INS 15771
20 August 2020
ARTICLE IN PRESS
No. of Pages 18, Model 3G

Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering

Zhenwen Ren [a,b], Haoyun Lei [a], Quansen Sun [b,*], Chao Yang [a,*]

[a] School of National Defense Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China
[b] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

ARTICLE INFO

Article history:
Received 10 January 2020
Received in revised form 11 June 2020
Accepted 15 August 2020
Available online xxxx

Keywords:
Subspace clustering
Kernel method
Multiple kernel learning
Affinity matrix

ABSTRACT

Due to the effectiveness of handling non-linear data and avoiding kernel customization, multiple kernel clustering (MKC) has been widely investigated and achieved promising results for challenging non-linear clustering tasks. Generally, the existing MKC methods mainly consist of first learning a coefficient matrix by leveraging multiple kernel learning (MKL) and sample self-expressiveness property, and then constructing an affinity graph relying on this coefficient matrix to accomplish spectral clustering. However, the quality of the affinity matrix (graph) is largely determined by the learned coefficient matrix, thus the two independent steps are not conducive to learn an optimal affinity graph. To tackle this problem, in this paper, we propose a new MKC method that uses one-step learning scheme rather than two-step learning scheme to learn an affinity graph, termed SLMKC. Specifically, SLMKC bridges the relationship between the affinity matrix and coefficient matrix by an adaptive local structure learning strategy, so it can simultaneously learn both in a mutual promotion manner. Furthermore, a self-weighted MKL strategy is introduced to learn an optimal consensus kernel, which can avoid selecting a specific kernel function and tuning its associated parameters. Extensive experiments validate that our SLMKC outperforms the state-of-the-art MKC competitors significantly.

1. Introduction

Clustering is one of the most fundamental and important research topic in the realm of machine learning and data mining, which aims to partition the given unlabeled samples into disjoint groups [8]. To date, clustering problem has been widely researched in the last few decades and many typical methods have been surged in several studies [10,45,10,21]. In general, the existing clustering methods can be roughly classified into five categories, including iterative based methods [11], statistical based methods [12], algebraic based methods [20], matrix factorization based methods [26], and spectral clustering based methods [20].

Among these methods, spectral clustering based methods have become extremely popular for challenging clustering tasks, thanks to their capabilities to handle arbitrarily shaped clusters and their well-defined mathematical principles. Traditionally, such type of methods splits the clustering process into two independent steps. In the first one, an affinity graph/-matrix is constructed by exploiting the given unlabeled data. In the second one, spectral clustering algorithm is employed to obtain the final clustering assignments relying on this affinity graph. There is no arguing that the first step is the most impor-

* Corresponding authors.
E-mail addresses: sunquansen@njust.edu.cn (Q. Sun), ychao1983@126.com (C. Yang).

https://doi.org/10.1016/j.ins.2020.08.056
0020-0255/

tant one since the performance of spectral clustering largely depends on constructing a desired affinity graph. Although these spectral clustering based methods can produce good performance, there are two great challenges that limit their applications in real-world settings: (1) how to handle non-linear data when they are not strictly collected from linearly independent subspaces; (2) how to learn an optimal affinity graph for spectral clustering purpose.

In order to tackle the challenging problem of how to handle non-linear data, single kernel learning (SKL) method has been in common use [39,1,4]. Theoretically, SKL maps the non-linear data from the original space into the Reproducing Kernel Hilbert Space (RKHS) where the linear property of the data is satisfied [2]. In practice, the kernel trick [3] is applied to avoid the explicit mapping, which is often computationally cheaper than the explicit computation of the projected coordinates. Although promising performance has been reported in a number of tasks, SKL methods have to require the user to select and tune a pre-defined kernel [16] since their performance is largely determined by the choice of kernel function. This is not user-friendly since the most suitable kernel function for a specific task or dataset is usually challenging to decide. Therefore, how to select an appropriate pre-defined single kernel function and how to tune its associated kernel parameters are worthy to study. In recent years, a powerful and flexible learning model, multiple kernel learning (MKL) [42,7,46,35], has attracted more and more attention to avoid kernel customization by constructing a consensus kernel from multiple candidate base kernels. More importantly, MKL has the great potential to fully exploit complementary information between these candidate kernels. Generally, MKL has been widely used in clustering [14], object recognition [7], and classification [31]. For illustrative purposes, the thoughts of the SKL and MKL are illustrated in Fig. 1(a) and (b), respectively.

In addition, to learn a better affinity graph for spectral clustering [18], many scholars have proposed different methods to learn an expected affinity graph [39,29]. Overall, the mainstream technologies can be typically divided into three main categories. The first category is to construct a predefined similarity graph as affinity graph, such as complete graph and k-nearest graph. Usually, the complete graph is constructed based on binary $(0 - -1)$ similarity, cosine similarity, or gaussian kernel similarity [41], while k-nearest graph selects its neighbor nodes by using k Nearest Neighbor algorithm. The second category is the adaptive neighbors graph learning [33,17], which builds a graph by assigning a probability for each sample as the neighborhood of another samples. Accordingly, the homogeneous samples have high affinity (similarity) values, while those heterogeneous samples have low affinity values. The third category is based on the sample self-expressiveness property [29], which reconstructs every data point by a linear combination of all other data points, and then produces a coefficient matrix that is used to construct an affinity graph. To the best of our knowledge, the self-expressiveness property based technology has been the most widely studied and obtained impressive performance in a wide variety of applications. This paper focuses on the sample self-expressiveness property [29] to construct an affinity graph used for spectral clustering.

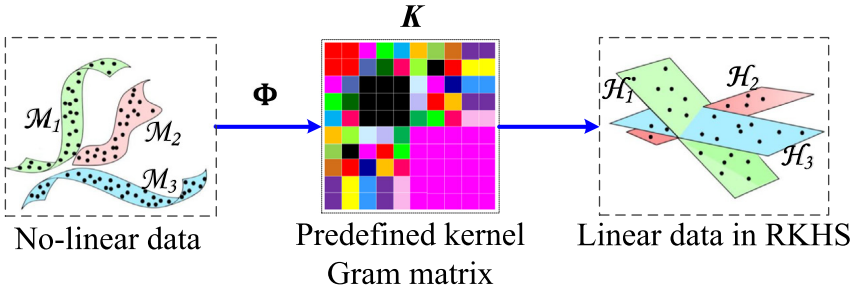
Combining both MKL and self-expressiveness, many multiple kernel clustering (MKC) methods have been proposed [27,14,9,13,16,19]. Overall, these existing MKC methods usually first learn a self-expressiveness coefficient matrix and then construct an affinity matrix to accomplish spectral clustering independently. However, the quality of affinity matrix is largely determined by the coefficient matrix. Obviously, such two-step learning scheme is not conductive to learn an optimal affinity matrix, thus impairs the final clustering performance greatly.

To effectively handle non-linear data and learn an optimal affinity matrix for challenging clustering, this paper proposes a novel method, namely Simultaneous Learning Self-expressiveness Coefficients and Affinity Matrix Multiple Kernel Clustering (SLMKC). Different from the existing MKC methods, SLMKC proposes a unified optimization objective function that uses one-step learning scheme rather than two-step learning scheme to learn an optimal affinity matrix in multiple kernels scene. Specifically, the relationship between the coefficient matrix and the affinity matrix is bridged via adaptive local structure learning. Meanwhile, SLMKC fuses multiple candidate base kernels to learn an optimal consensus kernel. Furthermore, to ensure that the learned affinity graph contains exactly c connected components, a constrained Laplacian rank term is introduced; to remove the negative effects of redundancy affinity graph edges, a top-k probabilistic neighbors selection strategy is also employed. Consequently, a high-quality affinity graph is subsequently obtained. In summary, the contributions of this paper are summarized below.

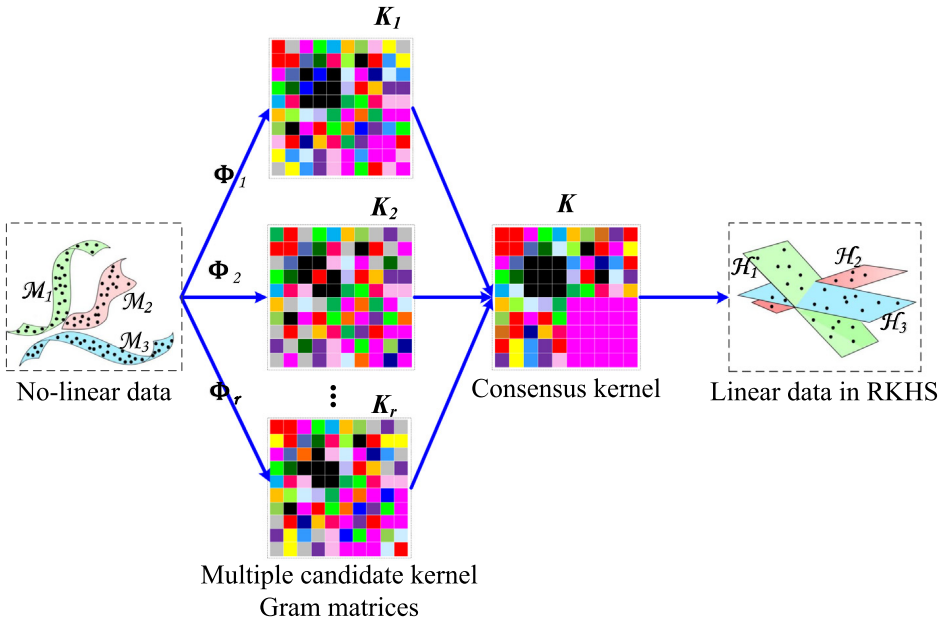
- In order to more effectively handle the non-linear data and learn an optimal affinity graph for clustering purpose, SLMKC can address the difficult problem of how to select an appropriate single kernel function and tune the associated parameters of the selected kernel function.
- Different from existing MKC methods, which obey a two-step learning scheme to learn the coefficient matrix and the affinity matrix independently, SLMKC proposes a unified optimization model to simultaneously learn the consensus kernel, the self-expressiveness coefficient matrix, and the affinity matrix. Thus, SLMKC can address the challenging problem of performance degradation.
- Comprehensive experimental results accomplished on nine benchmark datasets (i.e., six image datasets and three text datasets) demonstrate that SLMKC substantially improves the clustering performance and computational cost, compared with the existing state-of-the-art MKC methods.

The rest of this paper is structured as follows. Section 2 revisits the related self-expressiveness learning model and MKC methods for clustering. Then, the proposed SLMKC method and its optimization algorithm and complexity are given in Section 3. Finally, Section 4 and Section 5 offer the sufficient experimental results and conclusion, respectively.

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056



(a) Illustration of SKL based methods.



(b) Illustration of MKL based methods

Fig. 1. Illustrations of SKL based methods and MKL based methods. Different from SKL based methods, which use only one kernel function to map non-linear data from the original space to RKHS, MKL based methods integrate multiple candidate kernels to learn a consensus kernel. By doing so, MKL can avoid selecting and tune a single pre-defined kernel function; meanwhile, can strongly promote to effectively use the complementary information of these candidate kernels.

2. Related works

In this section, we first present the notations used throughout the paper, and then briefly revisit the self-expressiveness learning model and the related MKC methods.

2.1. Notations

Throughout this paper, we denote scalars by normal italic letters, matrices by boldface uppercase letters, and vectors by boldface lowercase letters, respectively. Some important notations are summarized in Table 1. Note here that the data matrix drawn from c independent subspaces is denoted by $\boldsymbol{X} = \{\boldsymbol{X}_i | \boldsymbol{X}_i \in \mathbb{R}^{d\times n_i}\}_{i=1}^{c}$, where $\boldsymbol{X}_i = [x_i, x_2, \cdots, x_{n_i}]$ and $\sum_{i=1}^{c} n_i = n$.

2.2. Self-expressiveness learning model

As we know, self-expressiveness property is derived based on the fact that every sample can be represented by a linear combination of all other samples [29], and subspace clustering methods rely on such property to construct a self-expressiveness coefficient matrix. Mathematically, the self-expressiveness learning model is defined as follows:

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

Table 1
Notations used throughout the paper.

Notation	Definition
$\boldsymbol{X} \in \mathbb{R}^{d\times n}$	The column-wise data matrix
x_j or $\boldsymbol{X}_j$	The jth column of $\boldsymbol{X}$
x_{ij} or $\boldsymbol{X}_{ij}$	The (i,j)th entry of $\boldsymbol{X}$
$\boldsymbol{Z} \in \mathbb{R}^{n\times n}$	The self-expressiveness coefficient matrix
$\boldsymbol{A} \in \mathbb{R}^{n\times n}$	The affinity matrix
$\boldsymbol{P} \in \mathbb{R}^{n\times c}$	The clustering indicator matrix
$\boldsymbol{K}^i \in \mathbb{R}^{n\times n}$	The ith candidate base kernel
$\boldsymbol{K} \in \mathbb{R}^{n\times n}$	The consensus kernel
$\boldsymbol{1}$	The all-one column vector
$\boldsymbol{I}$	The identity matrix
$\|\cdot\|_F$	The Frobenius norm
$\mathrm{Tr}(\cdot)$	The trace operator
$\mathrm{diag}(\cdot)$	Vectorize the diagonal elements of a matrix
$\mathrm{Diag}(\cdot)$	Rearrange a diagonal matrix to a vector
$\lfloor\cdot\rfloor_+$	Give the nonnegative part of a matrix or vector

$$\min_{\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}\|_F^2 + \lambda\mathcal{R}(\boldsymbol{Z}) \tag{1}$$

where $\lambda > 0$ is a trade-off parameter, $\boldsymbol{Z}$ is the self-expressiveness coefficient matrix, and $\mathcal{R}(\boldsymbol{Z})$ is a specific regularization term or constraint term [24], such as sparse constraint [10], low-rank constraint [25], adaptive weighted representation [43], continuous label learning [18], block diagonal representation [29], and structure preserving [46], etc. After obtaining matrix $\boldsymbol{Z}$, the undirected affinity matrix (also known as affinity graph) for spectral clustering is constructed by $\left(|\boldsymbol{Z}|^T + |\boldsymbol{Z}|\right)/2$.

The main challenge of problem (1) is that it can not effectively handle non-linear data. By taking this point into consideration, the non-linear data always can be mapped into RKHS through a kernel function. Then, based upon the aforementioned self-expressiveness property, an affinity matrix in RKHS can be learned by upgrading problem (1) to

$$\begin{aligned} &\min_{\boldsymbol{Z}} \tfrac{1}{2}\|\phi(\boldsymbol{X}) - \phi(\boldsymbol{X})\boldsymbol{Z}\|_F^2 + \lambda\mathcal{R}(\boldsymbol{Z}) \\ =\ &\min_{\boldsymbol{Z}} \mathrm{Tr}\Big(\phi(\boldsymbol{X})^T\phi(\boldsymbol{X}) - 2\phi(\boldsymbol{X})^T\phi(\boldsymbol{X})\boldsymbol{Z} + \boldsymbol{Z}^T\phi(\boldsymbol{X})^T\phi(\boldsymbol{X})\boldsymbol{Z}\Big) + \lambda\mathcal{R}(\boldsymbol{Z}) \\ =\ &\min_{\boldsymbol{Z}} \mathrm{Tr}\Big(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\Big) + \lambda\mathcal{R}(\boldsymbol{Z}) \end{aligned} \tag{2}$$

where ϕ is a specific mapping function and $\boldsymbol{K}$ is the kernel Gram matrix. Note that the kernel trick is used here. Such a model is also called as kernel self-expressiveness learning model. For SKL methods, $\boldsymbol{K}$ is fixed, while for MKL methods, $\boldsymbol{K}$ is an unknown consensus kernel matrix that is desired to be learned from multiple candidate kernels.

2.3. Related MKC methods

Based on self-expressiveness learning model [46], k-means algorithm [14], or graph learning technology [17], many previous MKC methods have made attempts to address the non-linear clustering problem [36,38]. For example, multiple kernel k-means (MKKM) [14] extends the traditional k-means algorithm into a multiple kernel setting for challenging clustering. To enhance robustness of MKKM, an extended version of MKKM, robust multiple kernel k-means (RMKKM) [9], has been proposed to simultaneously find the best clustering indicator label and the optimal combination of the given multiple kernels. Regarding the similarity between each candidate kernel matrix and the resulting affinity matrix, affinity aggregation for spectral clustering (AASC) [13] can be deemed as a MKL version of spectral clustering, which jointly replaces the single affinity matrix with multiple kernel matrices for spectral clustering. For learning a better consensus kernel, spectral clustering with multiple kernels (SCMK) [18] constructs a suitable consensus kernel from a linear combination of multiple candidate kernels. Eventually, SCMK proposes to learn kernel weights, cluster indicator, and similarity simultaneously. By considering the structure character of the consensus kernel, low-rank kernel learning graph based clustering (LKGr) [19] and sparse kernel learning graph based clustering (LKGs) [19] seek an optimal consensus kernel from a linear combination of candidate kernels by imposing a low-rank constraint and a sparse constraint on this consensus kernel matrix, respectively. Later, local structural graph and low-rank consensus multiple kernel learning (LLMKL) [35] suggests that the low-rank kernel used in LKGr is actually a Frobenius norm, and thus LLMKL proposes to use a substitute matrix to replace the consensus kernel matrix, and then imposes a low-rank constraint on this substitute matrix rather than consensus kernel matrix itself. Different from SCMK, LKGr, and LKGs, which use linear weighted strategy to learn a consensus kernel, self-weighted multiple kernel learning (SMKL) [16] introduces a novel self-weighted MKL model for clustering and semi-supervised learning, which learns a suitable consensus kernel based on the assumption that the consensus kernel is a neighbor of all candidate kernels

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

and the important kernel should receive a relatively large weight, and vice versa. By aligning each base kernel with the corresponding local optimal similarity matrix, local kernel alignment maximization (LKAMKC) [22] improves the clustering performance for multiple kernel clustering. Based on LKAMKC, global and local structure alignment model for multiple kernel clustering (GLSAMKC) [40] well considers both the alignment between the global and local structure of data with the same optimal similarity matrix. Neighbor-Kernel MKC [48] proposes an effective neighbor-kernel-based multiple kernels clustering method by considering the intrinsic neighborhood structure among base kernels. Recently, to improve the robustness and the quality of the consensus kernel, joint robust multiple kernel subspace clustering (JMKSC) [46] proposes an effective method for data clustering, which combines the subspace self-expressiveness, a correntropy induced kernel weighting strategy, and the block diagonal regularizer into a unified objective function, as a result, it can efficiently handle non-linear data and defy the impulsive noise and non-Gaussian noise.

However, these existing spectral clustering based MKC methods usually adopt a two-step learning scheme, which have to face the problem of not being able to learn the optimal affinity graph.

3. Simultaneous learning scheme for multiple kernel clustering

We first introduce the formulation of the proposed method and then present an effective algorithm to solve the final objective function.

3.1. Problem formulation

Based on the kernel self-expressiveness learning model and considering $\mathcal{R}(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_F^2$, an affinity graph can be learned in kernel space, i.e.,

$$\min_{\boldsymbol{Z}} \mathrm{Tr}\left(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\right) + \lambda\|\boldsymbol{Z}\|_F^2 \tag{3}$$

where the regularization term, $\|\boldsymbol{Z}\|_F^2$, can avoid the trivial solution (i.e., $\boldsymbol{Z} = \boldsymbol{I}$) and avoid scale change [23], and meanwhile enhance the convexness [29] and grouping effect [30]. Existing MKC methods always use problem (3) to learn a self-expressiveness coefficient matrix $\boldsymbol{Z}$ and require an additional step to construct an affinity matrix. Whereas, these is a drawback (presented in the Section 1) that needs to be defeated.

Now, we aim to bridge the relationship between coefficient matrix and affinity matrix to learn both simultaneously. As we know, the self-expressiveness coefficient matrix $\boldsymbol{Z}$ can be deemed as the substitute of data matrix $\boldsymbol{X}$ because each entry of $\boldsymbol{Z}$ can quantify the similarity between two samples in $\boldsymbol{X}$; that is to say, if two samples are close to each other in original space, the new representations of the two samples in new space must be similar to each other, too. Additionally, the adaptive local structure learning (ALSL) [47,15], $\min_{\boldsymbol{G}} \|\boldsymbol{X}_i - \boldsymbol{X}_j\|_2^2\boldsymbol{G}_{ij}$, is a novel graph learning technology to learn a similarity graph $\boldsymbol{G}$ that can preserve the local manifold structure of data, which is based on the fact that if two samples (e.g., $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$) are close to each other, they will have higher similarity (e.g., $\boldsymbol{G}_{ij}$), and vice versa. Motivated by ALSL, if the similarity between $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$ is smaller, the corresponding affinity value (also known as probability) between them is higher, and vice versa. Consequently, we bridge the relationship between affinity matrix $\boldsymbol{A}$ and coefficient matrix $\boldsymbol{Z}$ by solving the problem defined as follows:

$$\min_{\boldsymbol{A}} \sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{Z}_i - \boldsymbol{Z}_j\|_F^2\boldsymbol{A}_{ij} \tag{4}$$

Moreover, many recent studies [37] have shown that sparse representation is robust to noise and outliers. To improve robustness, we integrate sparse representation into problem (4) and then have

$$\min_{\boldsymbol{A}} \sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{Z}_i - \boldsymbol{Z}_j\|_F^2\boldsymbol{A}_{ij} + \lambda\sum_{i=1}^{n}\|\boldsymbol{A}_i\|_1 \tag{5}$$

where $\|\boldsymbol{A}_i\|_1$ encourages to obtain a sparse $\boldsymbol{A}_i$. Indeed, when $\boldsymbol{A}_i$ is normalized by $\boldsymbol{A}_i^T\boldsymbol{1} = 1$, it exactly forces term $\sum_{i=1}^{n}\|\boldsymbol{A}_i\|_1$ to be constant. That is, the constraint $\boldsymbol{A}_i^T\boldsymbol{1} = 1$ can be deemed as the sparse constraint on $\boldsymbol{A}_i$; meanwhile, the probabilistic property also should be held (i.e., $\boldsymbol{A}_i \succeq 0$ and $\boldsymbol{A}_i^T\boldsymbol{1} = 1$). Then, problem (5) is transformed into

$$\min_{\boldsymbol{A}} \sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{Z}_i - \boldsymbol{Z}_j\|_F^2\boldsymbol{A}_{ij} \quad s.t.\ \forall i,\ \boldsymbol{A}_i^T\boldsymbol{1} = 1,\ \boldsymbol{A}_i \succeq 0 \tag{6}$$

The proof that problem (6) can lead to a sparse matrix $\boldsymbol{A}$ is given in Proposition 1.Proposition 1 Problem (6) can lead to a sparse matrix $\boldsymbol{A}$.

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

ProofDefine that the (i,j)-th entry of $\boldsymbol{Q}$ is denoted as $\boldsymbol{Q}_{ij} = \|\boldsymbol{Z}_i - \boldsymbol{Z}_j\|_F^2$, then problem (6) can be divided into n independent subproblems algebraically, i.e., $\min_{\boldsymbol{A}_i^T\boldsymbol{1}=1,\boldsymbol{A}_i\succeq 0}\|\boldsymbol{Q}_i \odot \boldsymbol{A}_i\|_1$. Then, it is easy to see that each subproblem can be considered as a special case of sparse representation [28]. Obviously, this is to say that problem (6) can lead to a sparse matrix $\boldsymbol{A}$. The proof is completed. ■.

However, problem (6) inevitably has trivial solution. i.e., only one edge of graph $\boldsymbol{A}$ has the affinity value 1, while all the other edges have the affinity value 0. Accordingly, a regularization term, $\|\boldsymbol{A}_i\|_2^2$, is appended to problem (6) to prevent from trivial solution, and then problem (6) can be reformulated as

$$\min_{\boldsymbol{A}} \sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{Z}_i - \boldsymbol{Z}_j\|_F^2\boldsymbol{A}_{ij} + \lambda\sum_{i=1}^{n}\|\boldsymbol{A}_i\|_2^2 \quad s.t. \quad \forall i,\ \boldsymbol{A}_i^T\boldsymbol{1} = 1, \quad \boldsymbol{A}_i \succeq 0 \tag{7}$$

Mathematically, problem (7) can be transformed into

$$\min_{\boldsymbol{A}} \mathrm{Tr}\left(\boldsymbol{Z}\boldsymbol{L}_A\boldsymbol{Z}^T\right) + \lambda\|\boldsymbol{A}\|_F^2,\ s.t.\ \forall i\ \boldsymbol{A}_i^T\boldsymbol{1} = 1, \boldsymbol{A}_i \succeq 0 \tag{8}$$

where $\boldsymbol{L}_A$ is the Laplacian matrix corresponding to $\boldsymbol{A}$, i.e., $\boldsymbol{L}_A = \mathrm{Diag}(\boldsymbol{A}\boldsymbol{1}) - \boldsymbol{A}$. By using problem (8), the affinity graph $\boldsymbol{A}$ is learned, however, this affinity graph may contain too many redundant edges. Subsequently, a top-k probabilistic neighbors selection strategy is introduced to preserve the important edges and remove the unimportant edges so as to obtain a sparse affinity graph and to reduce computing overhead. The detail of such a strategy is given in Section 3.2.

Ideally, it is often more desirable to achieve an affinity graph that consists of exactly c graph connected components (sub-graphs) [33,47], where c is the numbers of clusters. That is, the matrix $\boldsymbol{A}$ is block diagonal matrix with proper permutations associated to c. Obviously, it is difficult to directly obtain the expected affinity graph since the graph produced by problem (8) is always a strongly connected graph. Fortunately, Theorem 1 can tackle this issue.Theorem 1*[29] For any affinity graph $\boldsymbol{A}$, the multiplicity c of the eigenvalue 0 of the corresponding Laplacian matrix $\boldsymbol{L}_A$ equals the number of connected components in $\boldsymbol{A}$.*

Motivated by such theorem, it can be proved that if the Laplacian rank constraint, $\mathrm{rank}(\boldsymbol{L}_A) = n - c$, is employed, the affinity graph $\boldsymbol{A}$ will contain exact c connected components, i.e., the Laplacian rank constraint is very necessary to help to partition the learned affinity matrix into c connected sub-graphs. Here, in order to give more freedom to the affinity matrix $\boldsymbol{A}$, instead of using affinity matrix $\boldsymbol{A}$, we impose the Laplacian rank constraint on the Laplacian matrix $\boldsymbol{L}_Z$ produced by coefficient matrix $\boldsymbol{Z}$, since $\boldsymbol{Z} = \left(\boldsymbol{Z}^T + \boldsymbol{Z}\right)/2$ also can be deemed as an affinity matrix. Mathematically, according to rank theory and Fan's theorem [34], we have

$$\mathrm{rank}(\boldsymbol{L}_Z) = n - c \Rightarrow \min_{\boldsymbol{P}\in\mathbb{R}^{n\times c},\boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I}} \mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right) \tag{9}$$

where $\boldsymbol{P}^T = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ is the cluster indicator matrix.

In addition, SKL is difficult to select a pre-defined kernel function and tuning its kernel parameters [36]. Inspired by the auto-weighted multi-view learning model [32], we introduce a nearest neighbor based auto-weighted MKL model by integrating r candidate base kernels $\left\{\boldsymbol{K}^i\right\}_{i=1}^r$ so as to learn an optimal consensus kernel $\boldsymbol{K}$, i.e.,

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

where w_i indicates the weight of the i-th candidate kernel $\boldsymbol{K}^i$. For ease of exploration, the detailed derivation of $\boldsymbol{w} = [w_1, w_2, \ldots, w_r]$ is presented in Section 3.3.

Regarding the thoughts mentioned above, we formulate a unified objective function to learn an optimal affinity graph for MKC, which is formulated as

$$\begin{aligned}&\min_{\boldsymbol{Z},\boldsymbol{K},\boldsymbol{A},\boldsymbol{P}} \mathrm{Tr}\left(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\right) + \lambda_1\left(\|\boldsymbol{Z}\|_F^2 + \|\boldsymbol{A}\|_F^2\right) + \lambda_2\mathrm{Tr}\left(\boldsymbol{Z}\boldsymbol{L}_A\boldsymbol{Z}^T\right) + \lambda_3\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right) + \lambda_4\sum_{i=1}^{r} w_i\|\boldsymbol{K}^i - \boldsymbol{K}\|_F^2 \\ &s.t.\ \forall i,\ \boldsymbol{A}_i^T\boldsymbol{1} = 1, \boldsymbol{A} \succeq 0, \boldsymbol{P} \in \mathbb{R}^{n\times c}, \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}\end{aligned} \tag{11}$$

where $\lambda_1, \lambda_2, \lambda_3$, and λ_4 are positive trade-off parameters. In fact, only two parameters λ_2 and λ_3 are need to be tuned.

3.2. Optimization

In this section, we solve the variables $\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{P}$, and $\boldsymbol{K}$ in problem (11) based on the alternating direction method of multipliers (ADMM) algorithm [37]. Due to the requirement of ADMM, one auxiliary variable $\boldsymbol{B}$ is first introduced to make the problem (11) separable, then we have

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

$$\min_{\boldsymbol{Z},\boldsymbol{A},\boldsymbol{B},\boldsymbol{P},\boldsymbol{K}} \mathrm{Tr}\left(\boldsymbol{K}-2\boldsymbol{K}\boldsymbol{Z}+\boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\right)+\lambda_1\left(\|\boldsymbol{Z}\|_F^2+\|\boldsymbol{A}\|_F^2\right)+\lambda_2 Tr\left(\boldsymbol{B}\boldsymbol{L}_A\boldsymbol{B}^T\right)+\lambda_3\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right)+\lambda_4\sum_i^r w_i\|\boldsymbol{K}^i-\boldsymbol{K}\|_F^2 \quad (12)$$
$$s.t.\ \forall i,\ \boldsymbol{A}_i^T\boldsymbol{1}=1,\boldsymbol{A}\succeq 0,\boldsymbol{B}=\boldsymbol{Z},\boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I}$$

The augmented Lagrangian function of problem (12) is given by

$$\begin{aligned}&\mathcal{L}\left\{\forall i,\boldsymbol{A}_i^T\boldsymbol{1}=1,\boldsymbol{A}\succeq 0,\boldsymbol{P}\in\mathbb{R}^{n\times c},\boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I}\right\}\\&=\mathrm{Tr}\left(\boldsymbol{K}-2\boldsymbol{K}\boldsymbol{Z}+\boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\right)+\lambda_1\left(\|\boldsymbol{Z}\|_F^2+\|\boldsymbol{A}\|_F^2\right)+\lambda_2\mathrm{Tr}\left(\boldsymbol{B}\boldsymbol{L}_A\boldsymbol{B}^T\right)+\lambda_3\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right)\\&+\lambda_4\sum_i^r w_i\|\boldsymbol{K}^i-\boldsymbol{K}\|_F^2+\frac{\mu}{2}\|\boldsymbol{B}-\boldsymbol{Z}+\frac{\boldsymbol{Y}}{\mu}\|_F^2\end{aligned} \quad (13)$$

where $\boldsymbol{Y}$ and μ are the Lagrangian multiplier matrix and the penalty parameter, respectively.

(1) Update $\boldsymbol{Z}$ as given $\boldsymbol{A},\boldsymbol{B},\boldsymbol{P}$ and $\boldsymbol{K}$:

The subproblem with respect to self-expressiveness coefficient matrix $\boldsymbol{Z}$ can be written as

$$\min_{\boldsymbol{Z}}\mathrm{Tr}\left(\boldsymbol{K}-2\boldsymbol{K}\boldsymbol{Z}+\boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\right)+\lambda_1\|\boldsymbol{Z}\|_F^2+\lambda_3\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right)+\frac{\mu}{2}\|\boldsymbol{B}-\boldsymbol{Z}+\frac{\boldsymbol{Y}}{\mu}\|_F^2 \quad (14)$$

Due to $\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right)=\frac{1}{2}\mathrm{Tr}(\boldsymbol{Z}\boldsymbol{Q})$ (the detailed proof is in Appendix A), where $\boldsymbol{Q}$ is the distance matrix mentioned at Section 3.1, problem (14) has a closed-from solution as follows:

$$\Rightarrow \boldsymbol{Z}^*=(2\boldsymbol{K}+2\lambda_1\boldsymbol{I}+\mu\boldsymbol{I})^{-1}\left(2\boldsymbol{K}+\mu\boldsymbol{B}+\boldsymbol{Y}-\frac{\lambda_3}{2}\boldsymbol{Q}^T\right) \quad (15)$$

(2) **Update $\boldsymbol{B}$ as given $\boldsymbol{Z},\boldsymbol{A},\boldsymbol{P}$ and $\boldsymbol{K}$:**

The updating of auxiliary matrix $\boldsymbol{B}$ is as follows:

$$\min_{\boldsymbol{B}}\lambda_2\mathrm{Tr}\left(\boldsymbol{B}\boldsymbol{L}_A\boldsymbol{B}^T\right)+\frac{\mu}{2}\|\boldsymbol{B}-\boldsymbol{Z}+\frac{\boldsymbol{Y}}{\mu}\|_F^2 \quad (16)$$

whose solution is provided by

$$\Rightarrow \boldsymbol{B}^*=(\mu\boldsymbol{Z}-\boldsymbol{Y})(2\lambda_2\boldsymbol{L}_A+\mu\boldsymbol{I})^{-1} \quad (17)$$

(3) **Update $\boldsymbol{A}$ as given $\boldsymbol{Z},\boldsymbol{B},\boldsymbol{P}$ and $\boldsymbol{K}$:**

The updating of affinity matrix/graph $\boldsymbol{A}$ is as follows:

$$\min_{\boldsymbol{A}}\lambda_2\mathrm{Tr}\left(\boldsymbol{B}\boldsymbol{L}_A\boldsymbol{B}^T\right)+\lambda_1\|\boldsymbol{A}\|_F^2\quad s.t.\ \forall i,\boldsymbol{A}_i^T\boldsymbol{1}=1,\boldsymbol{A}\succeq 0 \quad (18)$$

This problem can be divide into a series of independent subproblems. For $\boldsymbol{A}_i$, the subproblem is formulated as

$$\boldsymbol{A}_i=\arg\min\|\boldsymbol{A}_i+\boldsymbol{D}_i\|_F^2 \quad (19)$$

where $\boldsymbol{D}_{ij}=\frac{\lambda_2\|\boldsymbol{B}_i-\boldsymbol{B}_j\|_F^2}{4\lambda_1}$. The solution of problem (19) is derived by a top-k probabilistic neighbors selection strategy, given by

$$\Rightarrow \boldsymbol{A}_i^*=\left[\frac{1+\sum_{j=1}^k\boldsymbol{D}_{ij}^{\rightarrow}}{k}\boldsymbol{1}-\boldsymbol{D}_i\right]_+ \quad (20)$$

where k is used to control the number of important edges connected to a vertex in affinity graph $\boldsymbol{A}$, and $\boldsymbol{D}_i^{\rightarrow}$ is the $\boldsymbol{D}_i$ with all elements sorted by descending order. For clarity and completeness, the detailed solution steps of (19) are presented in Appendix B. By doing so, from the matrix perspective, the ith column $\boldsymbol{A}_i$ of affinity matrix $\boldsymbol{A}$ has k number of nonzero entries; from the graph perspective, there are k number of important vertices connecting to the i-th vertex in the affinity graph $\boldsymbol{A}$.(4) **Update $\boldsymbol{P}$ as given $\boldsymbol{Z},\boldsymbol{A},\boldsymbol{B}$ and $\boldsymbol{K}$:**

The updating of clustering indicator matrix $\boldsymbol{P}$ is as follows:

$$\min_{\boldsymbol{P}\in\mathbb{R}^{n\times c}}\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right)\quad s.t.\ \boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I} \quad (21)$$

whose solution is formed by the c eigenvectors of $\boldsymbol{L}_Z$ corresponding to its c smallest eigenvalues.(5) **Update $\boldsymbol{K}$ as given $\boldsymbol{Z},\boldsymbol{A},\boldsymbol{B}$ and $\boldsymbol{P}$:**

The subproblem for updating consensus kernel $\boldsymbol{K}$ is as follows:

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

$$\min_{\boldsymbol{K}} \mathrm{Tr}\left(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{Z} + \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\right) + \lambda_4 \sum_i^r w_i \|\boldsymbol{K}^i - \boldsymbol{K}\|_F^2 \tag{22}$$

It should be noted that the weight value w_i is used to control the contribution of the i-th candidate kernel for generating the consensual $\boldsymbol{K}$.

Taking the partial derivative of (22) with respect to $\boldsymbol{K}$ and setting the derivative to zero [5], the closed-form solution of (22) is given by

$$\Rightarrow \boldsymbol{K} = \frac{-\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{Z}^T + 2\boldsymbol{Z}^T + 2\lambda_4 \sum_i w_i \boldsymbol{K}^i}{2\lambda_4 \sum_i w_i} \tag{23}$$

The procedure of solving the problem (11) terminates when $\|\boldsymbol{B} - \boldsymbol{Z}\|_\infty \leqq \epsilon$ with $\epsilon = 1e-5$ or the number of iterators exceeds $mit = 1e3$. The pseudo-code is depicted as in Algorithm 1. By using Algorithm 1, both the affinity matrix $\boldsymbol{A}$ and self-expressiveness coefficient matrix $\boldsymbol{Z}$ can be achieved, and then we fuse the two matrices and calculate the final balanced affinity graph by

$$\boldsymbol{A} = \frac{|\boldsymbol{Z} \odot \boldsymbol{A}| + |(\boldsymbol{Z} \odot \boldsymbol{A})^T|}{2} \tag{24}$$

where $\odot$ denotes the Hadamard element-wise product. After obtaining the undirected graph $\boldsymbol{A}$, we perform spectral clustering [46] onto this fused graph $\boldsymbol{A}$ to obtain final clustering results.

Algorithm 1. Solve the proposed objective model (11) for MKC.

Input: r base kernels $\left\{\boldsymbol{K}^i\right\}_{i=1}^r$, parameters λ_2 and λ_3, and the number of neighbors k.

1: **Initialize for ADMM**: $\rho = 2$, $\mu_{max} = 1e5$, $\epsilon = 1e-5$, $t = 1$, and $mit = 1e3$.

2: **Initialize for model**: $\boldsymbol{Z} = \boldsymbol{I}$, $\boldsymbol{K} = \frac{1}{r}\sum_{i=1}^r \boldsymbol{K}^i$, $\left\{(w_k)^1\right\}_{k=1}^r = \frac{1}{r}$, $\lambda_1 = 1$, and $\lambda_4 = 20$. Other involved variables are initialized as 0.

3: **while** $t++ < mit$

4: Update $(\boldsymbol{Z})^{t+1}$ as (15).

5: Update $(\boldsymbol{B})^{t+1}$ as (17).

6: Update $(\boldsymbol{A})^{t+1}$ as (20).

7: Update $(\boldsymbol{P})^{t+1}$ as (21).

8: Update $(\boldsymbol{K})^{t+1}$ as (23), compute $\boldsymbol{w}$ via Proposition 2.

9: Update $(\boldsymbol{Y})^{t+1}$ using $(\boldsymbol{Y})^t + \mu\left((\boldsymbol{B})^{t+1} - (\boldsymbol{Z})^{t+1}\right)$.

10: Update $(\mu)^{t+1}$ using $\min\left(\mu_{max}, \rho(\mu)^t\right)$.

11: Break if $\|(\boldsymbol{B})^{t+1} - (\boldsymbol{Z})^{t+1}\|_\infty \leqq \epsilon$.

12: **end while**

13: Obtain the optimal $\boldsymbol{Z}$ and $\boldsymbol{A}$, and then fuse them via (24).

14: Perform spectral clustering.

Output: The clustering results: ACC, NMI, and purity.

3.3. Kernel weight strategy

After obtaining the consensus kernel $\boldsymbol{K}$ using (23), the following proposition can determine weight w_i automatically. Proposition 2The weight w_i is determined as $\frac{1}{2\sqrt{\|\boldsymbol{K}-\boldsymbol{K}^i\|_F^2+\xi}}$, where ξ is infinitely close to zero.

ProofDefine the auxiliary problem with respect to $\boldsymbol{K}$ as follows:

$$\min_{\boldsymbol{K}} \sum_{i=1}^r \sqrt{\left\|\boldsymbol{K} - \boldsymbol{K}^i\right\|_F^2} \tag{25}$$

By taking the derivative of (25) with respect to $\boldsymbol{K}$ and setting to be zero, then we have

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

$$\sum_{i=1}^{r}\widehat{w_i}\frac{\partial\left\|\boldsymbol{K}-\boldsymbol{K}^i\right\|_F^2}{\partial\boldsymbol{K}}=0 \tag{26}$$

where $\widehat{w_i}=\frac{1}{2\sqrt{\left\|\boldsymbol{K}-\boldsymbol{K}^i\right\|_F^2}}$. With applying the same operations on problem (10), we can obtain the same result as shown in Eq. (26). Thus, $\widehat{w_i}$ can be considered as w_i, i.e., $w_i=\frac{1}{2\sqrt{\left\|\boldsymbol{K}-\boldsymbol{K}^i\right\|_F^2}}$. To avoid dividing by zero in theory, w_i can be transformed into $w_i=\frac{1}{2\sqrt{\left\|\boldsymbol{K}-\boldsymbol{K}^i\right\|_F^2+\xi}}$, where ξ is infinitely close to zero. The proof is completed. ■

3.4. Complexity analysis

The main computational complexity of Algorithm 1 consists of five parts, i.e., the subproblems for updating $\boldsymbol{Z},\boldsymbol{A},\boldsymbol{B},\boldsymbol{P}$ and $\boldsymbol{K}$. Detailedly, the complexities of updating $\boldsymbol{Z}$ and $\boldsymbol{B}$ are $\mathcal{O}(n^3)$ due to the matrix inverse operation. In fact, since $(\boldsymbol{K}+2\lambda_4\boldsymbol{I})$ and $(2\lambda_2\boldsymbol{L}_A+\mu\boldsymbol{I})$ are positive semi-definite matrices, they also can be solved within $\mathcal{O}(n^2)$ [44]. For updating $\boldsymbol{A}$, it can be solved within $\mathcal{O}(n^2)$. For updating $\boldsymbol{P}$, it requires a SVD operator of $\boldsymbol{L}_Z$ within complexity $\mathcal{O}(n^3)$, but a rank constrained SVD can be effectively solved by some packages like PROPACK[1] within complexity $\mathcal{O}\left(bn^2\right)$, where $b=n-c$. Since the optimization of $\boldsymbol{K}$ takes closed form solution, its computational complexity is $\mathcal{O}(n^2)$. Moreover, the updating of $\boldsymbol{w}$ costs $\mathcal{O}(rn^2)$. In sum, the overall complexity of the proposed algorithm is approximate to $\mathcal{O}(tn^3)$, where t is the iteration number of our algorithm.

4. Experiments

We demonstrate the effectiveness and efficiency of our SLMKC method by conducting several experiments on nine public benchmark datasets. All codes are implemented in Matlab and run on a Mac mini PC with an Intel Core i7 processor at 3.2 GHz, RAM 16-GB, and macOS Mojave operating system.

4.1. Datasets

We employ nine widely used benchmark datasets, including six image datasets (i.e., Yale,[2] Jaffe,[3] ORL,[4] AR,[5] COIL20,[6] and BA[7]) and three text corporas[8] (i.e., TR11, TR41, and TR45). Some samples and the statistics descriptions of these used image datasets are shown in Fig. 2 and Table 2, respectively.

4.2. Compared methods and evaluation criteria

We compare our SLMKC with the following state-of-the-art MKC competitors, including the following: MKKM [14], RMKKM [9], AASC [13], SCMK [18], LKGr [19], JMKSC [46], SMKL [16], and JMKSC [46]. Amongst them, MKKM and RMKKM are k-means [49] based methods, while others are spectral clustering based methods. For fair comparison, the involved parameters of these competitors have been carefully tuned as recommended by their respective authors.

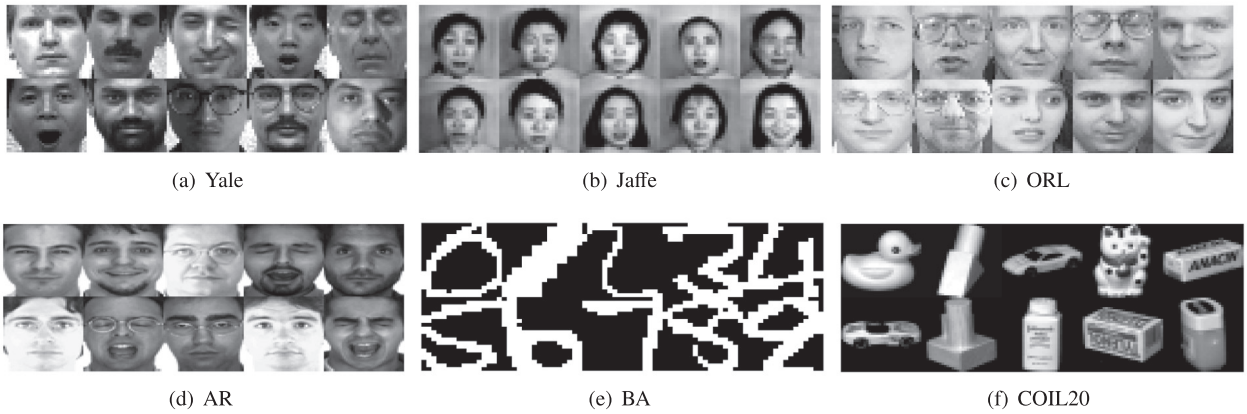
As in [46], to quantitatively evaluate the clustering results of the compared methods, three widely used clustering performance metrics are applied here, i.e., the clustering accuracy (ACC), normalized mutual information (NMI), and purity. For these three metrics used, the higher values indicate the better performance.

4.3. Multiple kernel construction

In the same way as [9,46], we also build 12 candidate base kernels, including seven radial basis function (RBF) kernels, whose (i,j)th entry is defined as $\boldsymbol{K}_{ij}=\exp\left(-\|\boldsymbol{X}_i-\boldsymbol{X}_j\|_2^2/(2\tau\sigma^2)\right)$, where τ varies in the set of $\{0.01,0.05,0.1,1,10,50,100\}$ and σ is the maximum distance between any two data points, four polynomial kernels, whose (i,j)th entry is defined as $\boldsymbol{K}_{ij}=\left(a+\boldsymbol{X}_i^T\boldsymbol{X}_j\right)^b$ with $a=\{0,1\}$ and $b=\{2,4\}$, and one cosine kernel, whose (i,j)th entry is defined as

[1] http://sun.stanford.edu/ rmunk/PROPACK/.
[2] http://www.cvc.Yale.edu/projects/Yalefaces/Yalefaces.html.
[3] http://www.kasrl.org/Jaffe.html.
[4] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedataset.html.
[5] http://www2.ece.ohio-state.edu/ aleix/ARdataset.html.
[6] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.
[7] https://cs.nyu.edu/ roweis/data/binaryalphadigs.mat.
[8] http://trec.nist.gov.

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056



(a) Yale (b) Jaffe (c) ORL

(d) AR (e) BA (f) COIL20

Fig. 2. Sample images of the six image datasets.

Table 2
Statistics of the nine benchmark datasets.

Dataset	# Classes	# Samples	# features	# Note
Yale	15	165	1024	Face image of 15 subjects
Jaffe	10	213	676	Face image of 10 Japanese females
AR	120	840	768	Face image of 70 men and 56 women
ORL	40	400	1024	Face images of 40 subjects
COIL20	20	1440	1024	Object images of 20 different objects
BA	36	1404	320	Digit images of '0' through '9' and 26 letters
TR11	9	414	6429	Text corpora
TR41	10	878	7454	Text corpora
TR45	10	690	8261	Text corpora

$\boldsymbol{K}_{ij} = \left(\boldsymbol{X}_i^T \boldsymbol{X}_j\right) / (\|\boldsymbol{X}_i\| \cdot \|\boldsymbol{X}_j\|)$. Finally, all the kernels are norw malized to $[0, 1]$ using $\boldsymbol{K}_{ij} = \boldsymbol{K}_{ij} / \sqrt{\boldsymbol{K}_{ii}\boldsymbol{K}_{jj}}$. Note that the number of candidate base kernels, r, is equal to 12.

4.4. Performance comparison

The results of all the comparison methods on the nine benchmark datasets are reported in Tables 3–5, and the highest scores are highlighted in boldface, where we repeat each experiment for 20 trials and report the average and standard deviation as final clustering result.

Obviously, the proposed SLMKC method mostly obtains the best performance, followed by JMKSC, SCMK, LKGr, RMKKM, SMKL, MKKM, and AASC. Compared with the second best method, JMKSC, over 4.3%, 3.1%, and 3.2% improvements are achieved in terms of average ACC, NMI and purity, respectively.

To further show the effectiveness of the proposed method, the affinity matrices produced by the comparison methods are also evaluated [6]. Taking the Jaffe dataset, for instance, it consists of ten clusters, and therefore ideally has ten diagonal blocks in affinity matrix. As shown in Fig. 3, the fused affinity matrix $\boldsymbol{A} \odot \boldsymbol{Z}$ produced by our SLMKC has better block diagonal property and inter-cluster separability than that of all other MKC methods.

For better flow of the paper, the detailed discussions are given in Section 4.7.

4.5. Parameter sensitivity

Although there are four trade-off parameters in the final objective function (11), it is very easy to tame these parameters since most of them can be fixed as constants. More specifically, the parameter λ_1 is used to balance the term $\|\boldsymbol{A}\|_F^2$ to resist trivial solution and avoid scale change; the parameter λ_4 is used to control the contribution of the multiple kernel learning term, which can be fixed to $\lambda_1 = 1$ and $\lambda_4 = 20$, respectively, by experience; and the parameters λ_2 and λ_3 are used to control the term $\mathrm{Tr}\left(\boldsymbol{Z}\boldsymbol{L}_A\boldsymbol{Z}^T\right)$ and the term $\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}_Z\boldsymbol{P}\right)$, respectively. Moreover, due to the top-k probabilistic neighbors selection strategy, the parameter k is used to limit the number of neighbors/edges when constructing affinity graph, which is also worthy of evaluating. Therefore, take the Yale and ORL datasets for example, we conduct two experiments successively, the former is used to evaluate the sensitivity of the parameters λ_2 and λ_3, and the latter is used to evaluate the sensitivity of the parameter k.

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

Table 3
Clustering performance comparison (average ± standard deviation) in term of ACC. Symbol 'Avg' indicates the average performance of one evaluated MKC method on all datasets. The highest scores are in bold.

Data	MKKM	RMKKM	AASC	SCMK	LKGr	SMKL	JMKSC	SLMKC
Yale	0.457 ± 0.041	0.521 ± 0.034	0.406 ± 0.027	0.582 ± 0.025	0.540 ± 0.030	0.582 ± 0.017	0.630 ± 0.006	**0.667 ± 0.000**
ORL	0.475 ± 0.023	0.556 ± 0.024	0.272 ± 0.009	0.656 ± 0.015	0.616 ± 0.016	0.573 ± 0.032	0.725 ± 0.014	**0.735 ± 0.008**
Jaffe	0.746 ± 0.069	0.871 ± 0.053	0.304 ± 0.008	0.869 ± 0.022	0.861 ± 0.052	0.967 ± 0.000	0.967 ± 0.007	**1.000 ± 0.000**
AR	0.286 ± 0.014	0.344 ± 0.012	0.332 ± 0.006	0.544 ± 0.024	0.314 ± 0.015	0.263 ± 0.009	0.609 ± 0.007	**0.630 ± 0.016**
BA	0.405 ± 0.019	0.434 ± 0.018	0.271 ± 0.003	0.384 ± 0.014	0.444 ± 0.018	0.246 ± 0.012	0.484 ± 0.015	**0.511 ± 0.012**
COIL-20	0.548 ± 0.058	0.667 ± 0.028	0.349 ± 0.050	0.591 ± 0.028	0.618 ± 0.051	0.487 ± 0.031	0.696 ± 0.016	**0.884 ± 0.003**
TR11	0.501 ± 0.048	0.577 ± 0.094	0.472 ± 0.008	0.549 ± 0.015	0.607 ± 0.043	0.708 ± 0.033	0.737 ± 0.002	**0.741 ± 0.004**
TR41	0.561 ± 0.068	0.627 ± 0.073	0.459 ± 0.001	0.650 ± 0.068	0.595 ± 0.020	0.671 ± 0.002	0.689 ± 0.004	**0.691 ± 0.001**
TR45	0.585 ± 0.066	0.640 ± 0.071	0.526 ± 0.008	0.634 ± 0.058	0.663 ± 0.042	0.671 ± 0.004	0.687 ± 0.036	**0.751 ± 0.000**
Avg	0.507 ± 0.045	0.582 ± 0.045	0.377 ± 0.013	0.607 ± 0.030	0.584 ± 0.032	0.574 ± 0.016	0.692 ± 0.012	**0.734 ± 0.005**

Table 4
Clustering performance comparison (average ± standard deviation) in term of NMI. Symbol 'Avg' indicates the average performance of one evaluated MKC method on all datasets. The highest scores are in bold.

Data	MKKM	RMKKM	AASC	SCMK	LKGr	SMKL	JMKSC	SLMKC
Yale	0.501 ± 0.036	0.556 ± 0.025	0.468 ± 0.028	0.576 ± 0.012	0.566 ± 0.025	0.614 ± 0.015	0.631 ± 0.002	**0.651 ± 0.004**
ORL	0.689 ± 0.016	0.748 ± 0.018	0.438 ± 0.007	0.808 ± 0.008	0.794 ± 0.008	0.733 ± 0.027	0.852 ± 0.012	**0.870 ± 0.004**
Jaffe	0.798 ± 0.058	0.893 ± 0.041	0.272 ± 0.006	0.868 ± 0.021	0.869 ± 0.031	0.951 ± 0.000	0.952 ± 0.010	**1.000 ± 0.000**
AR	0.592 ± 0.014	0.655 ± 0.015	0.651 ± 0.005	0.775 ± 0.009	0.648 ± 0.007	0.568 ± 0.014	0.820 ± 0.002	**0.824 ± 0.002**
BA	0.569 ± 0.008	0.585 ± 0.011	0.423 ± 0.004	0.544 ± 0.012	0.604 ± 0.009	0.486 ± 0.011	0.621 ± 0.007	**0.642 ± 0.009**
COIL-20	0.707 ± 0.033	0.773 ± 0.017	0.419 ± 0.027	0.726 ± 0.011	0.766 ± 0.023	0.628 ± 0.018	0.818 ± 0.007	**0.939 ± 0.002**
TR11	0.446 ± 0.046	0.561 ± 0.118	0.394 ± 0.003	0.371 ± 0.018	0.597 ± 0.031	0.557 ± 0.068	**0.673 ± 0.002**	0.664 ± 0.004
TR41	0.578 ± 0.042	0.635 ± 0.092	0.431 ± 0.000	0.492 ± 0.017	0.604 ± 0.023	0.625 ± 0.004	0.660 ± 0.003	**0.679 ± 0.002**
TR45	0.562 ± 0.056	0.627 ± 0.092	0.420 ± 0.014	0.584 ± 0.051	0.671 ± 0.020	0.622 ± 0.007	0.690 ± 0.022	**0.730 ± 0.000**
Avg	0.605 ± 0.034	0.670 ± 0.048	0.435 ± 0.010	0.638 ± 0.018	0.680 ± 0.020	0.643 ± 0.018	0.746 ± 0.007	**0.778 ± 0.003**

In the first experiment, we fix $k = 10$ and $k = 5$ for the Yale and ORL datasets, respectively, and then use a grid search strategy to tune λ_2 and λ_3. Specifically, the searching regions of λ_2 and λ_3 are both varied from the candidate set $\left\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\right\}$. The results in terms of ACC, NMI and purity are illustrated in Fig. 4. Obviously, the proposed SLMKC method works well for a wide range of the parameters λ_2 and λ_3, and can achieve better performance on both the Yale and ORL datasets when setting $\lambda_2 \in \left[10^{-2}, 10^{-1}\right]$ and $\lambda_3 \in \left[10^{-2}, 10^{-1}\right]$. Therefore, we fix $\lambda_2 = 10^{-2}$ and $\lambda_3 = 10^{-2}$ in all experiments for simplicity.

In the second experiment, we fix $\lambda_2 = 10^{-2}$ and $\lambda_3 = 10^{-2}$, and then tune k from the range of $[1, 2, \ldots, 40]$ to report the influence of different parameter k to the clustering performance in terms of ACC, NMI, and purity. The results are reported in Fig. 5. We can observe that SLMKC is easily tamed and there exists a large parameter space of $(\lambda_2, \lambda_3, k)$. More specifically, for the Yale dataset, the more promising results are achieved when k ranges from the range of $[8, 9, \cdots, 15]$; for the ORL dataset, it is obvious that the clustering performance is not very sensitive to parameter k when setting $k \geqslant 5$, especially the promising performance is achieved when k ranges from the range of $[5, 6, \ldots, 10]$.

In summary, the proposed SLMKC method is very easy to tame, i.e., it works well for a wide range of the parameters

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

Table 5
Clustering performance comparison (average ± standard deviation) in term of Purity. Symbol 'Avg' indicates the average performance of one evaluated MKC method on all datasets. The highest scores are in bold.

Data	MKKM	RMKKM	AASC	SCMK	LKGr	SMKL	JMKSC	SLMKC
Yale	0.475 ± 0.037	0.536 ± 0.031	0.423 ± 0.026	0.610 ± 0.014	0.554 ± 0.029	0.667 ± 0.014	0.673 ± 0.007	**0.706 ± 0.005**
ORL	0.514 ± 0.021	0.602 ± 0.024	0.316 ± 0.007	0.699 ± 0.015	0.658 ± 0.017	0.648 ± 0.017	0.753 ± 0.012	**0.800 ± 0.004**
Jaffe	0.768 ± 0.062	0.889 ± 0.045	0.331 ± 0.008	0.882 ± 0.023	0.859 ± 0.038	0.967 ± 0.000	0.967 ± 0.007	**1.000 ± 0.000**
AR	0.305 ± 0.012	0.368 ± 0.010	0.350 ± 0.006	0.642 ± 0.014	0.330 ± 0.014	0.530 ± 0.014	0.656 ± 0.010	**0.680 ± 0.004**
BA	0.435 ± 0.014	0.463 ± 0.015	0.303 ± 0.004	0.606 ± 0.009	0.479 ± 0.017	0.623 ± 0.011	0.563 ± 0.018	**0.642 ± 0.009**
COIL-20	0.590 ± 0.053	0.699 ± 0.022	0.391 ± 0.044	0.635 ± 0.013	0.650 ± 0.039	0.683 ± 0.004	0.806 ± 0.010	**0.944 ± 0.003**
TR11	0.655 ± 0.044	0.729 ± 0.096	0.547 ± 0.000	0.783 ± 0.011	0.776 ± 0.030	**0.835 ± 0.048**	0.819 ± 0.001	0.752 ± 0.004
TR41	0.728 ± 0.042	0.776 ± 0.065	0.621 ± 0.001	0.758 ± 0.034	0.759 ± 0.031	0.761 ± 0.003	**0.799 ± 0.003**	0.780 ± 0.001
TR45	0.691 ± 0.058	0.752 ± 0.074	0.575 ± 0.011	0.728 ± 0.048	0.800 ± 0.026	0.816 ± 0.004	0.822 ± 0.031	**0.844 ± 0.000**
Avg	0.573 ± 0.038	0.646 ± 0.042	0.429 ± 0.012	0.705 ± 0.020	0.650 ± 0.027	0.726 ± 0.013	0.762 ± 0.011	**0.794 ± 0.003**

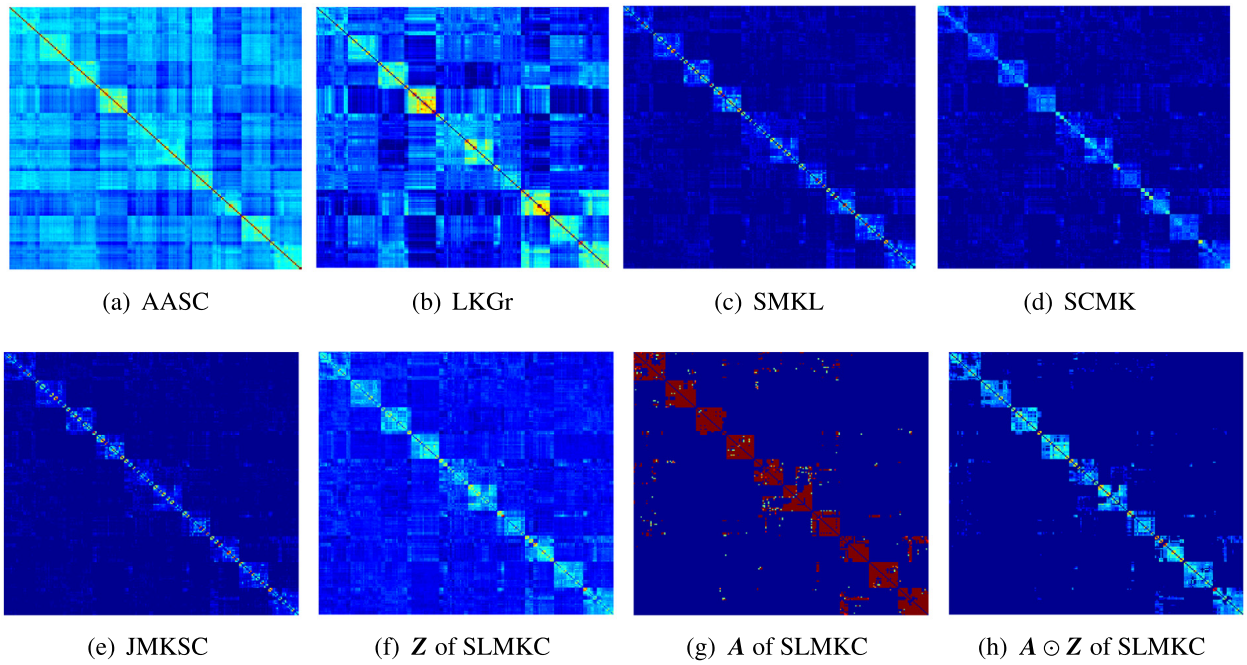


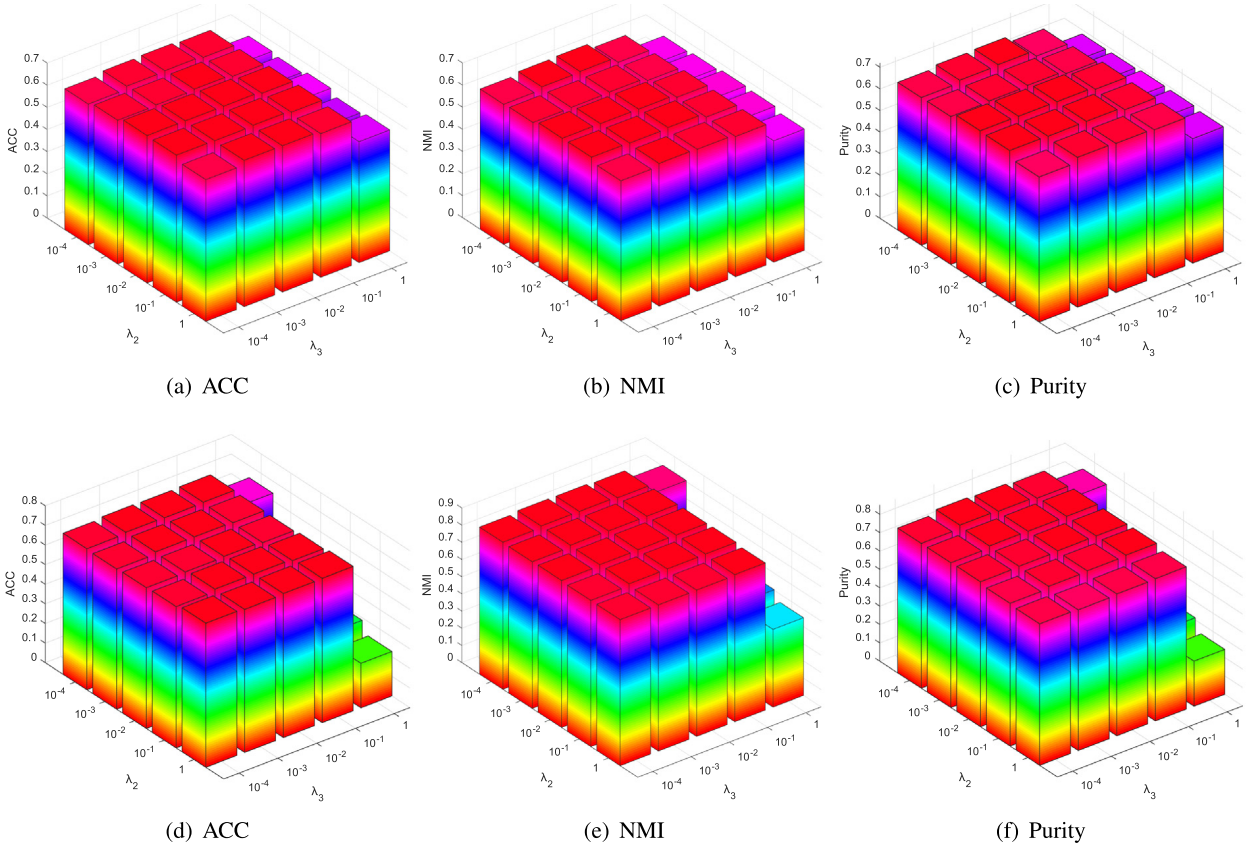
Fig. 3. Illustrations of the resulting affinity matrices produced by the competitors and our SLMKC on the Jaffe dataset. The Jaffe dataset consists of ten clusters. Our SLMKC simultaneously learns the self-expressiveness coefficient matrix $\boldsymbol{Z}$ and the affinity matrix $\boldsymbol{A}$, and then fuses the two matrices to obtain an optimal affinity matrix $\boldsymbol{A} \odot \boldsymbol{Z}$ for clustering purpose. The last sub-figure suggests that SLMKC can yield an intra-class dense and inter-class sparse affinity matrix having better block diagonal property.

$\lambda_1, \lambda_2, \lambda_3$, and λ_4. For simplicity, we fix $\lambda_1 = 1, \lambda_4 = 20$, and $\lambda_2 = \lambda_3 = 10^{-2}$.

4.6. Convergence study and computational time

We further investigate the convergence behavior of the proposed SLMKC method on four representative datasets (i.e., Yale, ORL, TR11, and TR45). Fig. 6 shows the trend of the convergence curve and difference curve with increasing iteration, where difference curve is defined as the different values of $\boldsymbol{Z}$ between two consecutive iterations. As observed, the proposed method converges quickly within about ten iterations and becomes steadily with more iterations; meanwhile, the difference

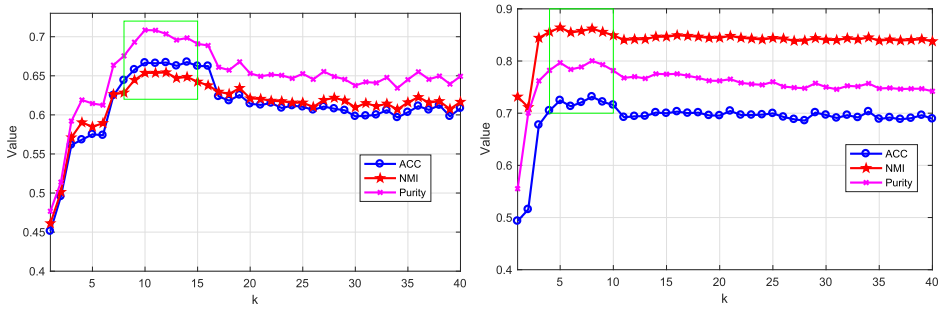
Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056



(a) ACC (b) NMI (c) Purity

(d) ACC (e) NMI (f) Purity

Fig. 4. Clustering performance of the proposed SLMKC method with respect to the two parameters λ_2 and λ_3 on the Yale (the first row) and ORL (the second row) datasets. Specially, the two parameters λ_2 and λ_3 are varied from the candidate set $\left\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\right\}$; the two parameters λ_1 and λ_4 are fixed to $\lambda_1 = 1$ and $\lambda_4 = 20$ for all experiments; the parameter k is fixed to $k = 10$ and $k = 5$ for the Yale and ORL datasets, respectively.



(a) Yale (b) ORL

Fig. 5. Clustering performance of the proposed SLMKC method with respect to the number of neighbors k on the Yale and ORL datasets. We fix $\lambda_1 = 1, \lambda_2 = \lambda_3 = 0.01$, and $\lambda_4 = 20$ for the two datasets.

curves quickly reduce close to zero. This indicates that our SLMKC can converge efficiently. Note that although it is difficult to present a theoretical proof for the algorithm convergence, these curves demonstrate the convergence and effectiveness of our SLMKC empirically.

Furthermore, the computation time is also the key criterion to evaluate a clustering method. Therefore, we compute the computational time (in seconds) of all the MKC competitors on the Yale, ORL, TR11, and TR45 datasets. In order to enhance the comparability among the competitors, we set the same convergence conditions for each algorithm during the experiments. Seen from Fig. 7, the computational cost of our SLMKC is lower than RMKKM, SCMK, SMKL, LKGr, and JMKSC, whose clustering performance is worse than ours. In addition, such as MKKM and AASC, although their computational costs are less than that of the proposed method, their clustering performance is poor.

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

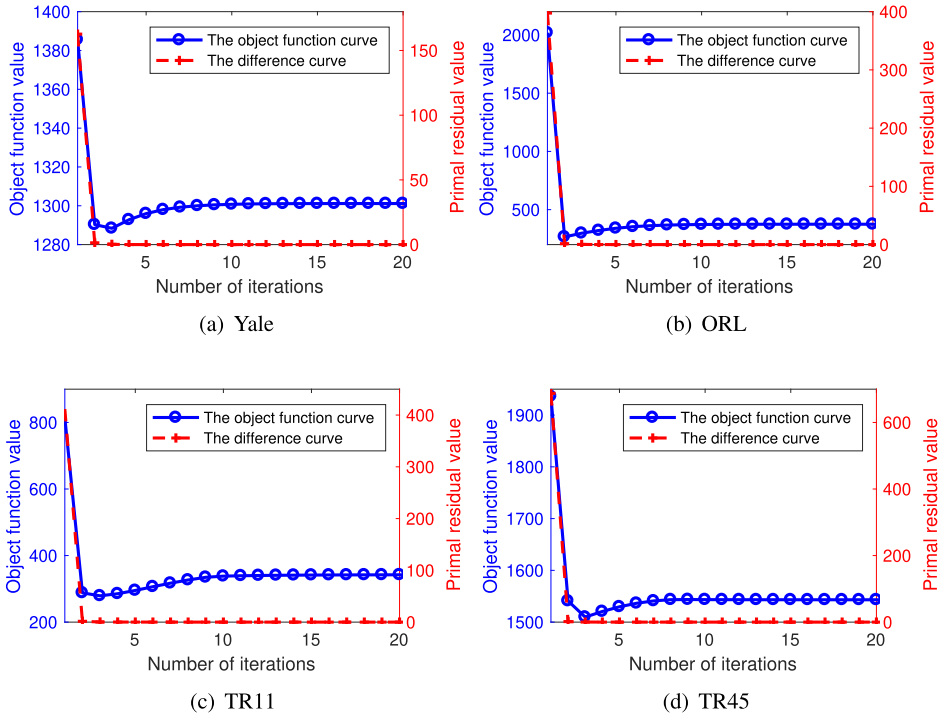


Fig. 6. Convergence curves and difference curves of our SLMKC on the Yale, ORL, TR11, and TR41 datasets. In each subfigure, the x-axis, the y-left-axis, and the y-right-axis denote the iteration number, the corresponding objective function value, and the corresponding primal residual value, respectively.

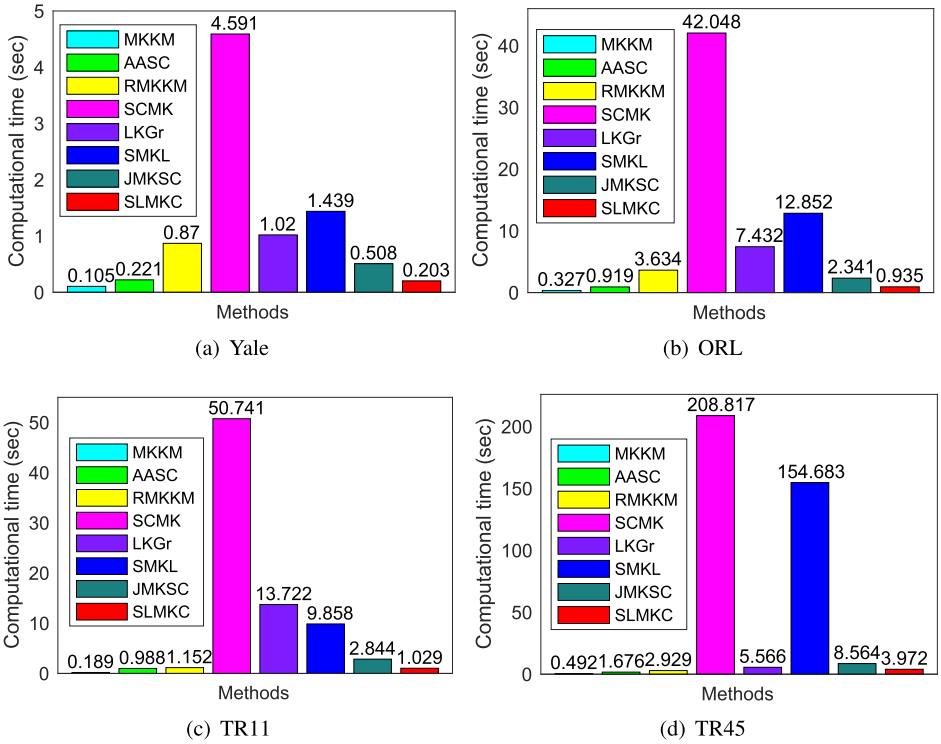


Fig. 7. Computational time (in seconds) of the proposed SLMKC method and the competitors on the Yale, ORL, TR11, and TR41 datasets.

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

Therefore, considering both the convergence and computational cost, the proposed SLMKC method is a fast MKC method.

4.7. Discussion

From these experimental results, one could observe that:(1) According to the experimental results in Tables 3–5, we have

- Overall, in most cases, our SLMKC method consistently achieves the highest clustering performance in terms of ACC, NMI, and purity when compared with some preexisting state-of-the-art MKC methods. This demonstrates that SLMKC can be qualified for clustering tasks.
- In most cases, the k-means based MKC methods (such as MKKM and RMKKM) usually achieve worse clustering performance than these spectral clustering based MKC methods (such as JMKSC, SCMK, SMKL, LKGr, and SLMKC). This suggests that the spectral clustering based methods are generally superior to the standard k-means based ones [16].
- Our SLMKC obtains the smallest standard deviation in most cases, this phenomenon indicates that SLMKC has good statistical coherence. One main concern is that SLMKC has zero standard deviations on the Jaffe and TR45 datasets.
- Compared with these competitors, our SLMKC simultaneously learns the optimal self-expressiveness coefficient matrix (i.e., $\boldsymbol{Z}$) and the affinity matrix (i.e., $\boldsymbol{A}$), but the competitors solely learn the self-expressiveness coefficient matrix rather than both. Especially SMKL, the major difference between SLMKC and SMKL is whether to learn both a coefficient matrix and affinity matrix simultaneously. Obviously, the fused affinity graph $\boldsymbol{A} \odot \boldsymbol{Z}$ proposed in this study is more effective for spectral clustering purpose than these competitors.
- Although we do not evaluate the clustering performance of single kernel clustering (SKC) methods, which are designed based on SKL, [9,46] have confirmed that the performance of SKC methods is largely depended on the choice of kernel function, so MKC methods are generally believed to perform better than SKC methods.

(2) According to the results in Fig. 3, we have

- It is obvious that our SLMKC method can yield an intra-class dense and inter-class sparse fused affinity matrix (i.e., $\boldsymbol{A} \odot \boldsymbol{Z}$) with proper block diagonal structure. On the one hand, the rank constraint of Laplacian matrix $\boldsymbol{L}_Z, rank(\boldsymbol{L}_Z) = n - c$, urges to learn an affinity graph with exactly c connected components. On the other hand, the top-k affinity graph construction strategy can effectively remove the redundant edges. It is noticeable, although the learned self-expressiveness coefficient matrix $\boldsymbol{Z}$ has proper block diagonal structure, it contains too many redundant entries (also knows as edges), i.e., the entries that are not on the diagonal blocks are not zeros (dark blue). Moreover, the learned affinity matrix $\boldsymbol{A}$ also has proper block diagonal structure, but it cannot reflect the local similarity information of between samples, i.e., the diagonal blocks of $\boldsymbol{A}$ only contain flat brown color. Apparently, the fused affinity matrix $\boldsymbol{A} \odot \boldsymbol{Z}$ can preserve the advantages of $\boldsymbol{Z}$ and $\boldsymbol{A}$, as well as eliminating the disadvantages of both.

(3) According to the results in Figs. 4 and 5, we have

- From Fig. 4, we can see that our SLMKC method is insensitive to the involved parameters, thus it is easy to be tamed. For simplicity, we can fix $\lambda_1 = 1, \lambda_2 = \lambda_3 = 10^{-2}$ and $\lambda_4 = 20$, and then tune k in all experiments.
- Additionally, from Fig. 5, it is well understood that the best performance is obtained when using a tunable k rather than a fixed k or a "fully connected" complete graph. In other words, the top-k affinity graph construction strategy can effectively improve clustering performance; meanwhile, it can reduce the computational cost since we only need to update the first k important edges in each iteration.

(4) According to the results in Figs. 6 and 7, we have

- Obviously, the proposed SLMKC method converges very fast almost within ten iterations. The fast convergence property ensures the speed of the whole algorithm.

5. Conclusion

In this paper, a novel SLMKC method is proposed for multiple kernel clustering. Basically, SLMKC can not only effectively handle non-linear data, but also can avoid selecting a pre-defined kernel and tuning its kernel parameters. Significantly, SLMKC completely differs from existing methods, which first learn a self-expressiveness coefficient matrix and then require an additional postprocessing step to construct the final affinity matrix for clustering, SLMKC simultaneously learns the coefficient matrix and affinity matrix so that the learning of coefficient matrix and the learning of affinity matrix can help each other in each iteration. Comprehensive experimental results on nine widely used datasets testify that SLMKC consistently outperforms the state-of-the-art competitors in terms of clustering performance and computational cost.

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Zhenwen Ren: Conceptualization, Methodology, Software, Visualization, Writing - original draft. **Haoyun Lei:** Data curation, Writing - original draft. **Quansen Sun:** Writing - review & editing, Supervision. **Chao Yang:** Supervision, Validation, Investigation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61673220), the Intelligent Manufacturing and Robot Special Project of Major Science and Technology Project in Sichuan (Grant No. 2020ZDZX0014), the Science and Technology Special Project of Major Scientific Instruments and Equipment Project in Sichuan (Grant No. 19ZDZX0119), and the Undergraduate Innovation and Entrepreneurship Project of Sichuan Province of China (Grant No. 19xcy099).

Appendix A. Appendix A

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

$$\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L_Z}\boldsymbol{P}\right) = \frac{1}{2}\mathrm{Tr}(\boldsymbol{ZQ}) \tag{27}$$

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:

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

Thus, Eq. (27) is achieved. The proof is completed.■.

Appendix B. Appendix B

Problem (19) can be simplified to a vector form, i.e.,

$$\min_{\boldsymbol{a}} \|\boldsymbol{a} + \boldsymbol{e}\|_2^2 \quad s.t.\ \boldsymbol{a}^T\boldsymbol{1} = 1, \boldsymbol{a} \geqslant 0 \tag{28}$$

To obtain a better affinity graph and reduce computing overhead, we only update the first k important edges and drop the remaining $n-k$ irrelevant edges, thus a sparse affinity graph can be obtained. Theoretically, the learned $\boldsymbol{a}$ has k non-zero entries and problem (28) can be efficiently solved by

$$\boldsymbol{a} = \left[\frac{1 + \sum_{j=1}^{k} e_j^{\rightarrow}}{k}\boldsymbol{1} - \boldsymbol{e}\right]_+ \tag{29}$$

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

where $\lfloor * \rfloor_+$ indicates the nonnegative part of a vector, and the entries of $\boldsymbol{e}^{\rightarrow}$ are the same as that of $\boldsymbol{e}$ but sorted in ascending order.Proof Considering the constraints $\boldsymbol{a}^T\mathbf{1} = 1$ and $\boldsymbol{a} \geqslant 0$, we then remove the two constraints, thus the Lagrangian function of problem (28) is specified as

$$\mathcal{L}(a, \pi, \psi) = \frac{1}{2}\|\boldsymbol{a} + \boldsymbol{e}\|_F^2 - \pi\left(\boldsymbol{a}^T\mathbf{1} - 1\right) - \psi^T\boldsymbol{a} \tag{30}$$

where π and $\psi \geqslant 0$ are two Lagrangian multipliers. According to KKT condition, we then have

$$\begin{cases} \boldsymbol{a} + \boldsymbol{e} - \pi\mathbf{1} - \psi = 0 \\ \boldsymbol{a}^T\mathbf{1} - 1 = 0 \\ \psi^T\boldsymbol{a} = 0 \end{cases} \tag{31}$$

Note that the equation $\psi^T\boldsymbol{a} = 0$ is held when the condition that if $\boldsymbol{a} \geqslant 0$ then $\psi_j = 0$ is satisfied. Therefore, we obtain

$$\boldsymbol{a} = \lfloor \pi\mathbf{1} - \boldsymbol{e} \rfloor_+ \tag{32}$$

In our top-k probabilistic neighbors selection strategy, the first important k edges are need to be updated, and other edges are removed. Namely, the first k entries of $\boldsymbol{a}$ should be positive, i.e.,

$$\pi - e_k^{\rightarrow} > 0 \text{ and } \pi - e_{k+1}^{\rightarrow} = 0 \tag{33}$$

Based on Eqs. (32) and (33), and the constraint, $\boldsymbol{a}^T\mathbf{1} = 1$, there is

$$\sum_{j=1}^{k}\left(\pi - e_j^{\rightarrow}\right) = 1 \tag{34}$$

Thus, we have

$$\pi = \frac{1 + \sum_{j=1}^{k} e_j^{\rightarrow}}{k} \tag{35}$$

Then, we substitute π in Eq. (32) by $\frac{1+\sum_{j=1}^{k} e_j^{\rightarrow}}{k}$ in Eq. (35), the solution of problem (28) (i.e., Eq. (29)) is obtained. The proof is completed. ■.

References

[1] Ali Akgül, A novel method for a fractional derivative with non-local and non-singular kernel, Chaos Solitons Fract. 114 (2018) 478–482.
[2] Ali Akgül, Reproducing kernel hilbert space method based on reproducing kernel functions for investigating boundary layer flow of a powell–eyring non-newtonian fluid, J. Taibah Univ. Sci. 13 (1) (2019) 858–863.
[3] Ali Akgül, Mahmut Modanli, Crank–nicholson difference method and reproducing kernel function for third order fractional differential equations in the sense of atangana–baleanu caputo derivative, Chaos Solitons Fract. 127 (2019) 10–16.
[4] Esra Karatas Akgül, Solutions of the linear and nonlinear differential equations within the generalized fractional derivatives. Chaos Interdisc. J. Nonlinear Sci. 29 (2) (2019) 023108. .
[5] Dumitru Baleanu, Arran Fernandez, Ali Akgül, On a fractional operator combining proportional and classical differintegrals, Mathematics 8 (3) (2020) 360.
[6] James C. Bezdek, Richard J, Hathaway, Vat: a tool for visual assessment of (cluster) tendency, in: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290), vol. 3, IEEE, 2002, pp. 2225–2230. .
[7] Serhat S. Bucak, Rong Jin, Anil K. Jain, Multiple kernel learning for visual object recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2013) 1354–1369.
[8] Tingquan Deng, Dongsheng Ye, Rong Ma, Hamido Fujita, Lvnan Xiong, Low-rank local tangent space embedding for subspace clustering, Inf. Sci. 508 (2020) 1–21.
[9] Liang Du, Peng Zhou, Lei Shi, Hanmo Wang, Mingyu Fan, Wenjian Wang, Yi-Dong Shen, Robust multiple kernel k-means using l21-norm, in: IJCAI, 2015, pp. 3476–3482. .
[10] Ehsan Elhamifar, Rene Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[11] Reza Ghaemi, Md. Nasir Sulaiman, Hamidah Ibrahim, Norwati Mustapha, et al., A survey: clustering ensembles techniques. World Acad. Sci. Eng. Technol. 50 (2009) 636–645. .
[12] Jeffrey Ho, Ming-Hsuan Yang, Jongwoo Lim, Kuang-Chih Lee, David Kriegman, Clustering appearances of objects under varying illumination conditions, in: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, pp. 11–18.
[13] Hsin-Chien Huang, Yung-Yu Chuang, Chu-Song Chen, Affinity aggregation for spectral clustering, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 773–780. .
[14] Hsin-Chien Huang, Yung-Yu Chuang, Chu-Song Chen, Multiple kernel fuzzy clustering, IEEE Trans. Fuzzy Syst. 20 (1) (2012) 120–134.
[15] Zhao Kang, Xiao Lu, Yiwei Lu, Chong Peng, Wenyu Chen, Zenglin Xu, Structure learning with similarity preserving, Neural Networks page 2020, 10.1016/j.neunet.2020.05.030. .
[16] Zhao Kang, Xiao Lu, Jinfeng Yi, Zenglin Xu, Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification, IJCAI (2018) 2312–2318. .
[17] Zhao Kang, Haiqi Pan, Steven CH Hoi, Zenglin Xu, Robust graph learning from noisy data, IEEE Trans. Cybern. (2019). .

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056

[18] Zhao Kang, Chong Peng, Qiang Cheng, Zenglin Xu, Unified spectral clustering with optimal graph, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 3366–3373.
[19] Zhao Kang, Liangjian Wen, Wenyu Chen, Xu. Zenglin, Low-rank kernel learning for graph-based clustering, Knowl.-Based Syst. 163 (2019) 510–517.
[20] Chun-Guang Li, Rene Vidal, Structured sparse subspace clustering: a unified optimization framework, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 277–286.
[21] Chun-Guang Li, Chong You, René Vidal, Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework, IEEE Trans. Image Process. 26 (6) (2017) 2988–3001.
[22] Miaomiao Li, Xinwang Liu, Lei Wang, Yong Dou, Jianping Yin, En Zhu, Multiple kernel clustering with local kernel alignment maximization, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 1704–1710. .
[23] Yong Li, Jing Liu, Hanqing Lu, Songde Ma, Learning robust face representation with classwise block-diagonal structure, IEEE Trans. Inf. Forensics Secur. 9 (12) (2014) 2051–2062.
[24] Guangcan Liu, Zhouchen Lin, Shuicheng Yan Ju, Yong Yu Sun, Yi Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 171–184.
[25] Guangcan Liu, Zhouchen Lin, Yong Yu, Robust subspace segmentation by low-rank representation, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 663–670. .
[26] Hua Feng Liu, Li Ping Jing, Y.U. Jian, Survey of matrix factorization based recommendation methods by integrating social information, J. Software (2018).
[27] Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, En Zhu, Multiple kernel k-means clustering with matrix-induced regularization, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1888–1894.
[28] Can-Yi Lu, Hai Min, Jie Gui, Lin Zhu, Ying-Ke Lei, Face recognition via weighted sparse representation, J. Visual Commun. Image Represent. 24 (2) (2013) 111–116.
[29] Canyi Lu, Jiashi Feng, Zhouchen Lin, Tao Mei, Shuicheng Yan, Subspace clustering by block diagonal representation, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2) (2019) 487–501.
[30] Canyi Lu, Jinhui Tang, Min Lin, Liang Lin, Shuicheng Yan, Zhouchen Lin, Correntropy induced l2 graph for robust subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1801–1808.
[31] Saeid Niazmardi, Begüm Demir, Lorenzo Bruzzone, Abdolreza Safari, Saeid Homayouni, Multiple kernel learning for remote sensing image classification, IEEE Trans. Geosci. Remote Sens. 56 (3) (2017) 1425–1443.
[32] Feiping Nie, Guohao Cai, Jing Li, Xuelong Li, Auto-weighted multi-view learning for image clustering and semi-supervised classification, IEEE Trans. Image Process. 27 (3) (2017) 1501–1511.
[33] Feiping Nie, Xiaoqian Wang, Heng Huang, Clustering and projected clustering with adaptive neighbors, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 977–986. ACM, 2014.
[34] Feiping Nie, Xiaoqian Wang, Michael I. Jordan, Heng Huang, The constrained laplacian rank algorithm for graph-based clustering, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1969–1976.
[35] Zhenwen Ren, Haoran Li, Chao Yang, Quansen Sun, Multiple kernel subspace clustering with local structural graph and low-rank consensus kernel learning, Knowl.-Based Syst. (2019) 105040. .
[36] Zhenwen Ren, Quansen Sun, Simultaneous global and local graph structure preserving for multiple kernel clustering, IEEE Trans. Neural Networks Learn. Syst. (2020). 10.1109/TNNLS.2020.2991366. .
[37] Zhenwen Ren, Quansen Sun, Bin Wu, Xiaoqian Zhang, Wenzhu Yan, Learning latent low-rank and sparse embedding for robust image feature extraction, IEEE Trans. Image Process. 29 (1) (2019) 2094–2107.
[38] Zhenwen Yang Ren, X. Simon, Quansen Sun, Tao Wang, Consensus affinity graph learning for multiple kernel clustering, IEEE Trans. Cybern. (2020) 10.1109/TCYB.2020.3000947. .
[39] Jingjing Tang, Yingjie Tian, Dalian Liu, Gang Kou, Coupling privileged kernel method for multi-view learning, Inf. Sci. 481 (2019) 110–127.
[40] Chuanli Wang, En Zhu, Xinwang Liu, Long Gao, Jianping Yin, Ning Hu, Multiple kernel clustering with global and local structure alignment, IEEE Access 6 (2018) 77911–77920.
[41] Hao Wang, Yan Yang, Bing Liu, Hamido Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019.
[42] Tinghua Wang, Jie Lu, Guangquan Zhang, Two-stage fuzzy multiple kernel learning based on hilbert–schmidt independence criterion, IEEE Trans. Fuzzy Syst. 26 (6) (2018) 3703–3714.
[43] Jie Wen, Bob Zhang, Yong Xu, Jian Yang, Na. Han, Adaptive weighted nonnegative low-rank representation, Pattern Recogn. 81 (2018) 326–340.
[44] Shijie Xiao, Mingkui Tan, Dong Xu, Weighted block-sparse low rank representation for face clustering in videos, in: European Conference on Computer Vision, Springer, 2014, pp. 123–138.
[45] Zhe Xue, Junping Du, Dawei Du, Siwei Lyu, Deep low-rank subspace ensemble for multi-view clustering, Inf. Sci. 482 (2019) 210–227.
[46] Chao Yang, Zhenwen Ren, Quansen Sun, Mingna Wu, Maowei Yin, Yuan Sun, Joint correntropy metric weighting and block diagonal regularizer for robust multiple kernel subspace clustering, Inf. Sci. 500 (2019) 48–66.
[47] Shuangyan Yi, Yingyi Liang, Zhenyu He, Yi Li, Yiu-Ming Cheung, Dual pursuit for subspace learning, IEEE Trans. Multimedia 21 (6) (2018) 1399–1411.
[48] Sihang Zhou, Xinwang Liu, Miaomiao Li, En Zhu, Li Liu, Changwang Zhang, Jianping Yin, Multiple kernel clustering with neighbor-kernel subspace segmentation. IEEE Trans. Neural Networks Learn. Syst. (2020) 1351–1362 .
[49] Jihua Zhu, Zutao Jiang, Georgios D. Evangelidis, Changqing Zhang, Shanmin Pang, Zhongyu Li, Efficient registration of multi-view point sets by k-means clustering, Inf. Sci. 488 (2019) 205–218. .

Please cite this article as: Z. Ren, H. Lei, Q. Sun et al., Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, Information Sciences, https://doi.org/10.1016/j.ins.2020.08.056