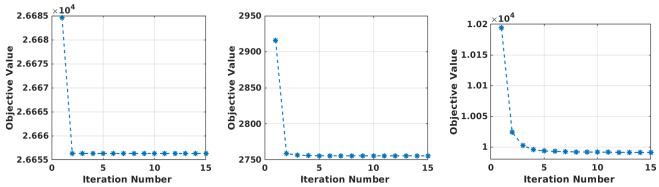


Fig. 6. Objective value of our algorithm at each iteration. The results on (a) UCI-Digit, (b) MultiFeature, and (c) Flower102 data sets are reported.

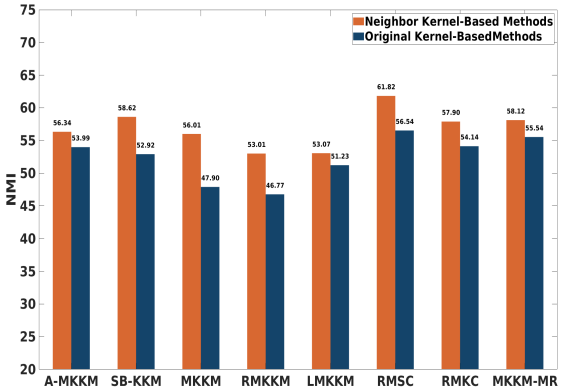


Fig. 7. Illustration of performance enhancement when neighbor kernels are applied to the existing state-of-the-art MKC algorithms.

that this kernel can be easily utilized as a plug-in component to help enhance the performance of the existing state-of-the-art methods. Specifically, to conduct the experiment, we simply compare the performance of the state-of-the-art MKC algorithms with original base kernels and with neighbor kernels as base kernels. In Fig. 7, the bar chart illustrates the average NMI of eight representative data sets, i.e., BBCSport, Caltech101-MIT, Flower17-DL, UCI MultiFeatures, Plant, ProteinFold, PsortPos, and UCI-Digit. From the figure, we can clearly observe a large performance boost on the state-of-the-art algorithms when the original kernels are replaced with the neighbor kernel. It is worth noting that many of the listed algorithms are designed with powerful noise eliminating and block diagonal structure extraction mechanisms. For example, in RMKKM [11], the introduced $\ell_{2,1}$-norm is effective in reducing the adverse effect of outliers; in RMKC, the low-rank and sparse decomposition setting can effectively extract the discriminative structure from multiple kernels [17], and so on. However, the proposed neighbor kernel still improves the performance of these algorithms to a preferable extent, indicating strong complementarity between the state-of-the-art algorithms and the neighbor kernels.

IV. Conclusion

In this paper, we proposed a neighbor-kernel-based subspace segmentation algorithm to better reveal the intrinsic cluster structure shared by the base kernels and eliminate the adverse effect of noise and outliers in MKC. Specifically, we first introduced a novel kernel denoted as neighbor kernel, which possesses a better block diagonal structure preservation capacity and robustness against noise and outliers. Based on the neighbor kernel, we utilized an exact-rank-constrained subspace segmentation algorithm to further refine the hidden clustering structure among samples. An iterative algorithm with proven convergence was proposed to solve the corresponding optimization problem. After that, we theoretically revealed the intrinsic effect of the exact-rank constraint in ridge regression, i.e., it back-projects the solution of the unconstrained problem to its principal components. Experiments on both synthetic and real-world data sets verified the superior performance of our proposed algorithm against other state-of-the-art MKC methods. The experimental results also indicated that the proposed neighbor kernels could be easily applied to enhance the performance of the existing MKC algorithms in a plug-and-play manner. In the future, we plan to integrate the process of neighbor extraction into the pipeline of MKC and find the most reasonable neighbors according to the optimal kernel combination.

References

[1] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. NIPS*, 2005, pp. 1537–1544.
[2] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Proc. ICDM*, Dec. 2009, pp. 1016–1021.
[3] S. Yu *et al.*, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.
[4] A. Kumar and H. Daume, "A co-training approach for multi-view spectral clustering," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, Jul. 2011, pp. 393–400.
[5] Y. Han, K. Yang, Y. Yang, and Y. Ma, "Localized multiple kernel learning with dynamical clustering and matrix regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 486–499, Feb. 2018.
[6] Y. Han, K. Yang, Y. Ma, and G. Liu, "Localized multiple kernel learning via sample-wise alternating optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 137–148, Jan. 2014.
[7] H. Valizadegan and R. Jin, "Generalized maximum margin clustering and unsupervised kernel learning," in *Proc. NIPS*, 2007, pp. 1417–1424.
[8] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. NIPS*, 2011, pp. 1413–1421.
[9] Y. Wang, X. Liu, Y. Dou, and R. Li, "Multiple kernel clustering framework with improved kernels," *Discover*, vol. 1, no. 2, pp. 3–4, 2017.
[10] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Proc. NIPS*, vol. 2, 2014, pp. 1305–1313.
[11] L. Du *et al.*, "Robust Multiple kernel k-means using $\ell_{2,1}$-norm," in *Proc. 4th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 3476–3482.
[12] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple Kernel k-means clustering with matrix-induced regularization," in *Proc. 30th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 1888–1894.
[13] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel k-means with incomplete kernels," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 2259–2265.
[14] X. Liu *et al.*, "Optimal neighborhood kernel clustering with multiple kernels," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 2266–2272.
[15] X. Liu *et al.*, "Multiple kernel k-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[16] D. Zhou and C. J. C. Burges, "Spectral clustering and transductive learning with multiple views," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 1159–1166.
[17] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, Jun. 2014, pp. 2149–2155.
[18] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. CVPR*, Jun. 2015, pp. 586–594.
[19] P. Zhou, L. Du, L. Shi, H. Wang, and Y.-D. Shen, "Recovery of corrupted multiple kernels for clustering," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 4105–4111.

[20] L. Feng, L. Cai, Y. Liu, and S. Liu, "Multi-view spectral clustering via robust local subspace learning," *Soft Comput.*, vol. 21, no. 8, pp. 1937–1948, 2017.
[21] T. Li, Y. Dou, X. Liu, Y. Zhao, and Q. Lv, "Multiple kernel clustering with corrupted kernels," *Neurocomputing*, vol. 267, pp. 447–454, Dec. 2017.
[22] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Jul. 2011.
[23] X. Liu *et al.*, "Absent multiple kernel learning algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[24] X. Liu *et al.*, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[25] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 851–864, Mar. 2019.
[26] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
[27] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1265–1271, Apr. 2019.
[28] J. Han, H. Liu, and F. Nie, "A local and global discriminative framework and optimization for balanced clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.
[29] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, "Dynamic affinity graph construction for spectral clustering using multiple features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6323–6332, Dec. 2018.
[30] C. Yao, J. Han, F. Nie, F. Xiao, and X. Li, "Local regression and global information-embedded dimension reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4882–4893, Oct. 2018.
[31] Y. Zhao, K. Xu, X. Liu, E. Zhu, X. Zhu, and J. Yin, "Triangle lasso for simultaneous clustering and optimization in graph datasets," *IEEE Trans. Knowl. Data Eng.*, to be published.
[32] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2014.
[33] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, vol. 16, 2016, pp. 1704–1710.
[34] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[35] C. de Bodt, D. Mulders, M. Verleysen, and J. A. Lee, "Nonlinear dimensionality reduction with missing data using parametric multiple imputations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1166–1179, Apr. 2019.
[36] Y. Zhao, Y. Ming, X. Liu, E. Zhu, K. Zhao, and J. Yin, "Large-scale k-means clustering via variance reduction," *Neurocomputing*, vol. 307, pp. 184–194, Sep. 2018.
[37] P. Wei, Y. Ke, and C. K. Goh, "Feature analysis of marginalized stacked denoising autoenconder for unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1321–1334, May 2019.
[38] X. Li, H. Zhang, R. Zhang, Y. Liu, and F. Nie, "Generalized uncorrelated regression with adaptive graph for unsupervised feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1587–1595, May 2019.
[39] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[40] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, *Robust and Efficient Subspace Segmentation via Least Squares Regression*. Berlin, Germany: Springer, 2012.
[41] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[42] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 795–828, Mar. 2012.
[43] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, Jul. 2014.
[44] M. Jaggi and M. Sulovský, "A simple algorithm for nuclear norm regularized problems," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 471–478.
[45] S. Xiang, Y. Zhu, X. Shen, and J. Ye, "Optimal exact least squares rank minimization," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 480–488.
[46] D. Goldfarb and S. Liu, "An $O(n^3L)$ primal interior point algorithm for convex quadratic programming," *Math. Program.*, vol. 49, nos. 1–3, pp. 325–340, Nov. 1990.
[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[49] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, Jun. 2015, pp. 1–9.
[50] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Multiple kernel fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120–134, Feb. 2012.
[51] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Sihang Zhou received the bachelor's degree in information and computing science from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012, the M.S. degree in computer science from the National University of Defense Technology (NUDT), Changsha, China, in 2014, where he is currently pursuing the Ph.D. degree.

His current research interests include machine learning, pattern recognition, and medical image analysis.

Xinwang Liu received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China.

He is currently an Assistant Researcher with School of Computer Science, NUDT. He has published over 60 peer-reviewed papers, including those in highly regarded journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the International Conference on Computer Vision, the Advertising Agencies Association of India, and the International Joint Conferences on Artificial Intelligence. His current research interests include kernel learning and unsupervised feature learning.

Miaomiao Li is currently pursuing the Ph.D. degree with the National University of Defense Technology, Changsha, China.

She is also a Lecturer with the College of Changsha, Changsha. She has published several peer-reviewed papers in journals and conferences, such as the Advancement of Artificial Intelligence (AAAI), the International Joint Conferences on Artificial Intelligence (IJCAI), *Neurocomputing*. Her current research interests include kernel learning and multiview clustering.

Ms. Li serves on the Technical Program Committees of IJCAI 2017–2019.

En Zhu received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China.

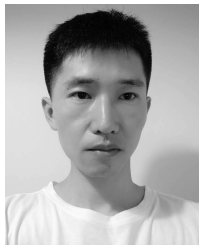
He is currently a Professor with the School of Computer Science, NUDT. He has published over 60 peer-reviewed papers in journals and conferences, including the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, the Advertising Agencies Association of India, the International Joint Conferences on Artificial Intelligence. His main research interests include pattern recognition, image processing, machine vision, and machine learning.

Dr. Zhu was a recipient of the China National Excellence Doctoral Dissertation.

Li Liu received the B.Sc. degree in communication engineering, the M.Sc. degree in photogrammetry and remote sensing, and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2003, 2005, and 2012, respectively.

She joined the faculty at NUDT in 2012, where she is currently an Associate Professor with the College of System Engineering. During her Ph.D. study, she spent more than two years as a Visiting Student with the University of Waterloo, Waterloo, ON, Canada, from 2008 to 2010. From 2015 to 2016, she spent ten months visiting the Multimedia Laboratory, The Chinese University of Hong Kong, Hong Kong. From 2016 to 2018, she was a Senior Researcher with the Machine Vision Group, University of Oulu, Oulu, Finland. Her current research interests include facial behavior analysis, texture analysis, image classification, object detection, and recognition.

Dr. Liu was the Co-Chair of seven international workshops at the Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Computer Vision, and the European Conference on Computer Vision. She is going to lecture a tutorial at CVPR 2019. She was a Guest Editor of special issues for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*. Her papers have currently over 1800 citations in Google Scholar. She currently serves as an Associate Editor of the *Visual Computer Journal*.



Changwang Zhang received the Ph.D. degree from University College London, London, U.K.

He is currently a Senior Researcher with Tencent Technology (Shenzhen) Co., Ltd., Shenzhen, China. His current research interests include machine learning and data mining.

Jianping Yin received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China.

He is currently a Distinguished Professor with the Dongguan University of Technology, Dongguan, China. His current research interests include pattern recognition and machine learning. He has published over 150 peer-reviewed papers in journals and conferences, including the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, PR, the Advertising Agencies Association of India, and the International Joint Conferences on Artificial Intelligence.

Dr. Yin was a recipient of the China National Excellence Doctoral Dissertation' Supervisor Award and the National Excellence Teacher Award. He served on the Technical Program Committees of over 30 international conferences and workshops.