Simultaneous Global and Local Graph Structure Preserving for Multiple Kernel Clustering

Zhenwen Ren, *Member, IEEE*, and Quansen Sun

***Abstract*— Multiple kernel learning (MKL) is generally recognized to perform better than single kernel learning (SKL) in handling nonlinear clustering problem, largely thanks to MKL avoids selecting and tuning predefined kernel. By integrating the self-expression learning framework, the graph-based MKL subspace clustering has recently attracted considerable attention. However, the graph structure of data in kernel space is largely ignored by previous MKL methods, which is a key concept of affinity graph construction for spectral clustering purposes. In order to address this problem, a novel MKL method is proposed in this article, namely, structure-preserving multiple kernel clustering (SPMKC). Specifically, SPMKC proposes a new kernel affine weight strategy to learn an optimal consensus kernel from a predefined kernel pool, which can assign a suitable weight for each base kernel automatically. Furthermore, SPMKC proposes a kernel group self-expressiveness term and a kernel adaptive local structure learning term to preserve the global and local structure of the input data in kernel space, respectively, rather than the original space. In addition, an efficient algorithm is proposed to solve the resulting unified objective function, which iteratively updates the consensus kernel and the affinity graph so that collaboratively promoting each of them to reach the optimum condition. Experiments on both image and text clustering demonstrate that SPMKC outperforms the state-of-the-art MKL clustering methods in terms of clustering performance and computational cost.**

***Index Terms*— Affinity graph learning, kernel self-expression, multiple kernel learning (MKL), structure preserving, subspace clustering.**

I. Introduction

HIGH-DIMENSIONAL data clustering is a widespread fundamental problem in many fields [1], such as data mining communities, computer vision, machine learning, bioinformatics, and more. Often such high-dimensional data can often be well described by a low-dimensional subspace corresponding to a class or category. Accordingly, subspace clustering denotes the problem of clustering data points drawn from a union of low-dimensional linear subspaces into their respective subspaces, which has been widely used in many applications [2]–[6], e.g., image representation, motion segmentation, community clustering in social networks, hybrid system identification in the control system, and genes expression profiles clustering in bioinformatics. Over the past few years, the subspace clustering has received a lot of attention, and many methods have been proposed [4], [7]–[9], including matrix factorization-based methods, algebraic-based methods, statistical-based methods, iterative-based methods, and spectral clustering-based methods.

Among the above-mentioned methods, spectral clustering-based ones have become extremely popular [8]. Owing to their strong capability to handle arbitrarily shaped clusters and their well-defined mathematical principles. Such methods typically contain two sequential steps [3], [9]: 1) constructing an affinity matrix (also known as affinity graph) from the input data, where each entry reflects the similarity relationship between two data points and 2) applying spectral clustering to this learned affinity graph to obtain the clustering results. Arguably, the first step is the critical one, as the success of the spectral clustering methods is largely dependent on constructing an informative affinity graph. Although these spectral clustering-based methods have achieved significant success, there are two great challenges: 1) how to handle nonlinear data when the data is not strictly collected from linearly independent subspaces and 2) how to learn an optimal affinity graph from the nonlinear data for clustering purpose. Ideally, if there are n samples from c clusters, the optimal affinity graph is a $n \times n$ square matrix that has two characteristics [3].

1) It obeys the block diagonal property with exact c connected blocks, where each block corresponds one-to-one with each subject of the data.
2) The between-cluster affinities are all zeros, but the inner-cluster affinities are not-zeros, where the affinity value reflects the similarity between two samples. Visually, each block has a dense and distinct appearance.

In order to tackle the challenge of handling nonlinear data, the single kernel learning (SKL) methods have been widely researched [10]–[12], which extend linear subspace clustering to nonlinear counterparts by kernel tricks [12]. Although improved performance has been reported in a wide variety

Manuscript received August 8, 2019; revised February 9, 2020; accepted April 25, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61673220 and Grant 61906091, in part by the National Defense Pre-Research Foundation of China under Grant JCKY2017209B010 and Grant JCKY2018209B001, and in part by the Natural Science Foundation of Jiangsu Province, China, under Grant BK20190440. *(Corresponding author: Quansen Sun.)*

Zhenwen Ren is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the School of National Defence Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China (e-mail: rzw@njust.edu.cn).

Quansen Sun is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: sunquansen@njust.edu.cn).

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

Digital Object Identifier 10.1109/TNNLS.2020.2991366

of tasks, the SKL methods require the user to select and tune a predefined kernel since the performance of the SKL methods are crucially determined by choice of kernel function [13]. Obviously, this is not user-friendly since the most suitable kernel and the associated parameters for a specific data set are usually challenging to decide. In order to boost the quality of affinity graph, many scholars have proposed different regularization terms and constraints terms, such as low-rank constraint [14], sparse constraint [15], continuous label learning [16], adaptive weighted representation [17], block-diagonal representation [3], structure preserving [18], and more. Among them, structure-preserving technology is widely used for learning a better affinity graph. We refer the readers to [9], [19], and [20] for a comprehensive review of the literature of graph structure learning. Naturally, joint SKL and structure learning technology, that's a good choice. However, this joint model faces two limitations: 1) how to define a suitable predefined single kernel and tune its kernel parameters and 2) how to preserve the structural information of the data in kernel space rather than the original space.

Regarding the limitations mentioned earlier, we propose a novel method named structure-preserving multiple kernel clustering (SPMKC) in this article. First, instead of using a predefined kernel, we incorporate multiple kernel learning (MKL) [21], [22] into our model to avoid selecting and tuning the predefined kernel. The MKL is a powerful model that has great potential to integrate complementary information between multiple kernels by constructing a consensual kernel from these basis kernels [23]. Accordingly, the MKL is normally more usable and performs better than that of a single kernel. Then, to preserve the global structural information of the data in kernel space, we introduce a kernel group self-expressiveness (KGS) term with the fixed connected components to excavate the global structure of the data and enhance the grouping effect of the learned affinity graph. Next, we propose a local structure preserving regularization term that can preserve the local structural information of the data in kernel space. Lastly, we elegantly integrate the mentioned terms in a unified objective function. With this unified objective function, we can automatically obtain the optimal consensus kernel and affinity graph. As a consequence, this affinity graph is fed into spectral clustering to generate final clustering results. Fig. 1 gives the illustration of our method. Here, we list the main contributions of this article as follows.

1) To handle well the input data with significant nonlinearity structure for subspace clustering tasks, we propose a novel MKL method, called SPMKC, in which, a kernel weight strategy is proposed, which can automatically weight each base kernel to find an optimal consensus kernel. In other words, SPMKC prefers to assign larger weights to important kernels and smaller weights to unimportant kernels; that is, it has a higher capability to solve the challenging problem of how to define a suitable kernel and tune kernel parameters.
2) To capture the latent global structure of the data in the Hilbert kernel space, we simultaneously integrate the kernel self-expressiveness framework with an exact rank constraint and a Frobenius norm regularization term. By doing so, the learned affinity graph has the desired number of connected components and a good grouping effect. Consequently, these properties can explore the underlying subspace of the unlabeled data so that the data points can be naturally partitioned into their respective clusters.
3) To preserve the local graph structure of the data in the Hilbert kernel space, we propose a local structure preserving term, which ensures to learn a higher-quality affinity graph for accurate clustering purposes. Furthermore, the proposed term can explain the negative term of the basic kernel self-expressiveness model [i.e., (2)] well. As we know, this is the first attempt to propose the local structure preserving in kernel space.
4) By comparing with several state-of-the-art MKL clustering methods on 11 widely used benchmark and synthetic nonlinear data sets, SPMKC substantially improves the clustering performance of the clustering performance and computational cost, which demonstrates the superiority of the proposed method.

TABLE I

NOTATIONS AND ABBREVIATIONS USED

Notation (Abbr.)	Definition
$\boldsymbol{X} \in \mathbb{R}^{m\times n}$	the data matrix with m features and n samples
x^i or $\boldsymbol{X}^i$	the i-th row of $\boldsymbol{X}$
x_j or $\boldsymbol{X}_j$	the j-th column of $\boldsymbol{X}$ or the transposition of x^j
x_{ij} or $\boldsymbol{X}_{ij}$	the (i, j)-th entry of $\boldsymbol{X}$
$\mathbf{1}$	the all-one column vector
$\boldsymbol{I}$	the identity matrix
$\boldsymbol{Z} \in \mathbb{R}^{n\times n}$	the affinity/similarity matrix
$\mathrm{Tr}(\cdot)$	the trace operator of a matrix
$\mathrm{diag}(\cdot)$	the diagonal elements of a matrix
$\odot$	the Hadamard product
$\|\cdot\|_*$	the nuclear norm
$\|\cdot\|_F$	the Frobenius norm
$\|\cdot\|_1$	the l_1 norm
$\boldsymbol{A} \geq 0$	the positive semi-definite $\boldsymbol{A}$

The rest of this article is organized as follows. Section II gives a brief review on related works. Section III is dedicated to present the proposed method for MKL subspace clustering. Section IV shows the experimental settings, results, and discussions. Section V presents the conclusions and further works.

II. Related Works

For convenience, we denote scalars, vectors, and matrices, respectively, as normal italic letters, boldface lowercase letters, and boldface uppercase letters. Some important notations and abbreviations used throughout this article are summarized in Table I.

A. Self-Expressiveness Graph Learning Framework

As we know, one sample from one subspace can be represented as a linear combination of other samples in the identical subspace, known as self-expressiveness property [3], [14].

Accordingly, the graph-based clustering methods can rely on the self-expressiveness to build a reliable affinity graph. The mathematical definition is as follows:

$$\min_{\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}\|_F^2+\alpha\mathcal{R}(\boldsymbol{Z}) \quad \text{s.t. } \boldsymbol{Z}\geqslant 0 \tag{1}$$

where $\alpha > 0$ is a tradeoff parameter, and $\mathcal{R}(\boldsymbol{Z})$ indicates a certain regularization term of $\boldsymbol{Z}$. The major difference among existing methods depends on the choice of $\mathcal{R}(\boldsymbol{Z})$ [3], [24]–[26], such as $\|\boldsymbol{Z}\|_*$, $\|\boldsymbol{Z}\|_1$, $\|\boldsymbol{Z}\|_F^2$, $\|\boldsymbol{Z}\|_*+\|\boldsymbol{Z}\|_1$, $\|\boldsymbol{Q}\odot\boldsymbol{Z}\|_F^2$ (where $\boldsymbol{Q}$ is a matrix with special structure), and more. After obtaining $\boldsymbol{Z}$, the balanced affinity graph is usually constructed by $\boldsymbol{Z}=(\boldsymbol{Z}^T+\boldsymbol{Z})/2$.

However, problem (1) cannot efficiently handle nonlinear data. To address this issue, the nonlinear data can be mapped into a high-dimensional reproducing kernel Hilbert space (RKHS) by kernel tricks, where a linear pattern analysis can be accomplished [27]. Hence, to learn an affinity graph in RKHS, problem (1) can be upgraded to

$$\begin{aligned}
&\min_{\boldsymbol{Z}} \frac{1}{2}\|\phi(\boldsymbol{X})-\phi(\boldsymbol{X})\boldsymbol{Z}\|_F^2+\alpha\mathcal{R}(\boldsymbol{Z}) \quad \text{s.t. } \boldsymbol{Z}\geqslant 0\\
&\quad=\min_{\boldsymbol{Z}} \frac{1}{2}\operatorname{Tr}(\phi(\boldsymbol{X})^T\phi(\boldsymbol{X})-2\phi(\boldsymbol{X})^T\phi(\boldsymbol{X})\boldsymbol{Z}\\
&\qquad+\boldsymbol{Z}^T\phi(\boldsymbol{X})^T\phi(\boldsymbol{X})\boldsymbol{Z})+\alpha\mathcal{R}(\boldsymbol{Z}) \quad \text{s.t. } \boldsymbol{Z}\geqslant 0\\
&\quad=\min_{\boldsymbol{Z}} \frac{1}{2}\operatorname{Tr}(\boldsymbol{K}-2\boldsymbol{K}\boldsymbol{Z}+\boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z})+\alpha\mathcal{R}(\boldsymbol{Z}) \quad \text{s.t. } \boldsymbol{Z}\geqslant 0
\end{aligned} \tag{2}$$

where $\phi(\cdot)$ is a kernel mapping and $\boldsymbol{K}\succeq 0$ is the kernel Gram matrix corresponding to $\phi(\cdot)$.

B. Adaptive Graph Structure Learning

The significance of preserving local manifold structure has been well recognized in the recent development of graph-based machine learning methods [18], [28]–[31].

For each point x_i, all the other points $\{\boldsymbol{x}_j\}_{j=1}^n$ are considered as the neighborhood of x_i with probability z_{ij}. Intuitively, the data points being close should have similar representation coefficients, i.e., a smaller distance $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2$ should be assigned a larger probability z_{ij}, and vice versa. As such, the intrinsic local structure of data in original space is preserved [28], [31], [32]. In light of this observation, the representation similarity (or probability) graph $\boldsymbol{Z}\in\mathbb{R}^{n\times n}$ can be calculated by solving the following problem:

$$\min_{z_i^T\mathbf{1}=1,0\leq z_{ij}\leq 1} \sum_{i,j}\|x_i-x_j\|_2^2 z_{ij} \tag{3}$$

where z_i is the transposition of the ith row of graph $\boldsymbol{Z}$.

To urge $\boldsymbol{Z}$ to hold desired structure or avoid the trivial solution, some additional regularization terms are usually appended into (3), such as $\|\boldsymbol{Z}\|_1$ [33] and $\|\boldsymbol{Z}\|_F^2$ [18].

However, how to preserve the local graph structure of the input data in kernel space is an important problem worthy of study.

C. Related MKL Clustering Methods

Based on the graph self-expressiveness framework (2), MKL-based spectral clustering has drawn more and more attention in recent years, and many variants have been proposed [13], [22], [34]–[37]. For instance, multiple kernel k-means (MKKM) [22] extends k-means into a multiple kernel setting. An extended version of MKKM, robust MKKM (RMKKM) [35], has been proposed, which simultaneously finds the best clustering label, the optimal combination of multiple kernels, and the cluster membership. Meanwhile, RMKKM overcomes the effect of noise and outliers in MKKM via an $\ell_{2,1}$ norm. Due to the similarity between the kernel matrix and the affinity matrix, affinity aggregation for spectral clustering (AASC) [36] replaces the single affinity matrix in spectral clustering with multiple matrices, which can be considered as an MKL version of spectral clustering. By considering the alignment between the local structure and global structure of the data has the same optimal similarity matrix, a global and local structure alignment framework for MKC has been proposed [38]. Since the kernel matrix can exploit the similarity characteristics of the kernel matrix and seek an optimal consensus kernel from a neighborhood of candidate kernels, self-weighted MKL (SMKL) [13] has been proposed, which is a novel MKL framework for subspace clustering and semisupervised classification. Spectral clustering with multiple kernels (SCMK) [16] learns an optimal consensus kernel by using the same way adopted by RMKKM, i.e., it uses the fact that the optimal consensus kernel is a linear combination of predefined candidate base kernels. By considering the intrinsic neighborhood structure among base kernels, a neighbor-kernel-based multiple kernel clustering method has been proposed [39]. By considering the low-rank and spare property of the samples, low-rank kernel learning graph-based clustering (LKGr) [37] and sparse kernel learning graph-based clustering (LKGs) [37] have been proposed to impose a low-rank or a sparse constraint on kernel matrix, respectively, while learning an optimal consensus kernel.

However, previous MKL methods mainly focus on how to define a kernel weight strategy, but ignored the global and local structural characteristics of the input data, especially in kernel space.

III. Proposed Method

As introduced in Section II, MKL graph-based clustering mainly focus on learn an optimal affinity graph and consensus kernel and has many disadvantages. In this section, we present a novel MKL method, termed SPMKC.

A. Problem Formulation

Given a data matrix $\boldsymbol{X}\in\mathbb{R}^{m\times n}$ with m high-dimensional features and n samples from c clusters/classes. Let $\phi:\mathbb{R}^d\rightarrow\mathcal{H}$ be a nonlinear feature mapping that maps $\boldsymbol{x}$ from the original space to the RKHS $\mathcal{H}$, and $\boldsymbol{K}\in\mathbb{R}^{n\times n}\succeq 0$ be a kernel Gram matrix whose (i,j)th entry is computed as

$$\boldsymbol{K}_{ij}=(\phi(\boldsymbol{X})^T\phi(\boldsymbol{X}))_{ij}=\phi(x_i)^T\phi(x_j)=ker(x_i,x_j) \tag{4}$$

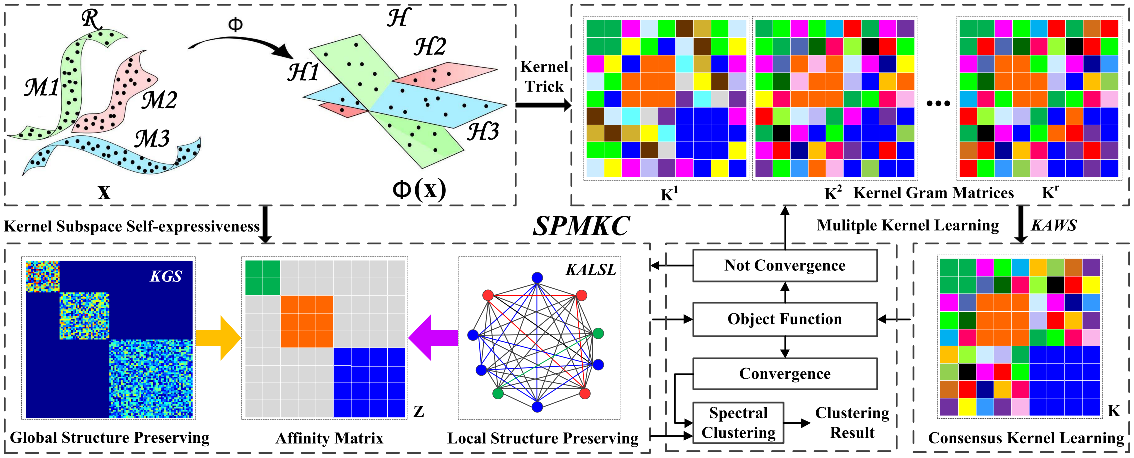


Fig. 1. Block diagram of the proposed SPMKC method.

where $ker : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a kernel function and $\phi(\boldsymbol{X}) = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)]$. Deservedly, we assume that $\phi(x_i)_{i=1}^n$ reside in multiple linear subspaces in $\mathcal{H}$.

As mentioned earlier, problem (2) can learn an affinity graph $\boldsymbol{Z}$ in RKHS; however, it cannot learn an affinity graph with exact c diagonal connected components (blocks). Fortunately, **Theorem 1** can tackle this problem.

Theorem 1 [3]: For any square $\boldsymbol{Z} \in \mathbb{R}^{n\times n} \geq 0$, the multiplicity c of the eigenvalue 0 of the corresponding Laplacian matrix $\boldsymbol{L_Z}$ equals the number of connected components in $\boldsymbol{Z}$.

Motivated by such a theorem, it can be proved that if $\text{rank}(\boldsymbol{L_Z}) = n - c$, the affinity graph $\boldsymbol{Z}$ will contain exact c connected components, where $\boldsymbol{L_Z} = \boldsymbol{D} - (\boldsymbol{Z} + \boldsymbol{Z}^T)/2$, and the degree matrix $\boldsymbol{D}$ is a diagonal matrix, whose ith diagonal entry is computed by $d_{ii} = \sum_j [(z_{ij} + z_{ji})/2]$. According to rank theory and Fan's theorem [40], we have

$$\text{rank}(\boldsymbol{L_Z}) = n - c \Rightarrow \min_{\boldsymbol{P}\in\mathbb{R}^{n\times c}, \boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I}} \text{Tr}(\boldsymbol{P}^T\boldsymbol{L_Z}\boldsymbol{P}) \tag{5}$$

where $\boldsymbol{P}^T = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ is the cluster indicator matrix. By doing so, the learned $\boldsymbol{Z}$ is block diagonal with proper permutation. Therefore, problem (2) can be transformed to

$$\min_{\boldsymbol{Z}} \frac{1}{2}\text{Tr}(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}) + \beta\,\text{Tr}(\boldsymbol{P}^T\boldsymbol{L_Z}\boldsymbol{P})$$
$$+\alpha\mathcal{R}(\boldsymbol{Z}) \text{ s.t. } \boldsymbol{Z} \geqslant 0, \quad \text{diag}(\boldsymbol{Z}) = \boldsymbol{0}, \quad \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I} \tag{6}$$

where $\beta > 0$ is a tradeoff parameter, and $\text{diag}(\boldsymbol{Z}) = \boldsymbol{0}$ indicates that the data point cannot be represented by the data point itself.

Inspired by least squares regression (LSR) [25], [41], the Frobenius norm-based regularization term, $\|\boldsymbol{Z}\|_F^2$, can encourage affinity graph $\boldsymbol{Z}$ to preserve grouping effect and ensure connectedness property. Moreover, $\|\boldsymbol{Z}\|_F^2$ is similar to the quadratic penalty term of augmented Lagrangian methods, which can promote the convergence of the objective function [42]. Consequently, from the perspective of the global structure, the learned affinity graph has good block diagonal property and grouping effect, which can groups the highly correlated data together possibly and lead to correct clustering [3]. Therefore, we let $\mathcal{R}(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_F^2$, and call (6) KGS.

Although many clustering methods [30], [40] and clustering guided feature selection methods [18], [29] also use (5) to exploit the underlying structure relationship between samples. Note that if parameter β is large enough, the numbers of connected components of the learned graph $\boldsymbol{Z}$ will be exactly c, such that we can obtain a stable output when performing clustering. Obviously, the fixed β can hardly satisfy such a requirement. Instead, we will propose a technology to self-tune β until the constraint condition is exactly satisfied.

For local structure preserving, a model, adaptive local structure learning (ALSL), is widely used in linear subspace for unsupervised feature learning [18], [29], [43], dimensionality reduction [44], and clustering [5], [32]. The model is shown in (3). To avoid the trivial solution, we append an additional regularization term $\sum_{i,j=1}^n z_{ij}^2$. Accordingly, we have

$$\min_{\boldsymbol{Z}} \sum_{i=1}^n \sum_{j=1}^n (\|x_i - x_j\|^2 z_{ij} + \gamma z_{ij}^2) \quad \text{s.t. } z_i^T\boldsymbol{1} = 1, 0 \leq z_{ij} \tag{7}$$

where γ is a tradeoff parameter, and the constraints $z_i^T\boldsymbol{1} = 1$ and $0 \leq z_{ij}$ are to guarantee the probability property of z_i.

However, (7) is designed to linear space rather than kernel space. To preserve the local manifold structure in kernel space, one may think of using $\|\phi(x_i) - \phi(x_j)\|_2^2$ instead of $\|x_i - x_j\|_2^2$ intuitively, where ϕ is a mapping from the input space to the kernel space; nevertheless, it is difficult to solve that. Based on kernel trick, we propose a new model, named kernel ALSL (KALSL), which uses $-ker(x_i, x_j)$ to measure the similarity between x_i and x_j in kernel space, that is

$$\min_{\boldsymbol{Z}} \sum_{i=1}^n \sum_{j=1}^n (-ker(x_i, x_j) z_{ij} + \gamma z_{ij}^2) \quad \text{s.t. } z_i^T\boldsymbol{1} = 1, \ 0 \leq z_{ij}. \tag{8}$$

Mathematically, problem (8) can be transformed into

$$\min_{\boldsymbol{Z}} \ -\text{Tr}(\boldsymbol{K}^T\boldsymbol{Z}) + \gamma\|\boldsymbol{Z}\|_F^2 = -\text{Tr}(\boldsymbol{K}\boldsymbol{Z}) + \gamma\|\boldsymbol{Z}\|_F^2$$
$$\text{s.t. } 0 \leq \boldsymbol{Z}, \ \boldsymbol{Z}\boldsymbol{1} = \boldsymbol{1}. \tag{9}$$

As can be seen, we already have $-\text{Tr}(\boldsymbol{K}\boldsymbol{Z})$ term in problem (6). By comparison, KALSL can explain the negative term in

(6) well, i.e., it can preserve the local manifold geometric structure in kernel space such that the affinity graph learning will benefit. Evidently, the effect of KALSL term is best to be tuned, rather than a fixed value.

It is well-known that the performance of SKL is crucially determined by the choice of kernel function and its parameters. Moreover, exhaustive search of the most suitable kernel from a prespecified kernel pool is very computationally expensive and hard to run in real-time. But MKL is an efficient way for automatic kernel selection and parameter tuning according to different data sets. Inspired by [30], [31], and [45], the MKL model is given by

$$\min_{\boldsymbol{K}} \sum_{k=1}^{r} w_k \|\boldsymbol{K}^k - \boldsymbol{K}\|_F^2 \tag{10}$$

where r is the number of base kernels, $\boldsymbol{K}$ is the anticipated consensus kernel, $\boldsymbol{K}^k$ is kth base kernel in the predefined kernel pool, and $\boldsymbol{w} = [w_1, w_2, \ldots, w_r]^T$ is the weight vector.

It should be note that the weight value w_k is used to control the contribution of ith kernel $\boldsymbol{K}^k$ to generate $\boldsymbol{K}$. Here, we simply require $\sum_k w_k = 1$ to make a balance between different kernels. Then, based on heat kernel distance, we proposed a new kernel weight strategy by assuming that the important kernels will have a big weight values while the unimportant kernels have low or near-zero weight values. We call kernel affine weight strategy (KAWS), that is

$$w_k = \frac{\exp(-\delta e_k/\overline{e})}{\sum_{k=1}^{r} \exp(-\delta e_k/\overline{e})} \tag{11}$$

where δ is a scalar parameter, $e_k = \|\boldsymbol{K}^k - \boldsymbol{K}\|_F^2$, and $\overline{e}$ indicates the mean value of $[e_1, e_2, \ldots, e_r]$ (i.e., $\overline{e} = 1/r \sum_{k=1}^{r} e_k$). As a result, KAWS has four advantages: 1) the weight $\sum_{k=1}^{r} w_k = 1$ and $\boldsymbol{w} > 0$, i.e., it implies an affine constrain to avoid scale change. 2) The value of heat kernel varies in the range $w_k \in (0, 1)$; for significant ones, the values are close to 1, while for negligible kernels, the values are close to 0. Compare with the Euclidean distance-based weight strategy (EDWS) used in [13], which uses $\|\boldsymbol{K}^k - \boldsymbol{K}\|_F \in (0, +\infty)$ to compute the weight of kernel $\boldsymbol{K}^k$, which may result in partiality toward one kernel due to noise and outliers. 3) Different from [16], which enforces the optimal consensus kernel being a linear combination of candidate base kernels (i.e., $\boldsymbol{K} = \sum_{k=1}^{r} \boldsymbol{K}^k$), KAWS is a nonlinear weight strategy. 4) KAWS is a self-weighted strategy, which can automatically update the weights according to the current state of the consensus kernel.

Hereto, by elegantly integrating the KGS, KALSL, and MKL into a uniform model, and performing slight algebraic transformation, the objective function can be finally formulized as follows:

$$\begin{aligned}\min_{\boldsymbol{Z},\boldsymbol{K},\boldsymbol{P}} \; & \frac{1}{2}\operatorname{Tr}(\boldsymbol{K} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}) - \lambda_1 \operatorname{Tr}(\boldsymbol{K}\boldsymbol{Z}) \\ & + \lambda_2 \operatorname{Tr}(\boldsymbol{P}^T\boldsymbol{L}_{\boldsymbol{Z}}\boldsymbol{P}) + \lambda_3 \sum_{k=1}^{r} w_k \|\boldsymbol{K}^k - \boldsymbol{K}\|_F^2 + \lambda_4 \|\boldsymbol{Z}\|_F^2 \\ \text{s.t. } & 0 \le \boldsymbol{Z},\ \boldsymbol{Z}\boldsymbol{1} = \boldsymbol{1},\ \operatorname{diag}(\boldsymbol{Z}) = \boldsymbol{0},\ \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}\end{aligned} \tag{12}$$

where $\lambda_{i\in[1,2,3,4]} > 0$ are used to balance the data local structure, graph connectivity, consensus kernel construction, and data global structure, respectively.

We denote the model (12) with KAWS as SPMKC for ease of notation. In addition, to see the effect of the proposed KAWS, we need to examine the model (12) with EDWS (i.e., $w_k = 1/\|\boldsymbol{K}^k - \boldsymbol{K}\|_F$) that is called SPMKC-E.

B. Optimization

Although (12) is not jointly convex with respect to $\boldsymbol{Z}$, $\boldsymbol{K}$ and $\boldsymbol{P}$, it is convex with respect to each of them while holding the other fixed. Therefore, (12) can be effectively solved with an alternating optimization algorithm [46].

1) Update $\boldsymbol{Z}$ as Given $\boldsymbol{K}$ and $\boldsymbol{P}$: The subproblem for updating $\boldsymbol{Z}$ becomes as follows:

$$\begin{aligned}\min_{\boldsymbol{Z}} \; & \frac{1}{2}\operatorname{Tr}(\boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}) - \lambda_1 \operatorname{Tr}(\boldsymbol{K}\boldsymbol{Z}) + \lambda_2 \operatorname{Tr}(\boldsymbol{P}^T\boldsymbol{L}_{\boldsymbol{Z}}\boldsymbol{P}) \\ & + \lambda_4\|\boldsymbol{Z}\|_F^2 \quad \text{s.t. } 0 \le \boldsymbol{Z}, \boldsymbol{Z}\boldsymbol{1} = \boldsymbol{1}, \operatorname{diag}(\boldsymbol{Z}) = \boldsymbol{0}\end{aligned} \tag{13}$$

where $\boldsymbol{L}_{\boldsymbol{Z}} = (\boldsymbol{Z} + \boldsymbol{Z}^T)/2$.

For the third term $\operatorname{Tr}(\boldsymbol{P}^T(\boldsymbol{Z} + \boldsymbol{Z}^T/2)\boldsymbol{P})$, we can rewrite it as a concise appearance via **Theorem 2.**

Theorem 2: Denote the Laplacian matrix and the degree matrix of the symmetric affinity matrix $\boldsymbol{Z} \in \mathbb{R}^{n\times n}$ as $\boldsymbol{L}_{\boldsymbol{Z}} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$, respectively, the following equality is held:

$$\operatorname{Tr}(\boldsymbol{P}^T\boldsymbol{L}_{\boldsymbol{Z}}\boldsymbol{P}) = \frac{1}{2}\operatorname{Tr}(\boldsymbol{Z}\boldsymbol{Q}) \tag{14}$$

where $\boldsymbol{Q} \in R^{n\times n}$ is a symmetric matrix, whose (i, j)th element is computed by $q_{ij} = \|p^i - p^j\|_2^2$, and p^i and p^j are the ith and jth rows of matrix $\boldsymbol{P}$, respectively.

Proof: Mathematically, we have the following derivation:

$$\begin{aligned}&\Rightarrow \operatorname{Tr}(\boldsymbol{P}^T\boldsymbol{L}_{\boldsymbol{Z}}\boldsymbol{P}) \\ &= \operatorname{Tr}(\boldsymbol{P}^T\boldsymbol{D}\boldsymbol{P}) - \operatorname{Tr}(\boldsymbol{P}^T\boldsymbol{Z}\boldsymbol{P}) \\ &= \frac{1}{2}\left(\sum_{i=1}^{n} d_{ii}p^i(p^i)^T - 2\sum_{i,j=1}^{n} z_{ij}p^i(p^j)^T + \sum_{j=1}^{n} d_{jj}p^j(p^j)^T\right) \\ &= \frac{1}{2}\sum_{i,j=1}^{n} z_{ij}\|p^i - p^j\|_2^2 = \frac{1}{2}\boldsymbol{1}^T(\boldsymbol{Q}\odot\boldsymbol{Z})\boldsymbol{1} = \frac{1}{2}\operatorname{Tr}(\boldsymbol{Z}\boldsymbol{Q}).\end{aligned}$$

Thus, (14) is achieved. The proof is completed. ■

For computational simplicity and efficiency, we consider a two-step fast approximation [47], [48] by defining an auxiliary problem of (13), and then compute its latent solution $\widetilde{\boldsymbol{Z}}$.

In step one, we solve the following problem:

$$\begin{aligned}\widetilde{\boldsymbol{Z}} = \arg\min_{\boldsymbol{Z}} \; & \frac{1}{2}\operatorname{Tr}(\boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}) - \lambda_1 \operatorname{Tr}(\boldsymbol{K}\boldsymbol{Z}) \\ & + \frac{\lambda_2}{2}\operatorname{Tr}(\boldsymbol{Z}\boldsymbol{Q}) + \lambda_4\|\boldsymbol{Z}\|_F^2.\end{aligned} \tag{15}$$

Obviously, by taking the derivative of (15) w.r.t. $\boldsymbol{Z}$ to zero, the closed-form solution of (15) is

$$\widetilde{\boldsymbol{Z}} = (\boldsymbol{K} + 2\lambda_4\boldsymbol{I})^{-1}\left(\lambda_1\boldsymbol{K} - \frac{\lambda_2}{2}\boldsymbol{Q}^T\right). \tag{16}$$

In step two, to satisfy the involved constraints of (13), we project the latent solution $\widetilde{\boldsymbol{Z}}$ to a capped simplex constrained space[1]. Thus, we can obtain the approximate solution of $\boldsymbol{Z}$ via the following minimization problem:

$$\min_{\boldsymbol{Z}\geq 0, \mathrm{diag}(\boldsymbol{Z})=\boldsymbol{0}, \boldsymbol{Z}\boldsymbol{1}=\boldsymbol{1}} \|\boldsymbol{Z}-\widetilde{\boldsymbol{Z}}\|_F^2 \tag{17}$$

which can be expanded to the following minimization problem:

$$\min_{\boldsymbol{Z}\geq 0, \mathrm{diag}(\boldsymbol{Z})=\boldsymbol{0}, \boldsymbol{Z}\boldsymbol{1}=\boldsymbol{1}} \sum_{i=1}^{n}\sum_{j=1}^{n} \|z_{ij}-\widetilde{z}_{ij}\|_2^2. \tag{18}$$

For $\forall j$, row z_j (with transposition) can be calculated by

$$\min_{z_j\geq 0, z_{jj}=0, z_j^T\boldsymbol{1}=1} \sum_{j=1}^{n} \|z_j-\widetilde{z}_j\|_2^2. \tag{19}$$

To compute z_j, analogous to [28], we denote the Lagrangian function of problem (19) as

$$\mathcal{L}(z_j,\alpha_j,\beta_j)=\frac{1}{2}\|z_j-\widetilde{z}_j\|_2^2-\alpha_j(z_j^T\boldsymbol{1}-1)-\beta_j^T z_j \tag{20}$$

where α_j and $\beta_j \geq 0$ are the two Lagrangian multipliers.

Taking the derivative of (20) with respect to z_j and setting the derivative to zero, we have

$$z_j-\widetilde{z}_j-\alpha_j\boldsymbol{1}-\beta_j=0. \tag{21}$$

Note that $\beta_j \odot z_j = 0$ according to the KKT condition [40], we have

$$\rightarrow z_j=\max(\bar{z}_j+\alpha_j\boldsymbol{1},0) \tag{22}$$

where $\bar{z}_j=\widetilde{z}_j$ subject to $\widetilde{z}_{jj}=0$.

According to the affine constraint $z_j^T\boldsymbol{1}=1$, we obtain

$$\sum_{i=1}^{n}(\alpha_j+\bar{z}_{ij})=1 \rightarrow \alpha_j=\left(1+\bar{z}_j^T\boldsymbol{1}\right)/(n-1). \tag{23}$$

After obtaining α_j, z_j can be constructed by (22). To eliminate unbalance, we then construct the affinity matrix by $\boldsymbol{Z}=1/2(|\boldsymbol{Z}|+|\boldsymbol{Z}|^T)$. Thus, the optimal solution $\boldsymbol{Z}^*$ of (13) is obtained.

2) Update $\boldsymbol{P}$ as Given $\boldsymbol{Z}$ and $\boldsymbol{K}$: The subproblem for updating $\boldsymbol{P}$ becomes as follows:

$$\boldsymbol{P}^*=\arg\min_{\boldsymbol{P}} \lambda_2 \,\mathrm{Tr}(\boldsymbol{P}^T\boldsymbol{L}_{\boldsymbol{Z}}\boldsymbol{P}) \quad \text{s.t. } \boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I}. \tag{24}$$

The optimal solution of such a problem (24) is the eigenvectors corresponding to the smallest c eigenvalues of the Laplacian matrix $\boldsymbol{L}_{\boldsymbol{Z}}$.

3) Update $\boldsymbol{K}$ as Given $\boldsymbol{Z}$ and $\boldsymbol{P}$: The subproblem for updating $\boldsymbol{K}$ becomes as follows:

$$\min_{\boldsymbol{K}} \frac{1}{2}\mathrm{Tr}(\boldsymbol{K}+\boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z})-\lambda_1\mathrm{Tr}(\boldsymbol{K}\boldsymbol{Z}) + \lambda_3\sum_{k=1}^{r} w_k\|\boldsymbol{K}^k-\boldsymbol{K}\|_F^2. \tag{25}$$

Then, the closed-form solution of (25) is as follows:

$$\rightarrow \boldsymbol{K}^*=\frac{-\boldsymbol{I}-\boldsymbol{Z}\boldsymbol{Z}^T+2\lambda_1\boldsymbol{Z}^T+4\lambda_3\sum_{k=1}^{r}w_k\boldsymbol{K}^k}{4\lambda_3\sum_{k=1}^{r}w_k}. \tag{26}$$

Then, the consensus kernel is remodeled by $\boldsymbol{K}=\max(\boldsymbol{K},0)$ and $\boldsymbol{K}=(\boldsymbol{K}^T+\boldsymbol{K})/2$. After obtaining $\boldsymbol{K}$, the weight vector $\boldsymbol{w}$ is computed according to KAWS [i.e., (11)].

4) Update λ_2: The parameter λ_2 is used to balance the number of the connected components of the learned affinity graph $\boldsymbol{Z}$. In order to force $\boldsymbol{Z}$ has exact c connected blocks, in each iteration, we automatically double it or halve it when the connected components of $\boldsymbol{Z}$ is smaller or greater than c, respectively, that is,

$$\lambda_2=\begin{cases}2\lambda_2, & g<c\\ \dfrac{\lambda_2}{2}, & g>c\end{cases} \tag{27}$$

where g is the number of the connected components of $\boldsymbol{Z}$ in each iterator. As a result, a large enough λ_2 is achieved.

The procedure of solving the problem (12) terminates when $\mathrm{rank}(\boldsymbol{L}_{\boldsymbol{Z}})=n-c$ or the maximal number of iterations is reached. The pseudocode is shown in Algorithm 1, and the implementation source code will be released in our github page: https://github.com/renzhenwen/demo_SPMKC.

Algorithm 1 Solve (12) for Multiple Kernel Clustering

```
Input: $r$ base kernels $\{\boldsymbol{K}^k\}_{k=1}^r$, and parameters $\lambda_1$ and $\lambda_3$.
  Initialize: $(\boldsymbol{Z})^1=\boldsymbol{I}$, $(\boldsymbol{K})^1=\frac{1}{r}\sum_{k=1}^{r}\boldsymbol{K}^k$, $\{(w_k)^1\}_{k=1}^r=\frac{1}{r}$,
  $\lambda_2=\lambda_4=1$, $\delta=10$, $t=1$, and $mit=1e3$.
while $(\mathrm{rank}(\boldsymbol{L}_{\boldsymbol{Z}})!=n-c)$ and $(++t<mit)$ do
    Update $(\boldsymbol{P})^{t+1}$ via (24), compute $\lambda_2$ via (27).
    Update $(\boldsymbol{Z})^{t+1}$ via (13).
    Update $(\boldsymbol{K})^{t+1}$ via (26), compute $\boldsymbol{w}$ via (11).
end while
Obtain the optimal $\boldsymbol{Z}^*$, $\boldsymbol{P}^*$, and $\boldsymbol{K}^*$.
Perform spectral clustering.
Output: The clustering results: ACC, NMI, and purity.
```

C. Complexity Analysis

The main computational cost of Algorithm 1 mainly depends on updating $\boldsymbol{Z}$, $\boldsymbol{P}$, $\boldsymbol{K}$, and $\boldsymbol{w}$. For updating $\boldsymbol{P}$, it requires an SVD operator of $\boldsymbol{L}_{\boldsymbol{Z}}$ with complexity $\mathcal{O}(n^3)$, but package, such as PROPACK[2], can compute a rank constrained SVD with complexity $\mathcal{O}(bn^2)$, where $b=n-c$. For updating $\boldsymbol{Z}$, the main computational complexity of matrix inverse operation (i.e., $(\boldsymbol{K}+2\lambda_4\boldsymbol{I})^{-1}$) is $\mathcal{O}(n^3)$. Since the optimization of $\boldsymbol{K}$ takes closed-form solution, its computational complexity is $\mathcal{O}(n^2)$. The computational complexity of updating $\boldsymbol{w}$ is $\mathcal{O}(rn^2)$. In sum, the whole computational complexity of Algorithm 1 is $\mathcal{O}(tn^3)$, where t is the number of iterations.

[1] https://canyilu.github.io/publications/2015-report-simplex.pdf

[2] http://sun.stanford.edu/ rmunk/PROPACK/

D. Convergence Analysis

As previously mentioned, problems (13) and (24) have optimal solutions, and problem (25) has closed-form solution. In this section, we prove that each subproblem is convex, and Algorithm 1 monotonically decreases the objective function value of (12) until it converges to a locally optimal solution.

For convenience, we denote the objective of (12) as $f(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{K})$, and $S = \{\boldsymbol{Z} | 0 \leq \boldsymbol{Z}, \boldsymbol{Z}\mathbf{1} = \mathbf{1}, \operatorname{diag}(\boldsymbol{Z}) = \mathbf{0}\}$. Let the indicator function of S as $l_S(\boldsymbol{Z})$. Then, the convergence proof of each subproblem is as follows.

For updating $\boldsymbol{P}$, the Hessian matrix of (24) is

$$\boldsymbol{G_P} = \frac{\partial^2 \operatorname{Tr}(\boldsymbol{P}^T \boldsymbol{L_Z} \boldsymbol{P})}{\partial \boldsymbol{P} \partial \boldsymbol{P}^T} = \boldsymbol{L_Z} + \boldsymbol{L_Z}^T. \tag{28}$$

Due to $\boldsymbol{L_Z} \succeq 0$, the Hessian matrix $\boldsymbol{G_P}$ is also positive semidefinite (i.e., $\boldsymbol{G_P} \succeq 0$). Thus, (24) is a convex function with respect to $\boldsymbol{P}$. Moreover, we can factorize $\boldsymbol{L_Z}$ as $\boldsymbol{B}^T \boldsymbol{B}$ [49] and then obtain $\operatorname{Tr}(\boldsymbol{P}^T \boldsymbol{L}_U \boldsymbol{P}) = \|\boldsymbol{B}\boldsymbol{P}\|_F^2$. Obviously, in each iteration, we have

$$f(\boldsymbol{Z}^k, \boldsymbol{P}^{k+1}, \boldsymbol{K}^k) \leq f(\boldsymbol{Z}^k, \boldsymbol{P}^k, \boldsymbol{K}^k). \tag{29}$$

For updating $\boldsymbol{Z}$, the second-order derivative of f w.r.t $\boldsymbol{Z}$ is $\bigtriangledown^2 f = (2\boldsymbol{K} + \lambda_4 \boldsymbol{I})$ subjecting to $\bigtriangledown^2 f \succeq 0$. Note that the nonnegative and linear constraints do not affect convexity [50]. Thus, (13) is a convex function with respect to $\boldsymbol{Z}$. Moreover, analogously to [3], (13) contains a regularization term $\|\boldsymbol{Z}\|_F^2$, which makes the subproblem strongly convex, and thus, the solution is unique and stable. Therefore, we have

$$\begin{aligned} f(\boldsymbol{Z}^{t+1}, \boldsymbol{P}^{t+1}, \boldsymbol{K}^t) + l_S(\boldsymbol{Z}^{t+1}) \\ \leq f(\boldsymbol{Z}^t, \boldsymbol{P}^{t+1}, \boldsymbol{K}^t) + l_S(\boldsymbol{Z}^t) - \frac{\lambda_4}{2}\|\boldsymbol{Z}^{t+1} - \boldsymbol{Z}^t\|_F^2. \end{aligned} \tag{30}$$

Similar to $\boldsymbol{P}$, the Hessian matrix w.r.t $\boldsymbol{K}$ of f is also positive semidefinite, and (25) contains a term $\sum_i w_i \|\boldsymbol{K}^i - \boldsymbol{K}\|_F^2$, which makes the subproblem involved in updating $\boldsymbol{K}$ strongly convex. Therefore, we have

$$\begin{aligned} f(\boldsymbol{Z}^{t+1}, \boldsymbol{P}^{t+1}, \boldsymbol{K}^{t+1}) \\ \leq f(\boldsymbol{Z}^{t+1}, \boldsymbol{P}^{t+1}, \boldsymbol{K}^t) - \frac{\theta}{2}\|\boldsymbol{K}^{t+1} - \boldsymbol{K}^t\|_F^2 \end{aligned} \tag{31}$$

where $\theta = \lambda_3 \sum_{i=1}^r w_i = \lambda_3$.

Combining (29)–(31), we have

$$\begin{aligned} f(\boldsymbol{Z}^{t+1}, \boldsymbol{P}^{t+1}, \boldsymbol{K}^{t+1}) + l_S(\boldsymbol{Z}^{t+1}) \leq f(\boldsymbol{Z}^t, \boldsymbol{P}^t, \boldsymbol{K}^t) \\ + l_S(\boldsymbol{Z}^t) - \frac{\lambda_4}{2}\|\boldsymbol{Z}^{t+1} - \boldsymbol{Z}^t\|_F^2 - \frac{\theta}{2}\|\boldsymbol{K}^{t+1} - \boldsymbol{K}^t\|_F^2. \end{aligned} \tag{32}$$

Hence, $f(\boldsymbol{Z}^t, \boldsymbol{P}^t, \boldsymbol{K}^t) + l_S(\boldsymbol{Z}^t)$ decreases monotonically in each iteration until it converges to a local optimum.

We also conduct experiments to prove the convergence property of the proposed algorithm. Section IV-I shows the convergence curves and difference curves, which consistent with the above-mentioned convergence analysis.

IV. Experiments and Analysis

In this section, evaluation data sets, experimental settings, experimental results, and discussions are presented to demonstrate the effectiveness of our SPMKC-E and SPMKC methods. Because the difference between SPMKC-E and SPMKC is only the kernel weight strategy, we only report the performance of SPMKC in most cases.

TABLE II

Summaries of the Used Data Sets

Dataset	Type	#(Clusters)	#(Samples)	#(Features)
Yale	face image	15	165	1024
Jaffe	face image	10	213	676
AR	face image	120	840	768
ORL	face image	40	400	1024
COIL20	object image	20	1440	1024
BA	handwriting	36	1404	320
TR11	text	9	414	6429
TR41	text	10	878	7454
TR45	text	10	690	8261
Two-moon	synthetic	2	200	2
Three-ring	synthetic	3	300	2

A. Data Sets

We employ 11 data sets, including four famous face data sets[3] (i.e., Yale, Jaffe, AR, and ORL), an object image data set (i.e., COIL20), a binary alpha digits data set (i.e., binaryalphadigs or BA for short), three text corporas (i.e., TR11, TR41, and TR45), and two synthetic nonlinear data sets (i.e., Two-moon and Three-ring). The summaries of these data sets are listed in Table II, and some samples are shown in Fig. 2. We refer the reader to [28] and [35] for more detailed descriptions of these data sets.

B. Comparison Methods

We compare the proposed methods, SPMKC-E and SPMKC, with six state-of-the-art MKL clustering methods (see Section II-C), including MKKM [22], RMKKM [35], AASC [36], LKGr [37], SCMK [16], and SMKL [13].

For a fair comparison, the important parameters of these comparison methods are carefully tuned by following the recommended experimental settings provided by their authors. In addition, the parameters of the proposed methods are initialized as $\lambda_2 = 1$, $\lambda_4 = 1$, and $\delta = 10$ for all the experiments.

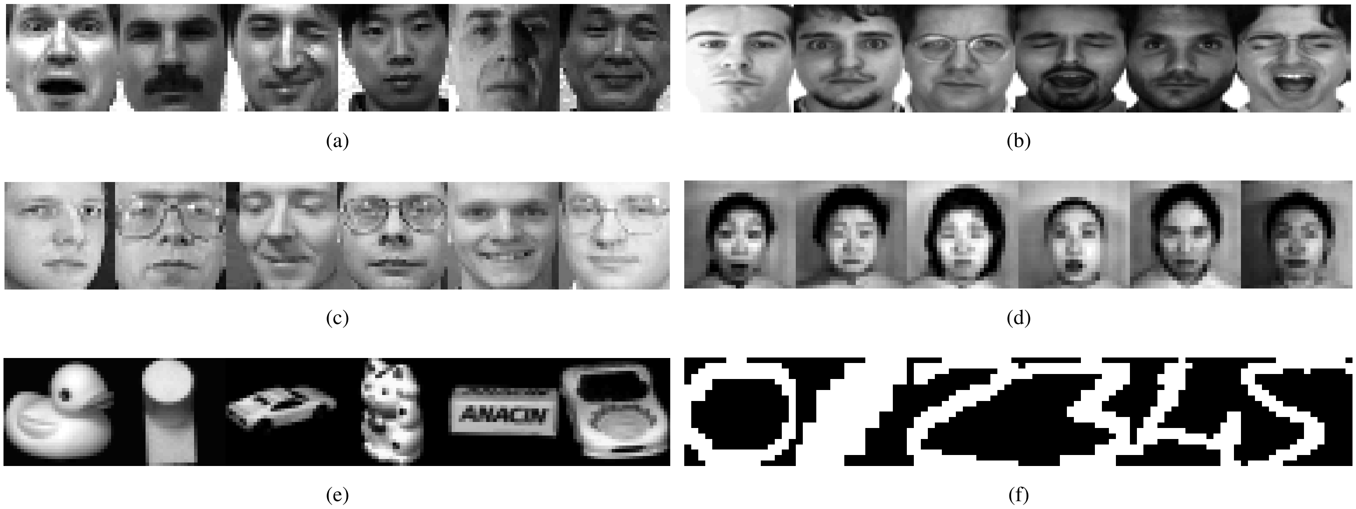
C. Evaluation Metrics

To quantitatively evaluate the clustering performance of the comparison methods for MKL clustering tasks, the widely used metrics, clustering accuracy (ACC), normalized mutual information (NMI), and purity, are applied in this article. For these widely used metrics, the larger value stands for better performance. For more detailed information on these metrics, refer to [39] and [51].

D. Multiple Kernels Construction

Following the settings in [35], we construct 12 base kernels (i.e., $r = 12$) and form a kernel pool in this article, including seven radial basis function kernels with $ker(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/(2\tau\sigma^2))$, where τ varies

[3] http://featureselection.asu.edu/data sets.php



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 2. Example images in the six image data sets that used in the experiments. (a) Yale. (b) AR. (c) ORL. (d) Jaffe. (e) COIL20. (f) BA.

TABLE III

CLUSTERING RESULTS WITH STANDARD DEVIATIONS IN TERMS OF ACC, NMI, AND PURITY

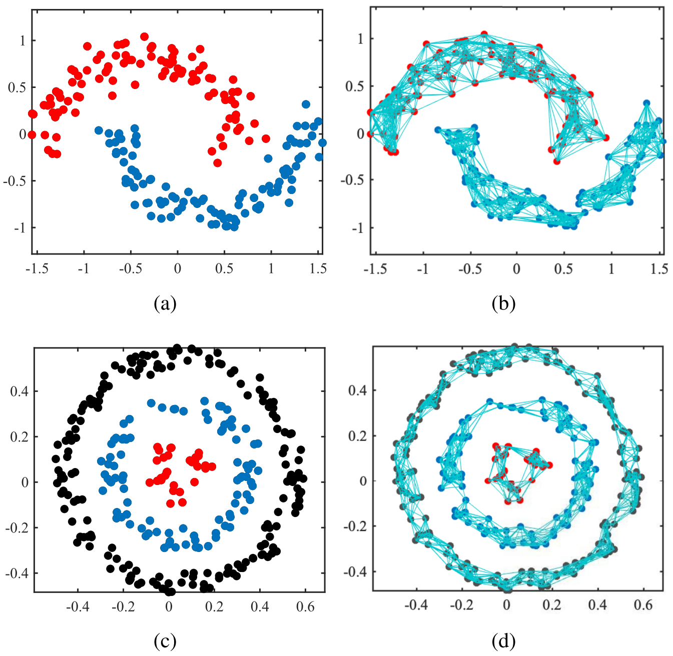
Dataset	Metrics	MKKM	RMKKM	AASC	LKGr	SCMK	SMKL	SPMKC-E	SPMKC
Yale	ACC	0.457(0.041)	0.521(0.034)	0.406(0.027)	0.540(0.030)	0.582(0.025)	0.582(0.017)	**0.658(0.000)**	**0.673(0.000)**
	NMI	0.501(0.036)	0.556(0.025)	0.468(0.028)	0.566(0.025)	0.576(0.012)	0.614(0.015)	**0.650(0.000)**	**0.660(0.000)**
	Purity	0.475(0.037)	0.536(0.031)	0.423(0.026)	0.554(0.029)	0.610(0.014)	0.667(0.014)	**0.700(0.000)**	**0.709(0.000)**
Jaffe	ACC	0.746(0.069)	0.871(0.053)	0.304(0.008)	0.861(0.052)	0.869(0.022)	0.967(0.000)	**1.000(0.000)**	**1.000(0.000)**
	NMI	0.798(0.058)	0.893(0.041)	0.272(0.006)	0.869(0.031)	0.868(0.021)	0.951(0.000)	**1.000(0.000)**	**1.000(0.000)**
	Purity	0.768(0.062)	0.889(0.045)	0.331(0.008)	0.859(0.038)	0.882(0.023)	0.967(0.000)	**1.000(0.000)**	**1.000(0.000)**
ORL	ACC	0.475(0.023)	0.556(0.024)	0.272(0.009)	0.616(0.016)	0.656(0.015)	0.573(0.032)	**0.745(0.000)**	**0.785(0.000)**
	NMI	0.689(0.016)	0.748(0.018)	0.438(0.007)	0.794(0.008)	0.808(0.008)	0.733(0.027)	**0.855(0.000)**	**0.873(0.000)**
	Purity	0.514(0.021)	0.602(0.024)	0.316(0.007)	0.658(0.017)	0.699(0.015)	0.648(0.017)	**0.780(0.000)**	**0.803(0.000)**
AR	ACC	0.286(0.014)	0.344(0.012)	0.332(0.006)	0.314(0.015)	0.544(0.024)	0.263(0.009)	**0.750(0.000)**	**0.798(0.000)**
	NMI	0.592(0.014)	0.655(0.015)	0.651(0.005)	0.648(0.007)	0.775(0.009)	0.568(0.014)	**0.889(0.000)**	**0.913(0.000)**
	Purity	0.305(0.012)	0.368(0.010)	0.350(0.006)	0.330(0.014)	0.642(0.014)	0.530(0.014)	**0.807(0.000)**	**0.858(0.000)**
COIL20	ACC	0.548(0.058)	0.667(0.028)	0.349(0.050)	0.618(0.051)	0.591(0.028)	0.487(0.031)	**0.842(0.000)**	**0.884(0.000)**
	NMI	0.707(0.033)	0.773(0.017)	0.419(0.027)	0.766(0.023)	0.726(0.011)	0.628(0.018)	**0.909(0.000)**	**0.939(0.000)**
	Purity	0.590(0.053)	0.699(0.022)	0.391(0.044)	0.650(0.039)	0.635(0.013)	0.683(0.004)	**0.907(0.000)**	**0.944(0.000)**
BA	ACC	0.405(0.019)	0.434(0.018)	0.271(0.003)	0.444(0.018)	0.384(0.014)	0.246(0.012)	**0.508(0.000)**	**0.522(0.000)**
	NMI	0.569(0.008)	0.585(0.011)	0.423(0.004)	0.604(0.009)	0.544(0.012)	0.486(0.011)	**0.632(0.000)**	**0.649(0.000)**
	Purity	0.435(0.014)	0.463(0.015)	0.303(0.004)	0.479(0.017)	0.606(0.009)	0.623(0.011)	**0.546(0.000)**	**0.634(0.000)**
TR11	ACC	0.501(0.048)	0.577(0.094)	0.472(0.008)	0.607(0.043)	0.549(0.015)	0.708(0.033)	**0.737(0.000)**	**0.748(0.000)**
	NMI	0.446(0.046)	0.561(0.118)	0.394(0.003)	0.597(0.031)	0.371(0.018)	0.557(0.068)	**0.644(0.000)**	**0.652(0.000)**
	Purity	0.655(0.044)	0.729(0.096)	0.547(0.000)	0.776(0.030)	0.783(0.011)	**0.835(0.048)**	0.807(0.000)	**0.831(0.000)**
TR41	ACC	0.561(0.068)	0.627(0.073)	0.459(0.001)	0.595(0.020)	0.650(0.068)	0.671(0.002)	**0.688(0.000)**	**0.715(0.000)**
	NMI	0.578(0.042)	0.635(0.092)	0.431(0.000)	0.604(0.023)	0.492(0.017)	0.625(0.004)	**0.654(0.000)**	**0.698(0.000)**
	Purity	0.728(0.042)	0.776(0.065)	0.621(0.001)	0.759(0.031)	0.758(0.034)	0.761(0.003)	**0.764(0.000)**	**0.771(0.000)**
TR45	ACC	0.585(0.066)	0.640(0.071)	0.526(0.008)	0.663(0.042)	0.634(0.058)	0.671(0.004)	**0.774(0.000)**	**0.781(0.000)**
	NMI	0.562(0.056)	0.627(0.092)	0.420(0.014)	0.671(0.020)	0.584(0.051)	0.622(0.007)	**0.739(0.000)**	**0.758(0.000)**
	Purity	0.691(0.058)	0.752(0.074)	0.575(0.011)	0.800(0.026)	0.728(0.048)	0.816(0.004)	**0.809(0.000)**	**0.816(0.000)**
Avg	ACC	0.507(0.045)	0.582(0.045)	0.377(0.013)	0.584(0.032)	0.607(0.030)	0.574(0.016)	**0.745(0.000)**	**0.767(0.000)**
	NMI	0.605(0.034)	0.670(0.048)	0.435(0.010)	0.680(0.020)	0.638(0.018)	0.643(0.018)	**0.775(0.000)**	**0.794(0.000)**
	Purity	0.573(0.038)	0.646(0.042)	0.429(0.012)	0.650(0.027)	0.705(0.020)	0.726(0.013)	**0.791(0.000)**	**0.818(0.000)**

* For each separate datasets, the last three rows are the average performance, and the best two results are highlighted in boldface. Note that we fix $\lambda_2 = 1$ and $\lambda_4 = 1$ in all experiments. If we tune λ_4 rather than a fixed value, we could achieve better clustering performance.

in the range of $\{0.01, 0.05, 0.1, 1, 10, 50, 100\}$ and σ is the maximum distance between samples; four polynomial kernels with $ker(x_i, x_j) = (a + x_i^T x_j)^b$, where $a = \{0, 1\}$ and $b = \{2, 4\}$ are selected; a cosine kernel with $ker(x_i, x_j) = (x_i^T x_j)/(\|x_i\| \cdot \|x_j\|)$. Finally, all the kernel matrices $\{\boldsymbol{K}^k\}_{k=1}^r$ are normalized to [0, 1] range.

E. Performance Evaluation on Synthetic Nonlinear Data Set

In order to evaluate our SPMKC in a visual manner, we use two synthetic nonlinear data sets, i.e., Two-moon data set and Three-ring data set, which are generated with a moon and a ring pattern, respectively. As shown in Fig. 3(a) and (c), different colors indicate different clusters.



(a) (b)

(c) (d)

Fig. 3. Results produced by SPMKC on two synthetic data sets. Only edges with weights greater than or equal to 0.001 are shown (b) and (d). (a) Two-moon data set. (b) Learned $\mathbf{Z}$. (c) Three-ring data set. (d) Learned graph $\mathbf{Z}$.

TABLE IV

Computational Time (in seconds) Comparison

Method	Yale	ORL	TR11	TR45
MKKM	0.057(0.026)	0.227(0.212)	0.319(0.036)	0.482(0.015)
RMKKM	0.221(0.041)	0.909(0.047)	0.976(0.078)	1.686(0.090)
AASC	0.795(0.063)	3.622(0.171)	1.143(0.102)	2.918(0.111)
LKGr	3.162(0.070)	42.039(2.012)	50.733(2.184)	208.815(5.407)
SCMK	1.740(0.057)	7.425(0.271)	13.717(0.370)	5.558(0.315)
SMKL	1.415(0.062)	12.835(0.325)	9.863(0.218)	154.662(3.205)
SPMKC	0.604(0.051)	2.917(0.140)	2.603(0.090)	6.801(0.325)

Fig. 3(b) and (d) shows the learned affinity graph $\mathbf{Z}$. It can be seen that there is no line between different clusters, and all the points within the same cluster are connected together. Thus, as a kernel method, SPMKC can handle the data well with nonlinearity. Significantly, SPMKC can preserve the true cluster structure and obtain the required numbers of connected graph components.

F. Performance Evaluation on Benchmark Data Set

The average clustering results of 20 times independent experiments of all the comparison methods are presented in Table III. Note that the last three rows are the mean ACC, NMI, and purity of each method for each separate data sets. For each data set and metric, the best two results are highlighted in boldface.

Obviously, it can be seen from Table III that the proposed methods SPMKC-E and SPMKC consistently obtain the best performance, followed by SCMK, LKGr, RMKKM, SMKL, MKKM, and AASC. More precisely, SPMKC improved by 16.08%, 15.53%, and 11.37%, respectively, compared with SCMK (the best comparison method) in terms of ACC, NMI, and purity. Moreover, it is also worth noting that the proposed SPMKC significantly outperforms the proposed SPMKC-E by over 2.27%, 1.89%, and 2.73% improvements in terms of ACC, NMI, and purity, respectively. Note that the standard deviations of SPMKC are zero; the reason is that SPMKC can automatically tune λ_2 to control the number of connected subgraphs of graph $\mathbf{Z}$, and if the number of connected subgraphs is equal to c, a stable output can be obtained when performing spectral clustering.

Furthermore, to show the effectiveness of the proposed methods further, we illustrate the affinity matrices learned from the Yale and Jaffe data sets by using a visual assessment similar to [52]. As shown in Figs. 4 and 5, our methods encourage us to learn a dense and distinct affinity graph with c connected components exactly, where c is the number of the clusters.

Consequently, these results clearly exhibit the superior clustering performance of SPMKC-E and SPMKC, which well demonstrates their effectiveness and superiority. For better flow, the detailed discussions are given in Section IV-J.

G. Parameter Sensitivity

To verify if the proposed SPMKC method is sensitive to the involved parameters, in this section, we present the clustering performance with different parameter values. Although we have up to four parameters (i.e., λ_1, λ_2, λ_3, and λ_4) in our model, the λ_2 is a self-tune one via (27), and the best results can always be achieve when λ_4 is fixed as $\lambda_4 = 1$. We can, therefore, assume that the proposed method actually only has two parameters (i.e., λ_1 and λ_3). Using a grid search strategy, the searching regions of λ_1 and λ_3 are selected from $\{1, 2, 3, 4, 5, 6\}$ and $\{1, 10, 100, 200, 400, 1000\}$, respectively. The results are shown in Fig. 6. Obviously, it can be observed that competitive performance is obtained over a wide range of parameters, i.e., SPMKC can achieve competitive performance on the Yale and ORL data sets while setting $\lambda_1 \in [3, 5]$ and $\lambda_3 > 100$. Due to the limitation of the space, the sensitivity of parameters on other data sets are not shown in this article, but similar conclusions can be obtained.

H. Computational Time

Furthermore, we give the average computation time (in seconds) of all comparison MKL methods on the Yale, ORL, TR11, and TR45 data sets. Our experimental platform is MATLAB 2016b operating on a Mac mini 2018 with an Intel Core i7 (3.2 GHz) CPU and 16-GB RAM. In order to enhance the comparability among the methods, we set the same convergence conditions for each algorithm during the experiment. As shown in Table IV, the computational cost of the proposed SPMKC method is lower than SCMK, SMKL, and LKGr, whose clustering performance is also worse than ours. In addition, such as MKKM, AASC and RMKKM, their computational costs are less than SPMKC's, but their clustering performance is poor.

I. Convergence Study

This section presents an experimental study on the convergence of the proposed SPMKC method on the Yale, ORL,

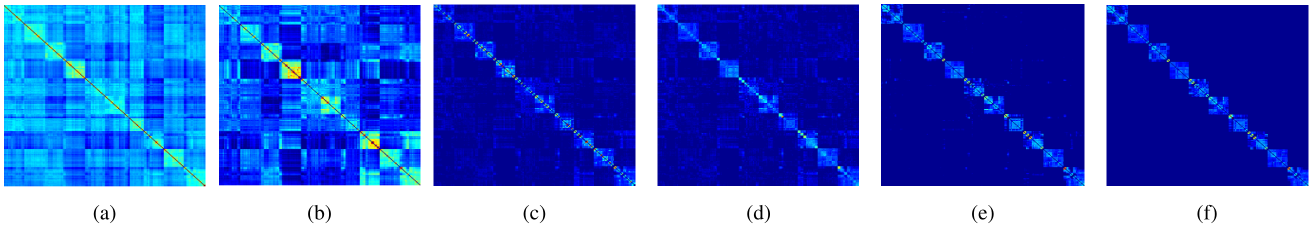


Fig. 4. Visualization of the learned affinity matrices on the Jaffe data set from AASC, LKGr, SMKL, SCMK, SPMKC-E, and SPMKC, respectively. The Jaffe data set consists of ten clusters. The darker the blue color, the value is closer to zero. (a) AASC. (b) LKGr. (c) SMKL. (d) SCMK. (e) Our SPMKC-E. (f) Our SPMKC.

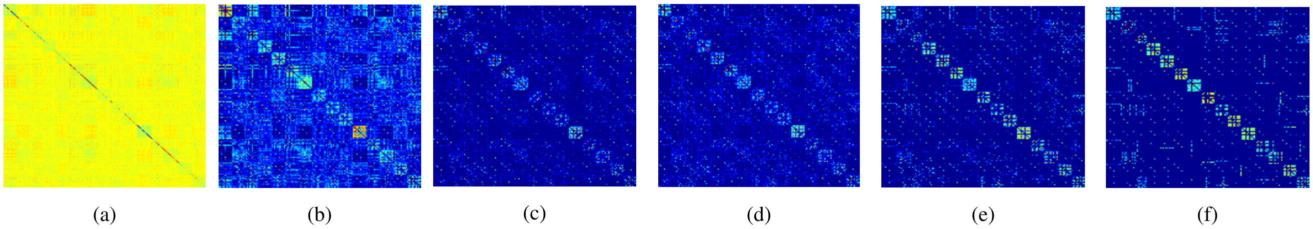


Fig. 5. Visualization of the learned affinity matrices on the Yale data set from AASC, LKGr, SMKL, SCMK, SPMKC-E, and SPMKC, respectively. The Yale data set consists of 15 clusters. The darker the blue color, the value is closer to zero (a) AASC. (b) LKGr. (c) SMKL. (d) SCMK. (e) Our SPMKC-E. (f) Our SPMKC.

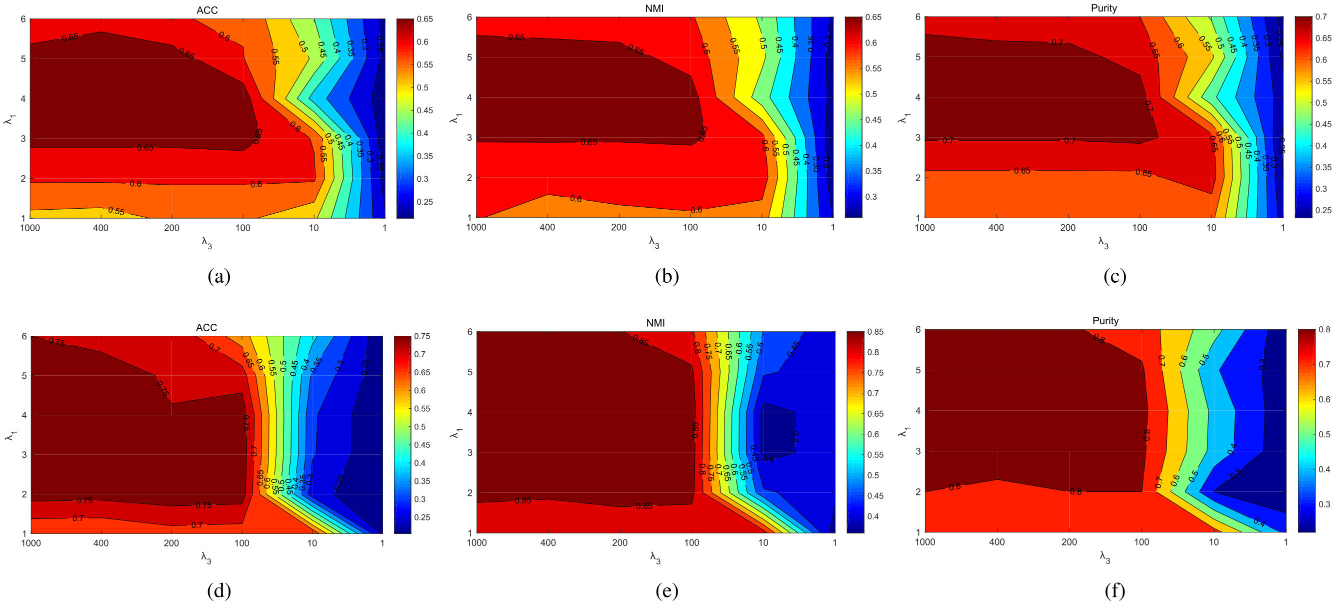
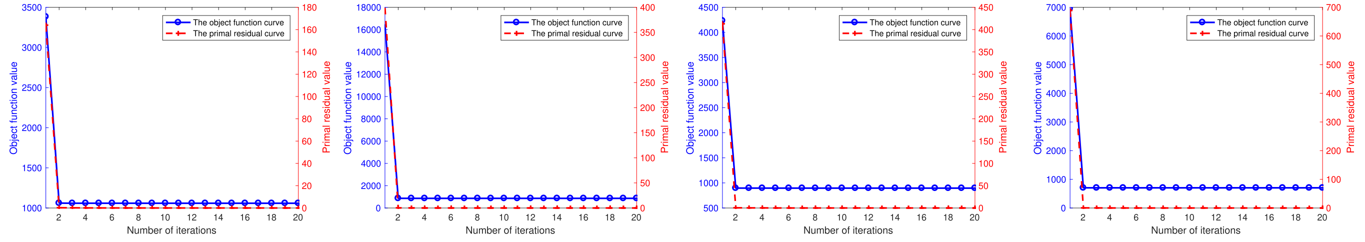


Fig. 6. Clustering performance in terms of ACC, NMI, and purity of the proposed SPMKC method w.r.t. λ_1 and λ_3 on the Yale (the first row) and ORL (the second row) data sets. (a) ACC. (b) NMI. (c) Purity. (d) ACC. (e) NMI. (f) Purity.

TR11, and TR45 data sets. The convergence curves and the difference curves are shown in Fig. 7, where the convergence curve and the difference curve are defined as $(1/2\operatorname{Tr}(\boldsymbol{K} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}) - \lambda_1 \operatorname{Tr}(\boldsymbol{K}\boldsymbol{Z}) + \lambda_2 \operatorname{Tr}(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}) + \lambda_3 \sum_{k=1}^{r} w_k \|\boldsymbol{K}^k - \boldsymbol{K}\|_F^2 + \lambda_4 \|\boldsymbol{Z}\|_F^2)^{t+1}$ and $\|\boldsymbol{Z}^{t+1} - \boldsymbol{Z}^t\|_F$, respectively. As seen in Fig. 7, our SPMKC converges quickly in the beginning within about four iterations and steadily on each data set. In addition, all the difference curves also quickly reduce close to zero. Note that the convergence criterion is $\operatorname{rank}(\boldsymbol{L}_Z) = n - c$, i.e., the required numbers of the connected components of graph $\boldsymbol{Z}$ is equal to c. So although the objective value becomes stable quickly within a few iterations, some additional iterations are needed to continuously self-tune parameter λ_2 until the required graph is obtained. The experimental results also demonstrate our theory proof (see Section III-D) that SPMKC can converge to a locally optimal solution.

J. Analysis and Discussion

For better readability, we transcribe some involved acronyms here: (6) is KGS, (9) is KALSL, (10) is multiple kernel learning (MKL); and (11) is KAWS. Besides, the



(a) (b) (c) (d)

Fig. 7. Convergence curves and difference curves of the proposed SPMKC method on four data sets (a) Yale. (b) ORL. (c) TR11. (d) TR45.

proposed SPMKC-E and SPMKC methods are collectively called SPMKCs. From Figs. 3–7 and Tables III and IV, the following observations are obtained.

1) We have accomplished some clustering experiments on nine popular data sets. Three evaluation metrics, ACC, NMI and purity, are adopted. According to the results in Table III, we have the following.
   a) It is obvious that the proposed SPMKCs consistently obtain the best performance, which confirms the effectiveness of them in dealing with clustering tasks.
   b) The standard deviations of SPMKCs are zero, since their learned affinity graphs have exact c connected components by self-tuning parameter λ_2. But other methods cannot generate exact c blocks due to fixed tradeoff parameter. This means that SPMKCs have good stability on statistical significance.
   c) The proposed SPMKCs are more adaptability, which are not limited for data sets. In other words, SPMKCs perform consistently on all data sets, but other methods have "short board." For instance, SMKL performs poorly on the AR, COIL20, and BA data sets, LKGr performs poorly on the AR data set, and SCMK performs poorly on the BA data set.
   d) In most cases, the spectral clustering based methods, SPMKCs, AASC, LKGr, SMKL, SCMK, are superior to the k-means-based methods in general, such as MKKM and RMKKM. It indicates that the spectral clustering based methods usually achieve better performance than the standard k-means based ones.
   e) The performance of SPMKC consistently outperform SPMKC-E's. This proves that the proposed KAWS (i.e., $w_k = \exp(-\delta e_k/\overline{e})/\sum_{k=1}^{r}\exp(-\delta e_k/\overline{e})$) is better than the kernel weight strategy EDWS proposed in [13] (i.e., $w_k = 1/2\|\boldsymbol{K}^k - \boldsymbol{K}\|_F$). By KAWS, we can yield a better optimal consensus kernel in order to obtain better clustering performance.
   f) Compared with SMKL and SCMK, the proposed method SPMKC-E simultaneously preserves the global and local structures of the data in kernel space, both of which contain important discriminative information for graph learning [43]. Experimental results also demonstrate that SPMKC-E achieves much better performance than SMKL and SCMK. This illustrates that the structural information in kernel space is effective and beneficial to affinity graph learning.
2) As mentioned in Section I, the optimal affinity graph has two characteristics for accurate clustering. We have illustrated the learned affinity graphs from the Yale and Jaffe data sets in Section IV-F. According to the results in Figs. 4 and 5, we have the following.
   a) Visually, there are 15 and 10 diagonal blocks for the Yale and Jaffe data sets, respectively. It is obvious that the proposed SPMKCs methods can yield dense and distinct affinity matrices with a better block diagonal structure. The main reasons are that, on the one hand, the rank constraint of Laplacian matrix (i.e., $\text{rank}(\boldsymbol{L_Z}) = n - c$) urges to learn an affinity graph with exactly c connected components (where c is the number of clusters); on the other hand, the KGS term [i.e., $1/2\,\text{Tr}(\boldsymbol{K} - 2\boldsymbol{KZ} + \boldsymbol{Z}^T\boldsymbol{KZ}) + \|\boldsymbol{Z}\|_F^2$] and the KALSL term [i.e., $-\text{Tr}(\boldsymbol{K}^T\boldsymbol{Z}) + \gamma\|\boldsymbol{Z}\|_F^2$] can capture the accurate similarity relations between samples. These results, again, demonstrate the effectiveness of the proposed structure-preserving terms.
3) As previously mentioned in Section IV-G, the proposed SPMKC method actually involves two parameters (i.e., λ_1 and λ_3). In other words, in all experiments, we fix $\lambda_2 = 1$ and $\lambda_4 = 1$, and tune λ_1 and λ_3. Note that λ_1 is used to control KALSL term that can preserve the local structure of the input data in kernel space, and λ_3 is used to control KAWS-based MKL learning term that promotes to learn an optimal consensus kernel. According to the results in Fig. 6, we have the following.
   a) The proposed SPMKC is insensitive to the involved parameters within a wide range. From Fig. 6, it can be seen that $3 < \lambda_1 < 5$ and $\lambda_3 > 100$ might be a good choice. This illustrates the proposed method is easy to be tamed.
   b) The proposed KALSL term plays an important role in enhancing clustering performance. Obviously, our final objective function [(12)] contains a twins term [i.e., $1/2\,\text{Tr}(\boldsymbol{K} + \boldsymbol{Z}^T\boldsymbol{KZ}) - \lambda_1\,\text{Tr}(\boldsymbol{KZ})$]. If we fix $\lambda_1 = 1$, the twins term can be rewritten as

$1/2\,\mathrm{Tr}(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}) = \min 1/2\|\phi(\boldsymbol{X}) - \phi(\boldsymbol{X})\boldsymbol{Z}\|_F^2$, which is the widely used kernel self-expression graph learning in (2). Importantly, if $\lambda_1 > 1$, the twins term can be rewritten as $1/2\,\mathrm{Tr}(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}) - (\lambda_1 - 1)\,\mathrm{Tr}(\boldsymbol{K}\boldsymbol{Z})$. From Fig. 5, it can be seen that the best clustering performance are obtained when $3 < \lambda_1 < 5$, rather than $\lambda_1 = 1$; therefore, the KALSL is proven to be effective. It is worth mentioning that λ_1 should be lower bounded 1 in order to guarantee the integrity of $1/2\,\mathrm{Tr}(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z})$ in (2).

4) For clustering, the computation cost is important to report. We have analyzed the computational complexity and the convergence of the proposed algorithm in Sections III-C and III-D, respectively. Furthermore, we have given the time and convergence experiments in Sections IV-H and IV-I, respectively. According to the results in Table IV and Fig. 7, we have the following.
   a) Theoretically and experimentally, the proposed method can converge fast to a locally optimal solution, and hence, it is an efficient MKL clustering method.

V. Conclusion

This article presented a novel method SPMKC for MKL subspace clustering, which simultaneously performs consensus kernel learning from multiple base kernels and learns an affinity graph in kernel space. The proposed kernel weight strategy can automatically assign an optimal weight for each base kernel without additional constraints such that the consensus kernel is automatically synthesized. Furthermore, the obtained optimal affinity graph has two advantages. On the one hand, the affinity graph can be directly partitioned into exact c connected components if there are c clusters. On the other hand, the affinity graph can preserve the global and local structure of the data in kernel space, which is crucial to the performance of clustering. We proposed an efficient and fast algorithm to optimize SPMKC. Extensive experiments on nine benchmark clustering data sets demonstrated that the proposed method is very efficient and achieves better clustering performance (in terms of ACC, NMI, and purity) than state-of-the-art MKL clustering methods.

This article opens up several intriguing directions for future research. First, we are interested in applying the proposed method to semisupervised learning and classification problems when data are partially labeled [53]. Second, it is also worth exploring to consider the canonical correlations [54] of kernels to improve the quality of the consensus kernel. Third, we will apply the proposed method to the multiview clustering problem [45].

Acknowledgment

The authors would like to thank the editors and anonymous reviewers who provided substantive suggestions for improving their work.

References

[1] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, "Deep self-evolution clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 809–823, Apr. 2020.

[2] X. Li, Q. Lu, Y. Dong, and D. Tao, "Robust subspace clustering by cauchy loss function," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2067–2078, Jul. 2019.

[3] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, Feb. 2019.

[4] Z. Deng, K.-S. Choi, Y. Jiang, J. Wang, and S. Wang, "A survey on soft subspace clustering," *Inf. Sci.*, vol. 348, pp. 84–106, Jun. 2016.

[5] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2988–3001, Jun. 2017.

[6] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, "Motion segmentation & multiple object tracking by correlation co-clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 140–153, Jan. 2020.

[7] H. Jia and Y.-M. Cheung, "Subspace clustering of categorical and numerical data with an unknown number of clusters," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3308–3325, Aug. 2018.

[8] 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.

[9] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, "Low-rank sparse subspace for spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1532–1543, Aug. 2019.

[10] 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, Jan. 2019.

[11] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2849–2853.

[12] C. Zhang, F. Nie, and S. Xiang, "A general kernelization framework for learning algorithms based on kernel PCA," *Neurocomputing*, vol. 73, nos. 4–6, pp. 959–967, Jan. 2010.

[13] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," in *Proc. IJCAI*, Jul. 2018, pp. 2312–2318.

[14] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn. (ICML-10)*, 2010, pp. 663–670.

[15] 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.

[16] Z. Kang, C. Peng, Q. Cheng, and Z. Xu, "Unified spectral clustering with optimal graph," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3366–3373.

[17] J. Wen, B. Zhang, Y. Xu, J. Yang, and N. Han, "Adaptive weighted nonnegative low-rank representation," *Pattern Recognit.*, vol. 81, pp. 326–340, Sep. 2018.

[18] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1302–1308.

[19] N. Zhou, Y. Xu, H. Cheng, J. Fang, and W. Pedrycz, "Global and local structure preserving sparse subspace learning: An iterative approach to unsupervised feature selection," *Pattern Recognit.*, vol. 53, pp. 87–101, May 2016.

[20] H. Wang, Y. Yang, B. Liu, and H. Fujita, "A study of graph-based system for multi-view clustering," *Knowl.-Based Syst.*, vol. 163, pp. 1009–1019, Jan. 2019.

[21] Y. Han, K. Yang, Y. Yang, and Y. Ma, "On the impact of regularization variation on localized multiple kernel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2625–2630, Jun. 2018.

[22] 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.

[23] 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.

[24] Y. Li, J. Liu, H. Lu, and S. Ma, "Learning robust face representation with classwise block-diagonal structure," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2051–2062, Dec. 2014.

[25] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, and Z. Lin, "Correntropy induced L2 graph for robust subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1801–1808.

[26] Z. Zhang, Y. Xu, L. Shao, and J. Yang, "Discriminative block-diagonal representation learning for image recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3111–3125, Jul. 2018.

[27] X. Xie, X. Guo, G. Liu, and J. Wang, "Implicit block diagonal low-rank representation," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 477–489, Jan. 2018.

[28] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 977–986.

[29] X. Zhu, S. Zhang, R. Hu, Y. Zhu, and J. Song, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 517–529, Mar. 2018.

[30] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1501–1511, Mar. 2018.

[31] F. Nie *et al.*, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. IJCAI*, 2016, pp. 1881–1887.

[32] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1984–1993, Oct. 2019.

[33] L. Du and Y.-D. Shen, "Unsupervised feature selection with adaptive structure learning," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2015, pp. 209–218.

[34] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1888–1894.

[35] L. Du *et al.*, "Robust multiple kernel k-means using l21-norm," in *Proc. IJCAI*, 2015, pp. 3476–3482.

[36] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 773–780.

[37] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl.-Based Syst.*, vol. 163, pp. 510–517, Jan. 2019.

[38] C. Wang, E. Zhu, X. Liu, L. Gao, J. Yin, and N. Hu, "Multiple kernel clustering with global and local structure alignment," *IEEE Access*, vol. 6, pp. 77911–77920, 2018.

[39] S. Zhou *et al.*, "Multiple kernel clustering with neighbor-kernel subspace segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1351–1362, Apr. 2020.

[40] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.

[41] Y. Chen and Z. Yi, "Locality-constrained least squares regression for subspace clustering," *Knowl.-Based Syst.*, vol. 163, pp. 51–56, Jan. 2019.

[42] R. Andreani, L. D. Secchin, and P. J. S. Silva, "Convergence properties of a second order augmented lagrangian method for mathematical programs with complementarity constraints," *SIAM J. Optim.*, vol. 28, no. 3, pp. 2574–2600, Jan. 2018.

[43] 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.

[44] H. Huang, G. Shi, H. He, Y. Duan, and F. Luo, "Dimensionality reduction of hyperspectral imagery based on spatial-spectral manifold learning," *IEEE Trans. Cybern.*, early access, Mar. 2019, doi: 10.1109/TCYB.2019.2905793.

[45] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. IJCAI*, Aug. 2017, pp. 2564–2570.

[46] A. Beck, "On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes," *SIAM J. Optim.*, vol. 25, no. 1, pp. 185–209, Jan. 2015.

[47] J. Wen, X. Fang, Y. Xu, C. Tian, and L. Fei, "Low-rank representation with adaptive graph regularization," *Neural Netw.*, vol. 108, pp. 83–96, Dec. 2018.

[48] M. Iliadis, H. Wang, R. Molina, and A. K. Katsaggelos, "Robust and low-rank representation for fast face identification with occlusions," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2203–2218, May 2017.

[49] M. Wei, J. Huang, X. Xie, L. Liu, J. Wang, and J. Qin, "Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 10, pp. 2910–2926, Oct. 2019.

[50] A. Mian, Y. Hu, R. Hartley, and R. Owens, "Image set based face recognition using self-regularized non-negative coding and adaptive distance metric learning," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5252–5262, Dec. 2013.

[51] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2019.

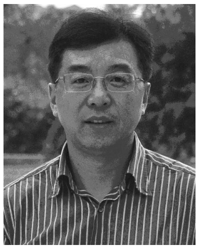
[52] J. C. Bezdek and R. J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3, 2002, pp. 2225–2230.

[53] A. H. Akbarnejad and M. S. Baghshah, "An efficient semi-supervised multi-label classifier capable of handling missing labels," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 229–242, Feb. 2019.

[54] J. Chen, G. Wang, Y. Shen, and G. B. Giannakis, "Canonical correlation analysis of datasets with a common source graph," *IEEE Trans. Signal Process.*, vol. 66, no. 16, pp. 4398–4408, Aug. 2018.

Zhenwen Ren (Member, IEEE) received the M.S. degree in communication and information system from the Southwest University of Science and Technology (SWUST), Mianyang, China, in 2014. He is currently pursuing the Ph.D. degree in control science and engineering with the Nanjing University of Science and Technology, Nanjing, China.

He is currently with the Department of School of National Defence Science and Technology, SWUST. His research interests include image set classification, subspace clustering, and deep learning.



Quansen Sun received the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology (NJUST), China, in 2006.

He is currently a Professor with the Department of Computer Science, NJUST. He was a Visiting Researcher with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, in 2004 and 2005, respectively. His current interests include pattern recognition, image processing, remote sensing information system, and medicine image analysis.