Late Fusion Multiple Kernel Clustering with Local Kernel Alignment Maximization

Tiejian Zhang, Xinwang Liu†, Senior Member, IEEE, Lei Gong, Siwei Wang, Xin Niu, Li Shen

Abstract—Multi-view clustering, which appropriately integrates information from multiple sources to reveal data's inherent structure, is gaining traction in clustering. Though existing procedures have yielded satisfactory results, we observe that they have neglected the inherent local structure in the base kernels. This may cause adverse effects on clustering. To solve the problem, we introduce LF-MKC-LKA, a simple yet effective late fusion multiple kernel clustering with local kernel alignment maximisation approach. In particular, we first determine the nearest k neighbours in the average kernel space for each sample and record the information in the nearest neighbor indicator matrix. Then, the nearest neighbor indicator matrix can be used to generate local structure matrix of each sample. The local kernels of each view may then be generated using the local structure matrix, retaining just the highly confident local similarities for learning the intrinsic global manifold of data. They can also be utilised to keep the block diagonal structure and improve the robustness of the underlying kernels against noise.We input the local kernels of each view into the kernel k-means (KKM) algorithm and get the local base partitions. Finally, we use a three-step iterative optimization approach to maximize the alignment of the consensus partition using base partitions and a regularisation term. As demonstrated, a significant number of trials on 11 multi-kernel benchmark datasets have shown that the proposed LF-MKC-LKA is effective and efficient. A number of experiments are also designed to demonstrate the fast convergence, excellent performance, robustness and low parameter sensitivity of the algorithm. Our code can be find at https://github.com/TiejianZhang/TMM21-LF-MKC-LKA.

Index Terms—Multiple kernel clustering, neighbor, local kernel, local base partition, block diagonal structure

I. Introduction

MUlti-view Clustering (MVC), a technology integrating complementary and consensus multi-view information to enhance clustering efficiency, has gotten increasing attention recently [1]–[7]. And the existing algorithms about it may be split into three groups: i) Multi-view subspace clustering; ii) Co-training style algorithm; iii) Multiple kernel clustering. Further, the multi-view subspace clustering looks for a consistent feature representation across all views [8]–[10]. Moreover, the co-training style algorithm intends to explore the mutual agreement in diverse views by using co-training strategy to get the maximize consensus [11]–[13]. Apparently, multiple kernel clustering uses different predefined kernels corresponding to multiple views and optimizes them for better clustering performance [3]–[7], [14]–[16]. As a simple but effective classical method, k-means has been extended in this way intuitively. And kernel k-means conducts k-means clustering in high-dimensional space to address the linearly non-sparable issue in original space. In [6], in order to diminish the redundancy of the pre-defined kernel, researchers introduce a matrix-induced regularization into multiple kernel k-means clustering. Specially, local kernel alignment has also be found the advantage of improving the clustering performance in multiple kernel clustering situation [3]. In [17], late fusion alignment maximization based on multi-view clustering (MVC-LFA) is proposed. By orthogonal transformation, it maximizes the alignment between the weighted base partition matrices of each view and the consensus partition matrix. But it does not take into account the inherent local structure in the base kernels. Our work falls within the third group.

Although the above methods have worked in enhancing multi-view clustering performance in various way, they fail in the following situations. i) Due to the intensive computational complexity, they are unable to perform large-scale clustering tasks, i.e., usually $\mathcal{O}(mn^3)$ per iteration where m is the number of views and n is the number of samples. ii) In terms of work in [18], the resultant optimization processes of these methods are usually too complex to achieve which raising the risk of over-fitting as well as reducing clustering performance. iii) They may ignore the local structure of the base kernels, which can have anadverse effect on clustering performance.

In this study, we show that a informative local structure of kernels can play an important role in multi-view clustering. We also suggest a new approach named Late Fusion Multiple Kernel Clustering with Local Kernel Alignment Maximization (LF-MKC-LKA). It keeps the competence of the alignment between consensus partition and weighted base partitions and pays attention to the robustness of the base kernels simultaneously, which has a close relation with clustering performance. We generate local kernels for each view with associated sample using the neighbour indicator matrix for each sample in the average kernel space, preserving only the highly confident local similarities for learning the intrinsic global manifold of data. And they can be used to retain the block diagonal structure and enhance the robustness against noise between the base kernels. As the late fusion input for MVC-LFA, the local base partitions can be got with the

T. Zhang, X. Liu, L. Gong, S. Wang, X. Niu and L. Shen are with School of Computer, National University of Defense Technology, Changsha, 410073, China.
†: corresponding author is Xinwang Liu (E-mail: xinwangliu@nudt.edu.cn.)
Manuscript received June 30, 2021; revised November 8, 2021.

local kernels of each view. As demonstrated, the proposed LF-MKC-LKA is proved to be effective and efficient by a vast number of trials on 11 multi-kernel benchmark datasets. Our technique outperforms numerous state-of-the-art multi-view kernel-based clustering algorithms in terms of clustering performance.

The following are the five contributions to this study,

- We find and address the issue that the available late fusion based multi-view clustering algorithms are insufficient in data mining with local structure. The structure of the local kernel we designed includes abundant local information and strengthens the robustness to noise between base kernels. Moreover, being the input of our algorithm, this structure greatly enhances the clustering performance.
- We separate and save the local structure matrix which contains relationship between local kernel structure and original base kernels in the form of operators, and it real reduces the time complexity of generating local kernels greatly.
- For the prior knowledge item, we explore a regularization to integrate the information of local structure, so that the effect of the algorithm is further improved.
- Our algorithm has strong generalization ability because of its robustness, fast convergence, low computational complexity and low parameter sensitivity in text, image, video, biomolecular structure, etc. It is proved to be effective and efficient by a vast number of trials on 11 multi-kernel benchmark datasets.
- Our algorithm and the idea of mining local structure information can be easily applied to other multi-view methods, such as multi-kernel clustering and subspace partitioning.

II. Related Work

A. K-means Clustering

The classic algorithm, k-means, provides an efficient and intuitionistic clustering method. Specifically, [19] proposes the clustering loss function of the typical k-means algorithms,

$$\begin{aligned} &\min_{\mathbf{F}} \ \mathrm{Tr}\left(\mathbf{X}\mathbf{X}^\top\right) - \mathrm{Tr}\left(\mathbf{F}^\top\mathbf{X}\mathbf{X}^\top\mathbf{F}\right) \\ &s.t. \ \ \mathbf{F} \in \mathbb{R}^{n\times k}, \ \mathbf{F}^\top\mathbf{F} = \mathbf{I}_k, \end{aligned} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the data matrix and each row corresponds to a data point with d features. $\mathbf{F} \in \mathbb{R}^{n\times d}$ is a cluster indicator matrix, each row $i(1 \le i \le n)$ indicates the cluster memberships between the point $\mathbf{x}_i$ and others. When the point $\mathbf{x}_i$ belongs to the cluster $\mathbf{C}_j$, then $\mathbf{F}_{ij} = \frac{1}{\sqrt{|\mathbf{C}_j|}}$. The optimal $\mathbf{F}$ for Eq. (1) can be produced by taking the k eigenvectors that correspond to the k largest eigenvalues of $\mathbf{X}\mathbf{X}^\top$.

Because of the single-view data matrix $\mathbf{X}$ in Eq. (1), it can not be used in multi-view tasks directly. Hence, many novel algorithms that can overcome the limitation appear. Researchers expand the k-means into multiple kernel k-means to deal with multi-view problem [3], [5]–[7], [14], [15], [20], [21]. We will introduce it in the next subsection.

B. Multiple Kernel k-means Clustering (MKKM)

Defining $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ as a collection of n data points in k clusters with m views. And $\phi_p(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}_p$ is the feature mapping for the p-th view which can map $\mathbf{x}$ into a reproducing kernel Hilbert space $\mathcal{H}_p(1 \le p \le m)$. Therefore, $\mathbf{x}$ can be transferred to $\phi_{\boldsymbol{\gamma}}(\mathbf{x})^\top = [\gamma_1\phi_1(\mathbf{x}), \cdots, \gamma_m\phi_m(\mathbf{x})]$, where $\boldsymbol{\gamma}^\top = [\gamma_1, \cdots, \gamma_m]$ are the weights of the m kernel functions $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$ respectively. Based on the above, the kernel function can be defined as

$$\kappa_{\boldsymbol{\gamma}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\gamma}}^\top(\mathbf{x}_i)\phi_{\boldsymbol{\gamma}}(\mathbf{x}_j) = \sum\nolimits_{p=1}^m \gamma_p^2\kappa_p(\mathbf{x}_i, \mathbf{x}_j), \tag{2}$$

Hence, the optimization objective of MKKM can be as eq. (3)

$$\begin{aligned} &\min_{\mathbf{F},\boldsymbol{\gamma}} \ \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\right) \\ &s.t. \ \ \mathbf{F} \in \mathbb{R}^{n\times k}, \ \mathbf{F}^\top\mathbf{F} = \mathbf{I}_k, \ \boldsymbol{\gamma}^\top\mathbf{1}_m = 1, \ \gamma_p \ge 0, \end{aligned} \tag{3}$$

where $\mathbf{I}_k \in \mathbb{R}^{n\times k}$ is an identity matrix. The optimal $\mathbf{F}$ for Eq. (3) can be acquired by updating $\mathbf{F}$ and $\boldsymbol{\gamma}$ alternately. The details are revealed as follow,

i) Updating $\mathbf{F}$ by fixed $\boldsymbol{\gamma}$.

$$\begin{aligned} &\max_{\mathbf{F}} \ \mathrm{Tr}\left(\mathbf{F}^\top\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{F}\right) \\ &s.t. \ \ \mathbf{F} \in \mathbb{R}^{n\times k}, \mathbf{F}^\top\mathbf{F} = \mathbf{I}_k, \end{aligned} \tag{4}$$

The optimal $\mathbf{F}$ in Eq. (4) can be obtained by taking the k eigenvectors that correspond to the k largest eigenvalues of $\mathbf{K}_{\boldsymbol{\gamma}}$ [22].

ii) Updating $\boldsymbol{\gamma}$ by fixed $\mathbf{F}$.

$$\begin{aligned} &\min_{\boldsymbol{\gamma}} \ \sum\nolimits_{p=1}^m \gamma_p^2\mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\right) \\ &s.t. \ \ \boldsymbol{\gamma}^\top\mathbf{1}_m = 1, \ \gamma_p \ge 0, \end{aligned} \tag{5}$$

The optimal $\boldsymbol{\gamma}$ for Eq. (5) is obtained by the application of linear constraints to quadratic programming.

Recently, many novel algorithms based on the MKKM have been proposed. Specially, [3] extends local kernel alignment into the MKKM and demonstrates its ability in improving clustering performance.

C. Multi-view Clustering via Late Fusion Alignment Maximization (MVC-LFA)

In [17], a multi-kernel clustering approach based on late fusion alignment maximisation is developed, which optimises the alignment between the consensus partition matrix and the basic partition matrices of linear combination. MVC-LFA can be written at length in the form of the following formula,

$$\begin{aligned} &\max_{\mathbf{F}^*,\{\mathbf{B}_p\}_{p=1}^m,\boldsymbol{\gamma}} \ \mathrm{Tr}\left(\mathbf{F}^{*\top}\mathbf{X}\right) + \mu\mathrm{Tr}\left(\mathbf{F}^{*\top}\mathbf{Q}\right) \\ &s.t. \ \ \mathbf{F}^{*\top}\mathbf{F}^* = \mathbf{I}_k, \ \mathbf{B}_p^\top\mathbf{B}_p = \mathbf{I}_k, \ \sum\nolimits_{p=1}^m \gamma_p^2 = 1, \\ &\gamma_p \ge 0, \ \mathbf{X} = \sum\nolimits_{p=1}^m \gamma_p\mathbf{H}_p\mathbf{B}_p, \end{aligned} \tag{6}$$

where $\{\mathbf{H}_p\}_{p=1}^m$ are the base partitions of each view. $\{\mathbf{B}_p\}_{p=1}^m$ are rotation matrices which solve the problem

of feature misalignment between different views. $\mathbf{Q}$ is the average partition and μ is a trade-off hyper-parameter. The new data partition is presented as $\mathbf{X} = \sum_{p=1}^{m} \gamma_p \mathbf{H}_p \mathbf{B}_p$ which maximizes the alignment with optimal clustering partition. And the regularization of the consensus partition, $\mathrm{Tr}\left(\mathbf{F}^{*\top}\mathbf{Q}\right)$, avoids excessive deviation of $\mathbf{F}^*$ from prior knowledge.

III. Method

A. Construction of local kernels

The effective methods of constructing local structure of data based on kernels in the aforementioned literatures [23]–[26] can be summarized as calculating similarity of paired samples and using them as basis to construct reliable local data structure. There are three reasons why this structure can well mine the local information inside the data.

First, the use of data containing local similarity information between data can improve the ability of the corresponding algorithm to reveal the global data structure, because local geometric patches can extract the global nonlinear high-dimensional structure [27], [28]. Second, as mentioned in [29], the estimation of similarity across relatively long-distance samples may be erroneous due to the ambient geometry in high-dimensional input space being severely folded, twisted, or bent. Third, the interference in the data induced by noise and outliers progressively destroys the underlying manifold, making long-range similarity less reliable. Therefore, it is a realistic and practical learning method in the unsupervised kernel learning scenario to maintain just the high credible local similarity of global manifold data without label distinguishing instruction.

To better illustrate our approach, the symbols we will use are recorded in Table I.The method of constructing local kernel structures in high quality is introduced in detail as follows. Firstly, we search the first τ neighbors of each sample and record their labels. Here, the similarity measure between samples is based on the similarity in the average kernel space. K is the average kernel matrix and $\mathrm{K}_{i,j}$ is the quantitative value of the similarity between sample i and sample j. Each column of K represents the similarity between the corresponding sample and all n samples. It is worth mentioning that this similarity measure method based on the similarity between samples of average kernel is very simple and feasible, because it only requires most of the base kernel matrix rather than all of them to be complementary and contain information. We can get the nearest neighbor indicator matrix $\mathbf{N} \in \mathbb{R}^{\tau \times n}$ by saving the first τ labels of each column of the average kernel.

Subsequently, we construct local structure matrix $\mathbf{R}^{(i)} \in \{0,1\}^{n \times n} (1 \leq i \leq n)$ for each sample based on corresponding column of nearest neighbor indicator matrix $\mathbf{N}_{(:,i)}$. Specifically speaking, all the elements in matrix $\mathbf{R}^{(i)}$ should be 0 initially, and then the elements in $\mathbf{R}^{(i)}\left(\mathbf{N}_{(:,i)}, \mathbf{N}_{(:,i)}\right)$ including elements should be set as 1. $\mathbf{R}^{(i)}$ can be thought of as a mask matrix for recording local structural information, which is apparently positive-definite.

Finally, according to the local structure matrix $\mathbf{R}^{(i)}$ which is got in the last step, we can construct local kernel for i-th sample of each view $\{\mathbf{K}_1^{(i)}, \mathbf{K}_2^{(i)}, \mathbf{K}_3^{(i)}, \ldots, \mathbf{K}_m^{(i)}\}$. The local structure of i-th sample on j-th kernel can be represented as:

$$\mathbf{K}_j^{(i)} = \mathbf{R}^{(i)} \circ \mathbf{K}_j, \tag{7}$$

where $\circ$ is the Hadamard product, which multiply the elements in the same position of two matrixes and obtain new matrix in the same shape. $\mathbf{K}_j^{(i)}$ means the local structure of the i-th composed by the selected $\tau \times \tau$ elements including local nearest neighbor information. Then the local kernel of j-th view can be obtained by $\tilde{\mathbf{K}}_j = \sum_{i=1}^{n} \mathbf{K}_j^{(i)} = \sum_{i=1}^{n} \mathbf{R}^{(i)} \circ \mathbf{K}_j$. Note that $\sum_{i=1}^{n} \mathbf{R}^{(i)}$ is shared between local kernels, so it can be stored independently. Next the local kernels $\{\tilde{\mathbf{K}}_1, \tilde{\mathbf{K}}_2, \tilde{\mathbf{K}}_3, \ldots, \tilde{\mathbf{K}}_m\}$ can be acquired through Hadamard product and can be as the input of the KKM. The results $\{\tilde{\mathbf{H}}_1, \tilde{\mathbf{H}}_2, \tilde{\mathbf{H}}_3, \ldots, \tilde{\mathbf{H}}_m\}$ including rich local structure information are the input of LF-MKC-LKA.

Notations	Meaning
$\mathbf{X} \in \mathbb{R}^{n \times d}$	Data matrix
$\mathbf{K}_i \in \mathbb{R}^{n \times n}$	Kernel of i-th view
$\bar{\mathbf{K}} \in \mathbb{R}^{n \times n}$	Average kernel of all $\mathbf{K}_i$
$\mathbf{K}^* \in \mathbb{R}^{n \times n}$	Local kernel structure of the average kernel
$\mathbf{N} \in \mathbb{R}^{\tau \times n}$	Nearest neighbor indicator matrix
$\mathbf{R}^{(i)} \in \{0,1\}^{n \times n}$	Local structure matrix of sample i
$\mathbf{K}_j^{(i)} \in \mathbb{R}^{n \times n}$	Local structure of i-th sample on j-th kernel
$\tilde{\mathbf{K}}_i \in \mathbb{R}^{n \times n}$	Local kernel of i-th view
$\tilde{\mathbf{H}}_i \in \mathbb{R}^{n \times k}$	Base partitions with local information of i-th view
$\mathbf{B}_p \in \mathbb{R}^{k \times k}$	Rotation matrix of i-th view
$\mathbf{Q} \in \mathbb{R}^{n \times k}$	Regularization term with prior knowledge
$\mathbf{L} \in \mathbb{R}^{n \times k}$	Regularization term including local information
$\mathbf{H} \in \mathbb{R}^{n \times k}$	Consensus partition
$\mathbf{I} \in \mathbb{R}^{n \times k}$	Identity matrix
γ_i	Weight of i-th kernel

Table I: Basic notations for the proposed LF-MKC-LKA

B. Construction prior knowledge with local information

In order to avoid the neglect influence by over-fitting, an effective way is involving prior knowledge in the optimization objective as regularization term [30]–[33]. For mining the information of local structure more pertinently, the local kernel of the average kernel which is a kind of effective prior knowledge is designed in this paper. And the plentiful experiments reflect that the suggested algorithm's robustness and generalization capability are greatly improved compared with the aforementioned algorithms.

The average kernel itself has been proved to be an effective prior knowledge, and the effect has been significantly improved by adding the average kernel as the regularization term in [17], [34]–[36]. The $\bar{\mathbf{K}}$ based regularization term $\mathbf{K}^*$ which is the local kernel structure

of the average kernel is formulated as eq. (8). Taking $\mathbf{K}^*$ as the input of KKM, the base partitions including the local information of average kernel $\mathbf{L}$ can be obtained,

$$\mathbf{K}^* = \sum\nolimits_{i=1}^{n} \bar{\mathbf{K}}^{(i)} = \sum\nolimits_{i=1}^{n} \mathbf{R}^{(i)} \circ \bar{\mathbf{K}}, \tag{8}$$

where $\bar{\mathbf{K}} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{K}_j$.

C. Late Fusion Multiple Kernel Clustering with Local Kernel Alignment Maximization (LF-MKC-LKA)

In the last subsection, we introduce several base partition matrices with local structure information $\{\tilde{\mathbf{H}}_1, \tilde{\mathbf{H}}_2, \tilde{\mathbf{H}}_3, \ldots, \tilde{\mathbf{H}}_m\}$ as well as prior knowledge $\mathbf{L}$. We use maximized alignment to learn the optimal consensus partition. The formulation of LF-MKC-LKA is as follows,

$$\begin{aligned} \max_{\mathbf{H},\{\mathbf{B}_p\}_{p=1}^m,\boldsymbol{\gamma}} \ & \mathrm{Tr}\left(\mathbf{H}^\top \mathbf{X}\right) + \mu \mathrm{Tr}\left(\mathbf{H}^\top \mathbf{L}\right) \\ s.t.\ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k,\ & \mathbf{B}_p^\top \mathbf{B}_p = \mathbf{I}_k,\ \sum\nolimits_{p=1}^{m} \gamma_p^2 = 1, \\ \gamma_p \geq 0,\ & \mathbf{X} = \sum\nolimits_{p=1}^{m} \gamma_p \tilde{\mathbf{H}}_p \mathbf{B}_p, \end{aligned} \tag{9}$$

in which m represents the number of base kernels, k is the number of clusters, and γ_p represents the weight of the p-th base partition. $\{\tilde{\mathbf{H}}_p\}_{p=1}^m \in \mathbb{R}^{n\times k}$ are the base partitions with local information of each view. $\{\mathbf{B}_p\}_{p=1}^m \in \mathbb{R}^{k\times k}$ are rotation matrices, which can unify the permutations with the same clustering results. $\mathbf{L} \in \mathbb{R}^{n\times k}$ is the regularization term including local structure information. $\mathbf{X} = \sum_{p=1}^m \gamma_p \tilde{\mathbf{H}}_p \mathbf{B}_p$ shows that $\mathbf{X}$ is a linear combination of multiple view base partitions with local structure information.

The first part of the formulation, which is reported as $\mathrm{Tr}\left(\mathbf{H}^\top \mathbf{X}\right)$, aims to maximize alignment between the consensus partition $\mathbf{H}$ and the underlying partition $\mathbf{X}$ that incorporates information from multiple views. The second part $\mathrm{Tr}(\mathbf{H}^\top \mathbf{L})$ is a regularization on the consensus partition that prevents $\mathbf{H}$ from deviating too much from knowledge with local structure information. A trade-off coefficient μ is used to combine the two items to obtain an optimal consensus partition $\mathbf{H}$ that incorporates information from multiple views. This framework has been proved to be convergent, and an alternate optimization technique can readily solve it.

D. Optimization Algorithm

A three-step iterative optimization algorithm is used to solve the problem and it can be represented as follow.

i) Optimizing $\mathbf{H}$ by fixed $\{\mathbf{B}_p\}_{p=1}^m$ and $\boldsymbol{\gamma}$. Singular value decomposition (SVD) of matrix $\mathbf{Y}$ can be used to find $\mathbf{H}$ when $\{\mathbf{B}_p\}_{p=1}^m$ and $\boldsymbol{\gamma}$ are fixed. The optimization is reduced to the following form,

$$\max_{\mathbf{H}} \ \mathrm{Tr}\left(\mathbf{H}^\top \mathbf{Y}\right) \ \ s.t.\ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \tag{10}$$

where $\mathbf{Y} = \mathbf{X} + \mu\mathbf{L}$. And the following theorem can give a closed-form solution for Eq. (10).

Theorem 1: Suppose that the matrix $\mathbf{Y}$ in Eq. (10) has the economic rank-k singular value decomposition form as $\mathbf{Y} = \mathbf{J}_k \boldsymbol{\Sigma}_k \mathbf{M}_k^\top$, where $\mathbf{J}_k \in \mathbb{R}^{n\times k}, \boldsymbol{\Sigma}_k \in \mathbb{R}^{k\times k}, \mathbf{M}_k \in \mathbb{R}^{k\times k}$. We can get a closed-form solution of optimization as follows,

$$\mathbf{H} = \mathbf{J}_k \mathbf{M}_k^\top. \tag{11}$$

Proof 1: By taking the the normal singular value decomposition $\mathbf{Y} = \mathbf{J}\boldsymbol{\Sigma}\mathbf{M}^\top$, Eq. (10) can be rewritten as,

$$\mathrm{Tr}(\mathbf{H}^\top \mathbf{J}\boldsymbol{\Sigma}\mathbf{M}^\top) = \mathrm{Tr}(\mathbf{M}^\top \mathbf{H}^\top \mathbf{J}\boldsymbol{\Sigma}). \tag{12}$$

Let $\mathbf{P} = \mathbf{M}^\top \mathbf{H}^\top \mathbf{J}$ and we have $\mathbf{P}\mathbf{P}^\top = \mathbf{M}^\top \mathbf{H}^\top \mathbf{J}\mathbf{J}^\top \mathbf{H}\mathbf{M} = \mathbf{I}_k$. Moreover, we can get $\mathrm{Tr}(\mathbf{M}^\top \mathbf{H}^\top \mathbf{J}\boldsymbol{\Sigma}) = \mathrm{Tr}(\mathbf{P}\boldsymbol{\Sigma}) \leq \sum_{i=1}^k \sigma_i$. Hence, The optimal solution in Eq. (10) is in the form of Eq. (11).

ii) Optimizing $\{\mathbf{B}_p\}_{p=1}^m$ by fixed $\mathbf{H}$ and $\boldsymbol{\gamma}$. When $\mathbf{H}$ and $\boldsymbol{\gamma}$ are fixed, for each rotation matrix $\mathbf{B}_p$, the optimization problem in Eq. (9) can be rewritten as Eq. (13),

$$\max_{\mathbf{B}_p} \ \mathrm{Tr}\left(\mathbf{B}_p^\top \mathbf{A}\right) \ s.t.\ \mathbf{B}_p^\top \mathbf{B}_p = \mathbf{I}_k, \tag{13}$$

where $\mathbf{A} = \gamma_p \mathbf{H}_p^\top \mathbf{H}$. Using the same SVD method for supplied matrix $\mathbf{A}$, the solution of Eq. (13) can be easily obtained. If the matrix $\mathbf{A}$ has the singular value decomposition form as $\mathbf{A} = \mathbf{N}\boldsymbol{\Sigma}\mathbf{G}^\top$, the optimization solution in Eq. (13) can be represented as the closed-form $\mathbf{B} = \mathbf{N}\mathbf{G}^\top$, much like the closed-form in Theorem1. Consequently, we optimize one $\mathbf{B}$ while keeping other $\mathbf{B}_{i\neq p}$ constant at each iteration. With the above method, we can get a set of optimized $\{\mathbf{B}_p\}_{p=1}^m$.

iii) Optimizing $\boldsymbol{\gamma}$ by fixed $\{\mathbf{B}_p\}_{p=1}^m$ and $\mathbf{H}$. When $\{\mathbf{B}_p\}_{p=1}^m$ and $\mathbf{H}$ are fixed, the optimization problem in Eq. (9) is equivalent to the optimization problem as Eq. (14),

$$\max_{\gamma} \ \sum\nolimits_{p=1}^{m} \gamma_p \eta_p \ \ s.t.\ \ \sum\nolimits_{p=1}^{m} \gamma_p^2 = 1,\ \gamma_p^2 \geq 0, \tag{14}$$

where $\eta_p = \mathrm{Tr}(\mathbf{H}^\top \tilde{\mathbf{H}}_p \mathbf{B}_p)$. The closed-form solution of above quadratic programming problem can be described in the form of Eq. (15),

$$\gamma_p = \eta_p / \sqrt{\sum\nolimits_{p=1}^{m} \eta_p^2}. \tag{15}$$

To illustrate the optimization algorithm methodically, we present the technique for Eq. (9) in Algorithm 1, where $\mathrm{obj}^{(t)}$ represents the objective value at the t-th iteration.

E. Algorithm Analysis

1) Convergence Analysis: All in all, the proposed algorithm is described as Algorithm 1, where $\mathrm{obj}^{(t)}$ represents the target value in t-th iterations. And its convergence is proved in following Theorem 2.

Theorem 2: The proof procedure of proposed Algorithm 1 is the same as [17].

Proof 2: Note that $\forall p, q$, $\mathrm{Tr}[(\gamma_p \mathbf{H}_p \mathbf{B}_p)^\top (\gamma_q \mathbf{H}_q \mathbf{B}_q)] \leq \mathrm{Tr}[(\mathbf{H}_p \mathbf{B}_p)^\top (\mathbf{H}_q \mathbf{B}_q)] \leq \frac{1}{2}(\mathrm{Tr}[(\mathbf{H}_p \mathbf{B}_p)^\top (\mathbf{H}_p \mathbf{B}_p)] + \mathrm{Tr}[(\mathbf{H}_q \mathbf{B}_q)^\top (\mathbf{H}_q \mathbf{B}_q)])$. Under the constraints ${\mathbf{H}_p}^\top \mathbf{H}_p = \mathbf{I}_k$ and ${\mathbf{B}_p}^\top \mathbf{B}_p = \mathbf{I}_k$ in which k is the cluster number, we have $\mathrm{Tr}[(\mathbf{H}_p \mathbf{B}_p)^\top (\mathbf{H}_p \mathbf{B}_p)] = \mathrm{Tr}({\mathbf{B}_p}^\top {\mathbf{H}_p}^\top \mathbf{H}_p \mathbf{B}_p) = \mathrm{Tr}(\mathbf{I}_k) = k = \mathrm{Tr}[(\mathbf{H}_q \mathbf{B}_q)^\top (\mathbf{H}_q \mathbf{B}_q)]$. Therefore, we can deduce the

Algorithm 1 The Proposed LF-MKC-LKA

Input: $\{\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_m\}, k, \mu$ and ϵ_0.
Output: $\mathbf{H}$.
Initialize $\{\mathbf{B}_p\}_{p=1}^m = \mathbf{I}_k, \boldsymbol{\gamma} = \frac{\mathbf{1}}{\sqrt{m}}$ and $t = 1$.
Generate $\{\tilde{\mathbf{K}}_1, \tilde{\mathbf{K}}_2, \tilde{\mathbf{K}}_3, \ldots, \tilde{\mathbf{K}}_m\}$ and $\mathbf{K}^*$.
Calculate $\{\tilde{\mathbf{H}}_1, \tilde{\mathbf{H}}_2, \tilde{\mathbf{H}}_3, \ldots, \tilde{\mathbf{H}}_m\}$ and $\mathbf{L}$.
repeat
    Update $\mathbf{H}$ by solving Eq. (10) with fixed $\{\mathbf{B}_p\}_{p=1}^m$ and $\boldsymbol{\gamma}$.
    Update $\{\mathbf{B}_p\}_{p=1}^m$ with fixed $\mathbf{H}$ by Eq. (13).
    Update $\boldsymbol{\gamma}$ by solving Eq. (14) with fixed $\mathbf{H}$ and $\{\mathbf{B}_p\}_{p=1}^m$.
    $t = t + 1$.
until $\left(\text{obj}^{(t-1)} - \text{obj}^{(t)}\right)/\text{obj}^{(t)} \leq \epsilon_0$

optimization objective's upper bound in Eq. (9). We can get $\text{Tr}(\mathbf{H}^\top \mathbf{X}) \leq \frac{1}{2}[\text{Tr}(\mathbf{H}^\top \mathbf{H}) + \text{Tr}(\mathbf{X}^\top \mathbf{X})] = \frac{1}{2}(\text{Tr}(\mathbf{H}^\top \mathbf{H}) + \text{Tr}[\sum_{p,q=1}^m (\gamma_p \mathbf{H}_p \mathbf{B}_p)^\top (\gamma_q \mathbf{H}_q \mathbf{B}_q)]) \leq \frac{k}{2}(m^2 + 1)$. Meanwhile, $\text{Tr}(\mathbf{H}^\top \mathbf{L}) \leq \frac{1}{2}[\text{Tr}(\mathbf{H}^\top \mathbf{H}) + \text{Tr}(\mathbf{L}^\top \mathbf{L})] = k$. Thus, the whole optimization function is upper bounded. All of the three subproblems are convex strictly when optimizing one variable while keeping the others fixed, thus at each iteration the objective of Algorithm 1 is monotonically increased under this condition. Simultaneously, the whole optimization problem is upper-bounded. Consequently, the convergent nature of the suggested method may be validated. The evidence is now fully comprehensive.

2) Computational Complexity Analysis: LF-MKC-LKA is composed of three parts which are generating local kernels and local average kernel, computing $\tilde{\mathbf{H}}_p$ and $\mathbf{L}$, a three-step iterative optimization process. In the first part, the Hadamard product with $\mathbf{R}_i$ and $\mathbf{K}_j$ which are sample i and base kernel j has a complexity cost of $\mathcal{O}(\tau^2)$, so the total complexity cost this part is $\mathcal{O}(mn\tau^2)$. However, it can be reduced to $\mathcal{O}((m+n)\tau^2)$ by storing and sharing $\sum_{i=1}^n \mathbf{R}_i$ when $m \ll n$. In the second part, the complexity cost of kernel k-means is $\mathcal{O}(mkn^2)$. In the third part, First two steps have the computing consumption of $\mathcal{O}(nk^2)$ with singular value decomposition. And the third step is $\mathcal{O}(mk^3)$. Thus, the total expense of part 3 is $\mathcal{O}(t(nk^2 + mk^3))$, where t is iteration times.

This means LF-MKC-LKA maintains a linear increase in complexity as the number of samples increases after storing the results of the first two parts. So it can give consideration to both efficiency and effectiveness when solving large-scale tasks.

3) Storage Complexity Analysis: The proposed method also has advantages in memory consumption. Algorithm 1 describes the proposed method for solving Eq. (9). In Step 3, $\{\mathbf{B}_p\}_{p=1}^m$ costs $O(mk2)$ and γ_p costs $O(m)$ memory. In Step 4, local kernel $\tilde{\mathbf{K}}_j$ costs $O(mn^2)$ memory. In Step 5, $\tilde{\mathbf{H}}_j$ costs $O(mnk)$. From Step 6 to Step 10, variable $\mathbf{H}$ costs $O(nk)$, $\{\mathbf{B}_p\}_{p=1}^m$ costs $O(mk2)$ and γ_p costs $O(m)$ memory. Overall memory consumption is $O(mn^2)$.

F. Extension

Obviously, LF-MKC-LKA and the idea of mining local structure information can be extended to multi-view tasks. First, when designing local kernels, we use average kernel's sample similarity as the benchmark for determining the nearest neighbor of the samples. And then, we calculate and store the sum of the local structure matrix $\mathbf{R}_i$ to generate each $\tilde{\mathbf{K}}_p$ which can be a very efficient method. $\tilde{\mathbf{K}}_p$ preserve only the highly confident local similarities for learning the intrinsic global manifold of data can be used to retain the block diagonal structure and increase the robustness against noise and outliers between the base kernels. In addition, the $\tilde{\mathbf{K}}_p$ and $\mathbf{L}$ generated by classical kernel k-means can be used as an input for late fusion algorithm and prior information with high quality local structure information, which can be used in any late fusion framework.

IV. Experiment

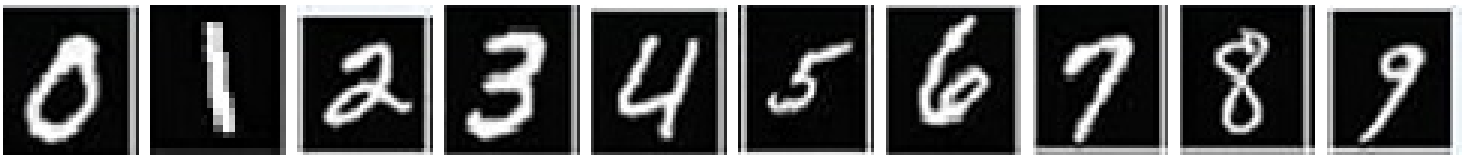
This part aims to evaluate the effectiveness of the proposed LF-MKC-LKA algorithm, especially the local kernel and the base partition with local structure information. We design 5 experiments. In the first experiment, we compare our algorithm with 10 the state-of-the-art MKC algorithms on real world datasets, and prove the effectiveness of our algorithm from ACC, NMI and Purity. Secondly, we compare the base kernel of the dataset with the local kernel involving local structure information to prove that the local kernel has block diagonal structure, and the latter has stronger robustness. Next, we compare kernel k-means, spectral clustering with the base partition generated by the proposed algorithm to prove that it is a good late fusion input including local structure information. In addition, we compare the direct impact of the three different inputs on the performance. And we examine the algorithm's sensitivity to the primary hyper-parameters. By learning the similarity of the consensus partition matrix, we also illustrate the usefulness of the alternative optimization approach. Finally, we verify the convergence of the target value of the algorithm from an experimental point of view.

A. Datasets Overview and Experimental Settings

1) Datasets introduction: The proposed algorithms are tested on twelve commonly used MKL benchmark datasets, which are listed in Table II, including BBCSports[1], ProteinFold and PsortPos[2], Oxford Flower17 and Flower102[3], Plant, Caltech101_mit and Caltech-15[4], YALE Face[5], MFeat[6] and Nonlp[7], UCI_DIGIT[8]. The number of samples, views, categories

[1] http://mlg.ucd.ie/datasets/bbc.html
[2] http://www.raetschlab.org/suppl/protsubloc
[3] http://www.robots.ox.ac.uk/∓vgg/data/flowers/
[4] http://www.vision.caltech.edu/Image_Datasets/
[5] http://www.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html
[6] http://http://mkl.ucsd.edu/dataset/
[7] http://mkl.ucsd.edu/dataset/protein-fold-prediction/
[8] http://ss.sysu.edu.cn/˜py/

(a) Yale



(b) Mfeat

(c) Flower17

(d) Caltech101

Figure 1: Selected example images from the image datasets used in the experiments: (a) YALE, (b) Mfeat, (c) Flower17, and (d) Caltech101.

of these datasets range from 165 to 8189, 2 to 69, and 3 to 102, respectively. All kernel matrices for these datasets are pre-computed using a well constructed similarity function. We present pictures of the image datasets in Figure 1

Dataset	#Samples	#Kernels	#Clusters	#Data Type
BBCSports2	554	2	5	New article
Plant	940	69	4	protein sequence
ProteinFold	694	11	27	protein sequence
PsortPos	541	69	4	protein sequence
Nonpl	2732	69	3	protein sequence
UCI-Digit	2000	3	10	Image
Mfeat	2000	12	10	Image
Flower102	8189	4	102	Image
Caltech101mit	1530	25	102	Image
Flower17	1360	7	17	Image
Caltech101-15	1530	48	102	Image
YALE	165	5	15	Image

Table II: Datasets used in our experiments.

2) Experimental setting: All base kernels are centered and then normalized in our tests. Therefore, for all samples $\mathbf{x}_i$ and p, we can make sure $\mathbf{K}_p(\mathbf{x}_i, \mathbf{x}_i) = 1$ by following [43]. The real number of clusters is expected to be known for all datasets and set as the true number of classes. Our algorithm contains two hyper-parameters, trade-off parameter μ and nearest neighbor number τ. We choose μ from $[2^{-12}, 2^{-10}, \ldots, 2^{12}]$ and τ from $[0.05n, 0.10n, \ldots, 0.95n]$ by grid search, in which n is the number of samples. Clustering accuracy (ACC), normalized mutual information (NMI), and purity are used to assess clustering performance.

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

In this part the proposed method is compared to 10 state-of the-art MKC algorithms to demonstrate its superior performance. All of the compared algorithms' MATLAB implementations are taken from the authors' websites for our tests. The following is a list of the information for the comparing algorithms.

(1) Average multiple kernel k-means (A-MKKM): A-MKKM calculate the average kernel of all kernels and apply a kernel k-means algorithm to it.

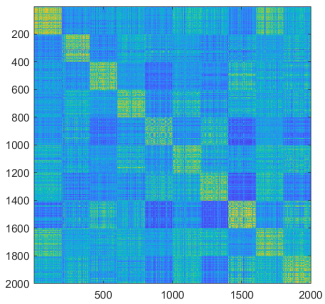
(2) Multiple Kernel k-means (MKKM) [37]: MKKM generally conducts kernel k-means using a linear combination of the base kernels and updates the kernel coefficients by turns.

(3) Multiple kernel k-means with matrix-induced regularization (MKKM-MR) [6]: MKKM-MR can improve kernel diversity and reduce redundancy by using matrix-induced regularization technology.

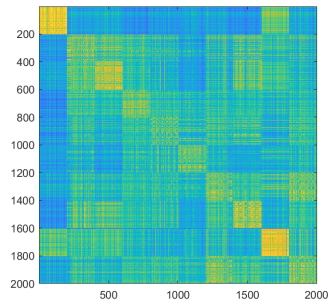
(4) Multiple kernel clustering with local kernel alignment maximization (MKC-LKA) [3]: With maximizing the local kernel alignment, MKC-LKA can keep the inherent local geometric structure of data greatly.

Datasets	A-MKKM	MKKM [37]	RMKKM [38]	MKKM-MR [6]	MKC-LKA [3]	RMSC [39]	MCLES [40]	MVC-LFA [17]	SwMC [41]	SPMKC [42]	Proposed
ACC(%)											
YALE	52.1200	52.1212	56.3636	60.0000	46.6667	58.0303	63.3600	54.5455	46.6667	60.0000	63.0300
Mfeat	71.9500	66.7500	73.7000	92.5500	96.6500	96.6000	95.2500	95.8000	78.6500	15.6000	97.8500
Flower17	51.0300	45.3676	53.3824	58.8235	57.8676	51.1029	56.4600	60.1600	7.0588	32.8676	63.9700
Flower102	27.2900	21.9600	28.1700	39.9100	40.8400	32.9700	-	42.7300	6.7163	5.8100	45.3300
UCI_DIGIT	88.7500	47.0000	44.0000	90.4000	95.1000	80.6500	90.1000	89.1000	78.3000	29.0500	95.5000
Nonpl	49.3800	49.3000	62.7700	56.5900	55.7800	60.6500	-	50.0700	60.2855	39.7145	65.0400
Caltech-15	29.1100	20.3922	24.9020	32.2876	27.3203	25.4902	30.0700	31.1111	16.6667	26.0784	36.1400
Caltech101_mit	35.2900	34.7700	32.0300	37.9100	31.9000	29.6700	35.6863	35.6200	22.4183	35.0980	38.5600
BBCSport2view	66.1800	66.1800	63.7900	66.1800	60.4800	86.0300	77.9412	78.6800	36.0294	36.2132	86.5800
Plant	61.4900	56.3800	55.5300	52.4500	47.3400	53.6200	53.0900	61.7000	38.9362	31.8085	61.6000
ProteinFold	28.1000	27.2300	33.2900	36.3100	33.2700	34.1600	36.4600	35.7300	14.9856	17.8674	38.7400
NMI(%)											
YALE	57.7200	54.1611	59.3247	58.6328	53.5125	57.5848	63.0977	59.8553	48.8594	59.9077	62.7600
Mfeat	69.6800	60.8426	73.0490	85.8956	92.7020	92.6413	90.3600	90.9217	84.5643	2.3800	94.9400
Flower17	50.1900	45.3453	52.5575	57.0543	56.0603	54.3910	55.57	59.7900	2.3783	38.5719	58.3600
Flower102	46.3200	42.3300	48.1700	57.2700	57.6000	53.36	-	57.5900	5.5103	16.5400	59.9800
UCI_DIGIT	80.5900	48.1600	48.0200	83.2200	90.0800	79.4200	83.9524	80.9300	83.8313	17.7552	90.2100
Nonpl	16.5500	14.9400	17.3400	15.5100	11.5300	20.3500	-	16.5400	0.0840	0.0626	20.3300
Caltech-15	53.6600	49.2661	52.0409	58.2464	55.1999	54.5701	50.5300	57.6641	24.3744	53.5544	60.8500
Caltech101_mit	59.9300	59.6400	56.2100	61.4700	58.1900	57.0900	56.5601	60.5000	30.9097	59.7132	62.4200
BBCSport2view	53.9300	53.9300	39.6200	53.9300	45.5700	73.8900	66.1633	58.6300	3.7123	1.6736	71.6300
Plant	26.5700	20.0200	19.3900	21.5600	17.3000	23.1800	23.0600	26.7200	0.5076	0.8555	26.6300
ProteinFold	38.5300	37.1600	40.1700	45.8900	41.2500	43.9100	45.5700	44.5800	7.9099	27.6567	46.1900
Purity(%)											
YALE	53.9400	52.7273	58.1818	60.0000	49.0909	57.2424	64.2400	55.7576	50.9091	60.0000	63.6400
Mfeat	71.9500	66.7500	73.7000	92.5500	96.6500	96.6000	95.2500	95.8000	78.8000	16.1000	97.8500
Flower17	51.9900	46.8382	55.0735	60.5147	59.2647	54.1176	53.2300	62.1320	8.2353	36.8382	64.1900
Flower102	32.2800	27.6100	33.8600	46.3900	48.2100	40.2400	-	49.7300	8.0840	7.4200	52.1900
UCI_DIGIT	88.7500	49.7000	47.2000	90.4000	95.1000	82.9000	90.0100	89.1000	78.6000	31.0500	95.5000
Nonpl	72.1800	71.2700	71.7100	63.9100	61.3800	70.5000	-	72.1800	60.3587	60.3587	73.0200
Caltech-15	31.8100	21.6340	26.2092	34.2484	28.8889	27.1242	32.0300	33.1373	20.3268	27.6471	38.0400
Caltech101_mit	37.5200	37.2500	33.7900	39.7400	34.2500	31.3100	37.9739	37.6500	26.0784	37.6471	41.2400
BBCSport2view	77.2100	77.2100	67.8300	77.2100	70.7700	86.0300	80.6985	79.2300	37.8676	36.5809	86.5800
Plant	61.4900	56.3800	55.5300	58.7200	57.4500	59.4700	58.9400	61.7000	39.3617	39.4681	61.6000
ProteinFold	36.1700	33.8600	37.6100	45.3900	37.9000	42.3600	43.2300	41.7900	18.2997	23.6311	45.5300

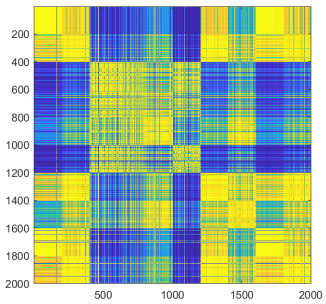
Table III: ACC, NMI and Purity comparison of different clustering algorithms on 11 benchmark datasets.



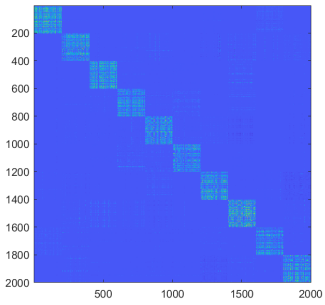
(a) The first original base kernel



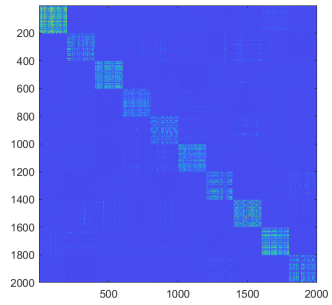
(b) The second original base kernel



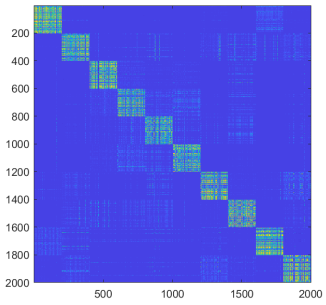
(c) The third original base kernel



(d) The first local kernel



(e) The second local kernel



(f) The third local kernel

Figure 2: Illustration of (a)–(c) original base kernels and (d)–(f) corresponding local kernels on UCI-Digit dataset.

(5) Robust Multi-view Spectral Clustering (RMSC) [39]: RMSC creates a transition probability matrix using information of each view firstly, and then applies a shared low rank transition probability matrix as the input of standard Markov chain for the final clustering task.

(6) Multi-view Clustering in Latent Embedding Space (MCLES) [40]: MCLES can cluster the multi-view data in a learned latent embedding space as well as learning the global structure and the cluster indicator matrix under a unified optimization framework.

(7) Multi-view clustering via late fusion alignment maximization (MVC-LFA) [17]: For more efficient clustering, MVC-LFA aligns the consensus partition with the weighted base partitions maximally.

(8) Robust Multiple Kernel k-means using $l_{2,1}$ (RMKKM) [38]: With adding $l_{2,1}$, RMKKM discovers the optimal combination of multiple kernels, the best clustering label as well as the cluster membership simultaneously.

(9) Self-weighted Multiview Clustering with Multiple Graphs (SwMC) [41]: SwMC provides a method of learning a reasonable weight to each view by exploring a Laplacian rank constrained graph.

(10) Simultaneous Global and Local Graph Structure Preserving for Multiple Kernel Clustering (SPMKC) [42]: SPMKC notes the importance of graph structure of data in kernel space and accomplishs MKC with an efficient way.

H obtained by Algorithm 1 is a continuous variable, in order to get the clustering result, we need to discretize it. The common method is to feed it into a K-mean clustering and receive the ultimate clustering outcome. We use k-means algorithm on the consensus partition **H** to get the final results. Actually, for all methods, each experiment is repeated for 50 times with random initialization in order to diminish the effect of randomness produced by k-means, and calculate the average result. We run all experiments on a desktop computer with a 3.70GHz Intel(R) Core(TM) $i9-10900X$ CPU and 64GB RAM and MATLAB $2018a$ (64bit). The ACC, NMI and Purity under the optimal hyper-parameter of the compared algorithms on the 11 benchmark datasets are displayed on Table III, with the best results in red and the second best results in blue. The following conclusions may be drawn from the findings::

- Firstly, average kernel and single best kernel approaches are strong rivals against other multiple kernel clustering methods, doing well on the majority of the datasets studied. This supports the idea of using the average kernel as the matrix which determines samples neighbors and calculating local kernel structure of the average kernel.
- Compared with the three indexes, LF-MKC-LKA has better performance than other algorithms in almost all datasets, especially in Yale, Flower17, Flower102, Nonpl which is more than 2% better than the suboptimal algorithm. It shows that our method can not only excavate high-quality local neighbor information, but also make full use of it through late fusion. However, some of the proposed algorithm's results are marginally poorer than best result on Yale, Plant, BBCSport2view datasets in Table III. We conjecture that there are two possible reasons for this. The first one is that the proposed method can enhance clustering effect by mining local structure information of data. However, there could be insufficient local information among these datasets. In such case, global method such as RMSC and MVC-LFA do a better job than our methods. The other possible reason is that the way of mining local information may not be effective on all datasets. This motivates us to design more effective approaches to capture the structural information among data, which will be left as a piece of our future work. Please check Section 4.2 in the left column of Page 8 in the revised version for the detail.

C. Evaluation of the Effectiveness of Local Kernels

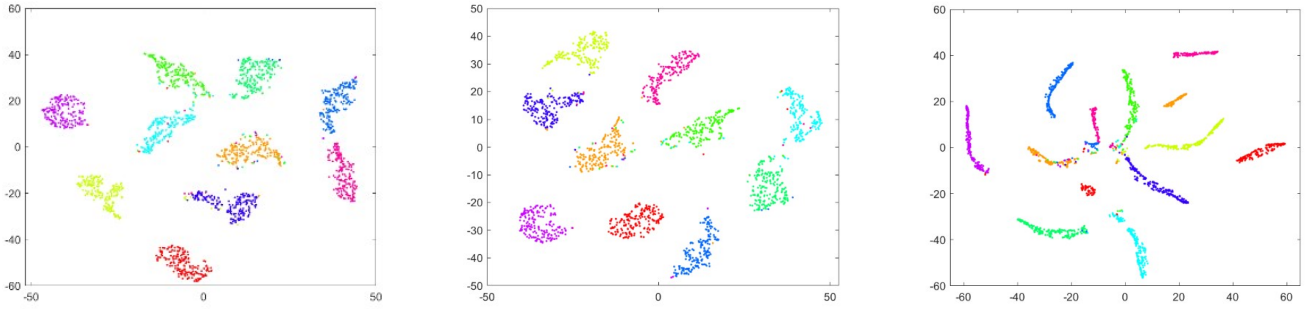
In this section, we will prove that the generated local kernels have good discrimination and robustness due to their block diagonal structure. We take $\tau = 0.05n$ in UCI_DIGIT data to generate local kernels, where τ is the number of nearest neighbors. As shown in Figure 2, compared with the original base kernels, all of the three generated local kernels have more obvious discrimination. Moreover, we can also observe that the robustness of kernels is significantly improved by involving local structure information, especially in Figure 2c and 2f.

Generally speaking, neighbor kernel matrix has better robustness to noise and outliers than conventional kernel matrix, mainly because of its internal weighting mechanism for different samples. By retaining the similarity with higher reliability in the kernel matrix and filtering out the similarity with smaller value and lower reliability, the kernel matrix can better extract the real clustering structure from the data, so as to resist the disturbance of noise to the data distribution.

D. Evaluation of the Effectiveness of base partition and Average Partition with Local Structure Information

In this section, we will prove the effectiveness of base partition and average partition with local structure information from different ways. Given that the spectral clustering clustering is a classical algorithm in utilizing the local structure information of data, we take it as a contrast experiment method of proposed algorithm. In the meantime, base partition and average partition generated by kernel k-means algorithm are also added to the comparison of the former two to prove the better performance and robustness of proposed method. In addition, we use contrast experiments without prior knowledge term for ablation study, and the results give eloquent proof of the effectiveness of the prior knowledge we designed.

We visualize three kinds of average partition with the t-SNE algorithm. From Figure 3, it can be observed that local average partition has more compact clustering partition than the other two, and each cluster has fewer



(a) Average kernel to do kernel k-means (b) Average kernel to do spectral clustering (c) Local average kernel to do kernel k-means

Figure 3: Illustration of three kinds of average partition with t-SNE on UCI-Digit dataset. (a)(b)(c) represent the results of average kernel to do kernel k-means, average kernel to do spectral clustering and local average kernel to do kernel k-means to get average partition.

doped points belonging to other clusters. As the prior knowledge of late fusion algorithm, $\mathbf{L}$ contains local structure information, which is undoubtedly a more close to the real clustering partition and can be used as a higher confidence prior input. We still take the UCI_DIGIT data kernels as an example to show the results of the three kinds of base partitions. We get the following results, as illustrated in Figure 4:

- $\tilde{\mathbf{H}}_p^\top \tilde{\mathbf{H}}_p$ has an obvious block diagonal structure, which has a strong division effect, where $\tilde{\mathbf{H}}_p$ is the base partition with local structure information;
- Figure 4c 4f 4i are the three base partitions $\mathbf{H}_3^1$, $\mathbf{H}_3^2$, and $\tilde{\mathbf{H}}_3$ for the third kernel generated by the three different methods respectively. And $\mathbf{H}_3^1$ and $\mathbf{H}_3^2$ obtained by the first two methods loss the ability of division under the interference of noise, on the contrary, $\tilde{\mathbf{H}}_3$ can suppress the noise well. It shows that kernel k-means is effective for local kernels to get the base partition.

In order to quantify the effectiveness of $\mathbf{L}$ and $\tilde{\mathbf{H}}_p$ as the input of late fusion, we add different Gaussian white noise in experiments. The comparison of the three alternative local partitions by the proposed algorithm, spectral clustering and kernel k-means as the input of late fusion is shown in Figure 5, where SNR is the signal-to-noise ratio, the setting range is 5db to 35db, the higher the signal-to-noise ratio, the smaller the noise. From the curve of change, we can know the following points:

- On all datasets, as the amount of noise increases, , the performance of all comparing algorithms shows a downward trend. However, in contrast, the proposed algorithm can always maintain superior performance, and its robustness to noise is very significant;
- In the low noise range (25-35db), the performance of the proposed algorithm has a significant advantage, which is 4% higher than that of the suboptimal algorithm. However, in the medium noise range (15-20db), the proposed algorithm is superior only in Meaf dataset, but only a little in Yale and BBC-Sport2view. But in the high noise area (below 10db), the performance of the three algorithms declines sharply, but the proposed algorithm can still keep a good result compared with the other two algorithms.

In general, base partition and average partition with local structure information as the input of late fusion has better robustness to noise and outliers. The main reason is that it contains local structure information and suppresses noise information.

We use ablation studies in our experiments to demonstrate the efficiency of $\mathbf{L}$. Deleting the prior knowledge, the optimization objective is as following:

$$\begin{aligned} \max_{\mathbf{H},\{\mathbf{B}_p\}_{p=1}^m,\boldsymbol{\gamma}} \ & \operatorname{Tr}\left(\mathbf{H}^\top \mathbf{X}\right) \\ s.t. \ & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \ \mathbf{B}_p^\top \mathbf{B}_p = \mathbf{I}_k, \ \sum\nolimits_{p=1}^m \gamma_p^2 = 1, \\ & \gamma_p \geq 0, \ \mathbf{X} = \sum\nolimits_{p=1}^m \gamma_p \tilde{\mathbf{H}}_p \mathbf{B}_p. \end{aligned} \tag{16}$$

We record the results of the ablation study in Table IV. From Table IV we can obtain two points: i) The ablation experiment without prior knowledge with $\mathbf{L}$ still has excellent performance, which also proves that our local nearest neighbor information is sufficient and effective, ii) Compared with the other algorithms, the proposed algorithm has obvious performance advantages in every dataset. And the reason is that the local kernel structure of the average kernel itself contains complementary information from all views and local neighbor information between samples. It is used as a regularization term to make the consensus matrix $\mathbf{H}$ a trade-off between $\mathbf{X}$ and $\mathbf{L}$, so that the final learned $\mathbf{H}$ will not deviate from the target value excessively and be more robust.

To further verify the validity of the algorithm, we supplemented the comparison experiments using different number of kernels on multiple datasets. As the number of kernels grows, the performance of the algorithm shows an overall upward trend with the increase of the number of kernels, which is not monotonic, as shown in Figure 6. As the number of kernels involved in clustering increases, the information from different views complements each other

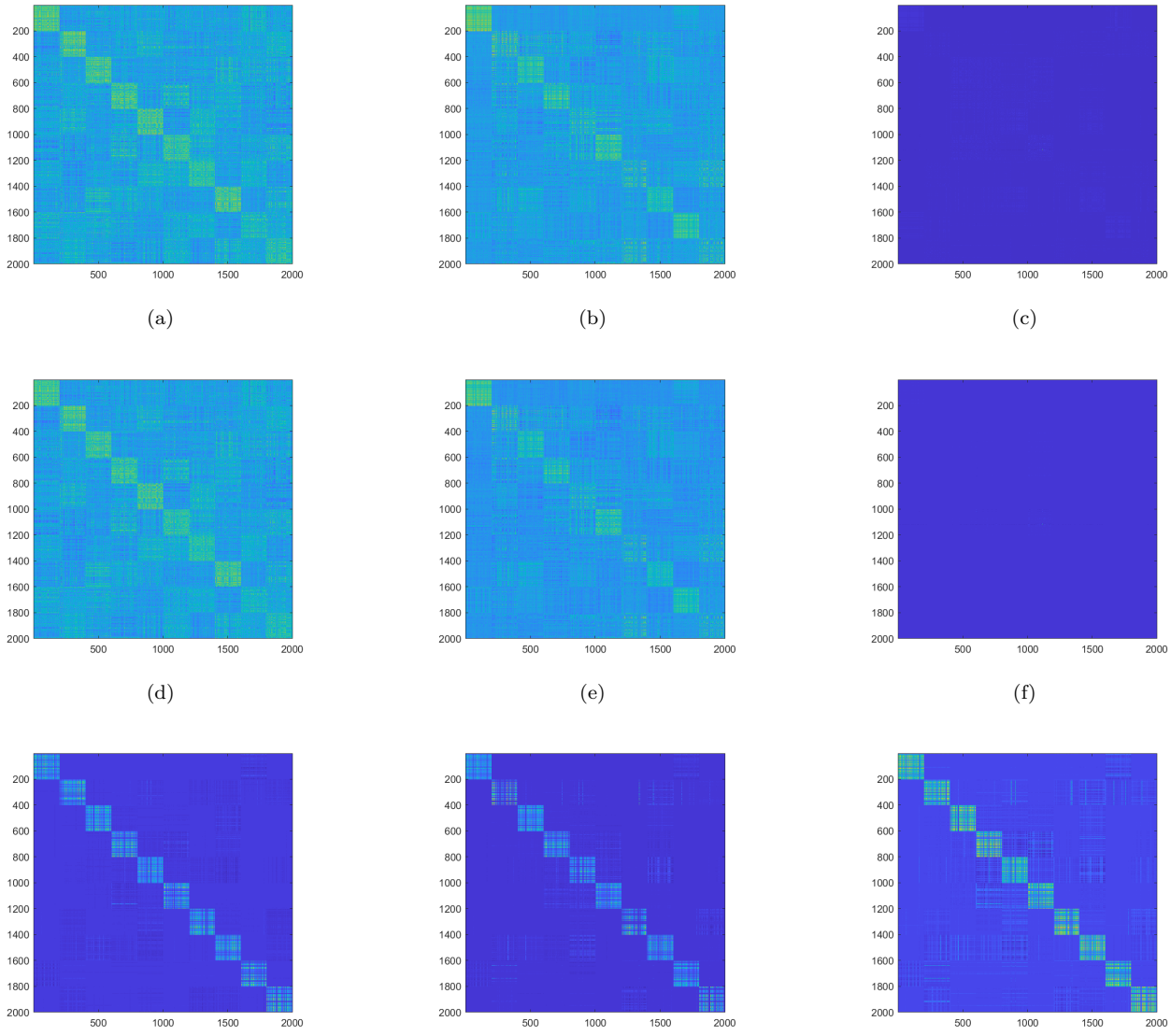


Figure 4: Illustration of (a)–(c) base partition generated by kernel k-means, (d)–(f) base partition generated by spectral clustering and (g)–(i) base partition with Local Structure Information on UCI-Digit dataset.

Datasets	ACC(%)		NMI(%)		Purity(%)	
	Propose	Ablation	Propose	Ablation	Propose	Ablation
YALE	63.0300	61.8182	62.7600	61.6961	63.6400	62.4242
Flower17	63.9700	61.1765	58.3600	56.9113	64.1900	61.7647
Flower102	45.3300	44.1690	59.9800	59.4710	52.1900	50.9708
Nonpl	65.0400	62.0791	20.3300	17.7717	73.0200	70.8638
Caltech-15	36.1400	34.5752	60.8500	60.0088	38.0400	36.2745
Caltech101_mit	38.5600	37.1242	62.4200	61.6637	41.2400	38.9542
BBCSport2view	86.5800	84.1912	71.6300	66.7503	86.5800	84.1912
Plant	61.6000	55.6383	26.6300	18.17	61.6000	55.6383
ProteinFold	38.7400	38.0403	46.1900	45.8632	45.5300	44.2363
PsortPos	59.7000	59.3346	29.9200	29.0990	63.0300	63.0314

Table IV: ACC, NMI and Purity comparison of ablation experiment.

and does benefit to clustering. However, not all kernels are helpful for clustering tasks. When some 'bad' kernels are added, they contain redundant and noisy information which can be harmful. Therefore, it is meaningful to learn the weights of kernels to fully utilize the complementary information of the kernels and reduce the redundant information.

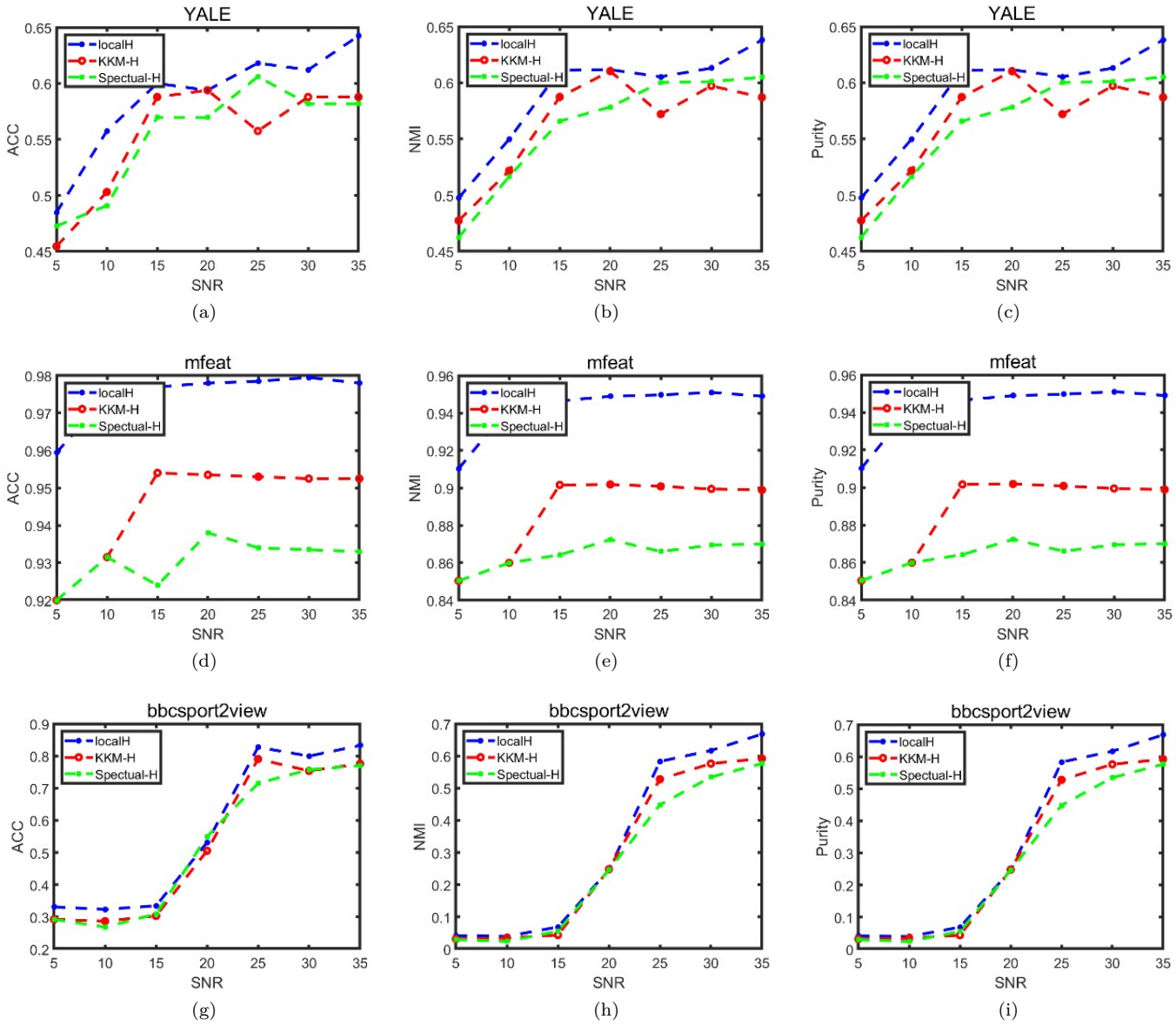


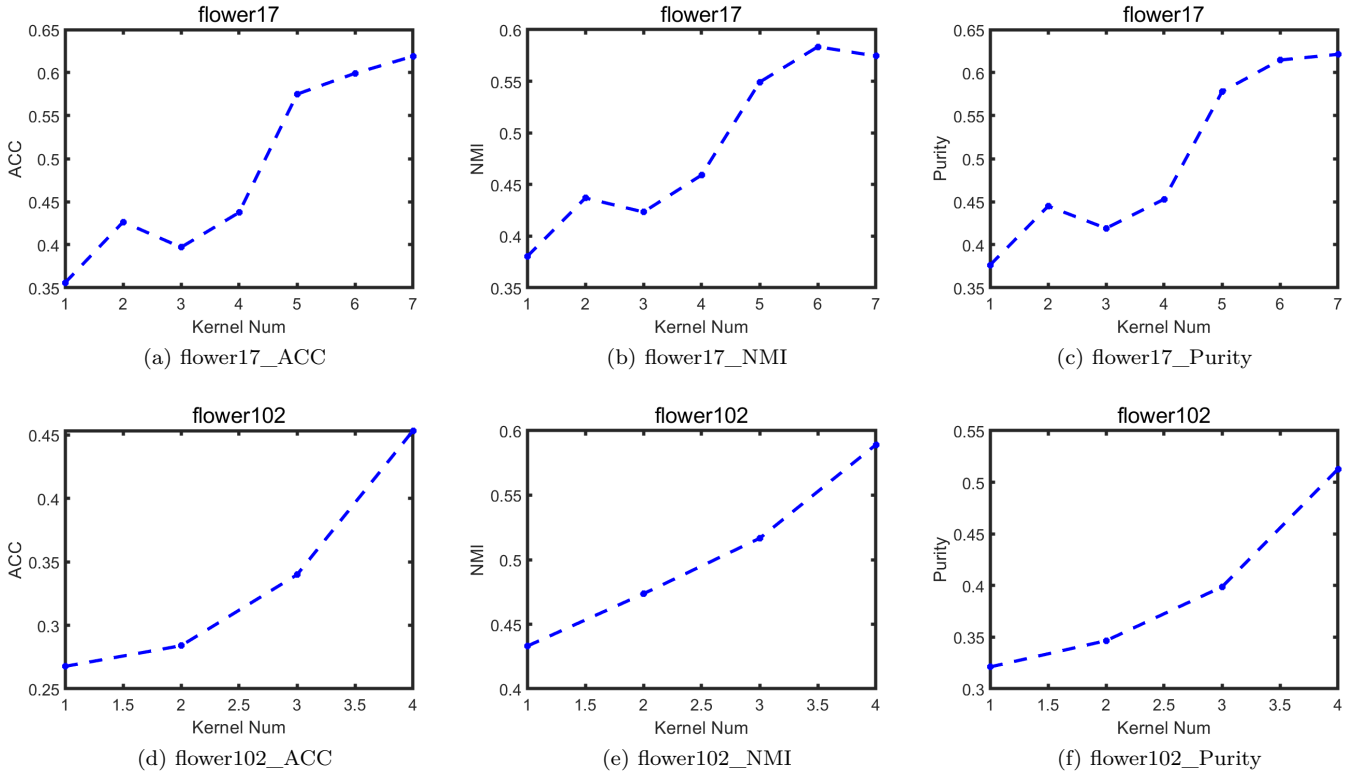
Figure 5: Comparison of clustering performance of algorithms on three datasets under three different late fusion inputs at different levels of noise. In the picture, blue, red, green dotted lines represent the performance of $\tilde{\mathbf{H}}$, $\mathbf{H}$ generated by kernel k-means and $\mathbf{H}$ generated by spectral clustering.

E. Parameter Sensitivity Analysis

Notice that LF-MKC-LKA introduces two hyper-parameters, i.e., the trade-off coefficient μ and the neighborhood number τ which denotes τ-nearest neighbors for the sample. In order to assess algorithm's sensitivity under the two settings, we use the rasterization method to extract the paired μ and τ and get the ACC. The range for μ is $[2^{-12}, 2^{-10}, \ldots, 2^{12}]$ and τ is $[0.05n, 0.15n, \ldots, 0.75n]$, n is the number of samples. Figure 5 shows the sensitivity experimental results on ProteinFold, PsortPos, Flower102, Mfeat, Plant and YALE.

The following can be obtained from Figure 7,

- Both the two hyper-parameters have a great influence on the experimental results,
- The proposed algorithm can maintain excellent performance within a fairly wide range of parameter setting,
- The accuracy of the algorithm has been maintained at a high level, although there is a slight change with the increase of μ. In a wide range, LF-MKC-LKA has stable performance,
- The proposed algorithm is relatively sensitive to the τ, the number of nearest neighbors, reflecting data's underlying structure and hyper-parameters' choice. Nevertheless, it outperforms the suboptimal algorithm on most benchmark datasets.



(a) flower17_ACC (b) flower17_NMI (c) flower17_Purity

(d) flower102_ACC (e) flower102_NMI (f) flower102_Purity

Figure 6: The clustering performance changes with the number of kernels on flower17 and flower102 datasets.

F. Effectiveness Analysis of Optimization Algorithm

In our method, $\mathbf{H}^\top\mathbf{H}$ with the variation of iterations can represent the similarity change of the learning sample and prove the effectiveness of the algorithm. We record the $\mathbf{H}$ of UCI_DIGIT after each iteration, and visualize $\mathbf{H}^\top\mathbf{H}$ under different iteration times in Figure 8. It can be seen that when the number of iterations grows, samples' similarity in cluster becomes higher and the similarity of the samples between clusters becomes lower, and the block diagonal structure of the similarity graph becomes more obvious.

G. Convergence of the Proposed Algorithm

Our algorithms can converge to a local minimum theoretically according to [44]. In order to verify the algorithm more intuitively, we make the convergence graph of the target value of the algorithm and the number of iterations. In Figure 9, we can see that the target values of the algorithm on six datasets increase monotonically as the number of iterations increases, and generally converge within 15 times. The experimental results strongly support our algorithm's convergence.

V. Conclusion

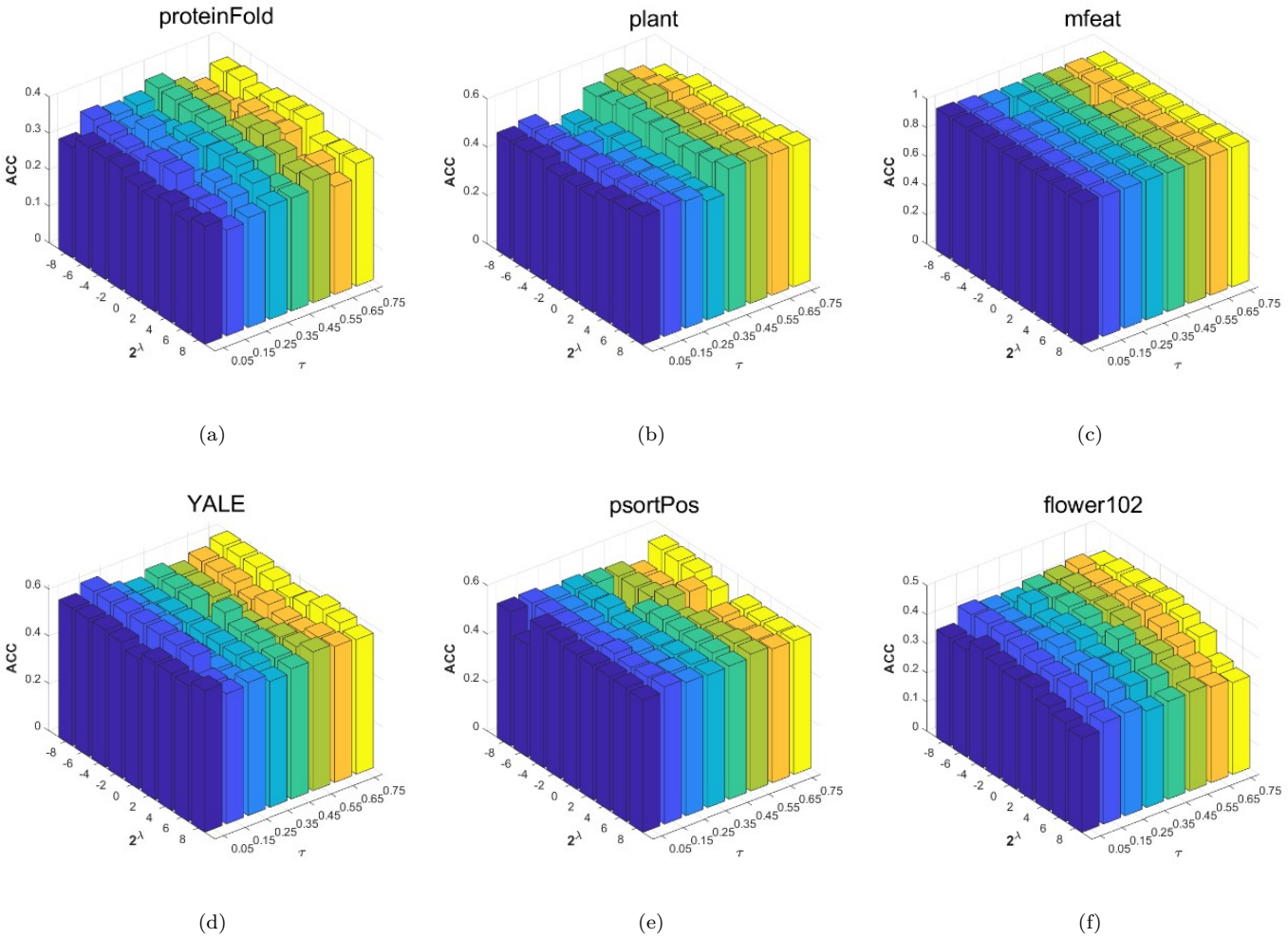
In this article, a brief but effective late fusion multiple kernel clustering with local kernel alignment maximization algorithm (LF-MKC-LKA) has been proposed to improve multi-kernel clustering performance. We used a quick but effective way to build local kernels which preserves only the highly confident local similarities. Then we use them to generate base and average partitions which are used as input to a late fusion algorithm. The three-step iterative optimization algorithm we designed can obtain the closed form optimal solution at each step to ensure the convergence. In consequence of the strong partition ability and robustness of late fusion inputs, our algorithm achieves superior performance over 11 benchmark datasets. Low time complexity, fast convergence, excellent performance, anti-noise, robustness, low parameter sensitivity and strong generalization ability on many different types of benchmark datasets are the highlights of the algorithm. The method of mining local structure information and the generated late fusion input which is better than spectral clustering and kernel k-means can be easily extended to other late fusion multiple kernel clustering problems. In the future, we plan to use OT distance or KL divergence instead of maximum alignment to get consensus partition.

Acknowledgment

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61922088, 61906020, 61872371 and 62006237).

References

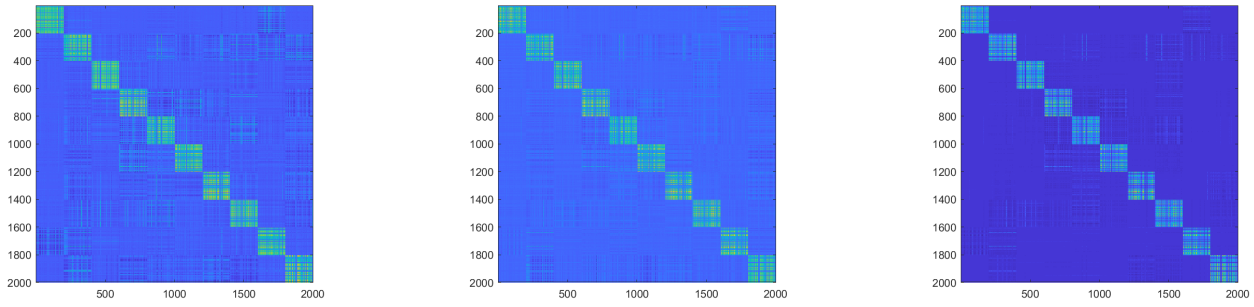
[1] Z. Li, C. Tang, X. Liu, X. Zheng, G. Yue, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," IEEE Transactions on Multimedia, 2021.



(a) (b) (c)

(d) (e) (f)

Figure 7: The sensitivity of the proposed LF-MKC-LKA with the variation of μ and τ on 6 benchmark datasets.



(a) 1 iteration of UCI_DIGIT (b) 10 iterations of UCI_DIGIT (c) 50 iterations of UCI_DIGIT

Figure 8: The similarity of UCI with 1 , 10 and 50 iterations

[2] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, and L. Wang, "Learning a joint affinity graph for multiview subspace clustering," IEEE Transactions on Multimedia, vol. 21, no. 7, pp. 1724–1736, 2018.

[3] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, ser. IJCAI'16. AAAI Press, 2016, p. 1704–1710.

[4] M. Sun, S. Wang, P. Zhang, X. Liu, S. Zhou, X. Guo, and E. Zhu, "Projective multiple kernel subspace clustering," IEEE Transactions on Multimedia, 2021.

[5] X. Liu, L. Wang, J. Yin, E. Zhu, and J. Zhang, "An efficient approach to integrating radius information into multiple kernel learning," IEEE transactions on cybernetics, vol. 43, no. 2, pp. 557–569, 2013.

[6] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, ser. AAAI'16. AAAI Press, 2016, p. 1888–1894.

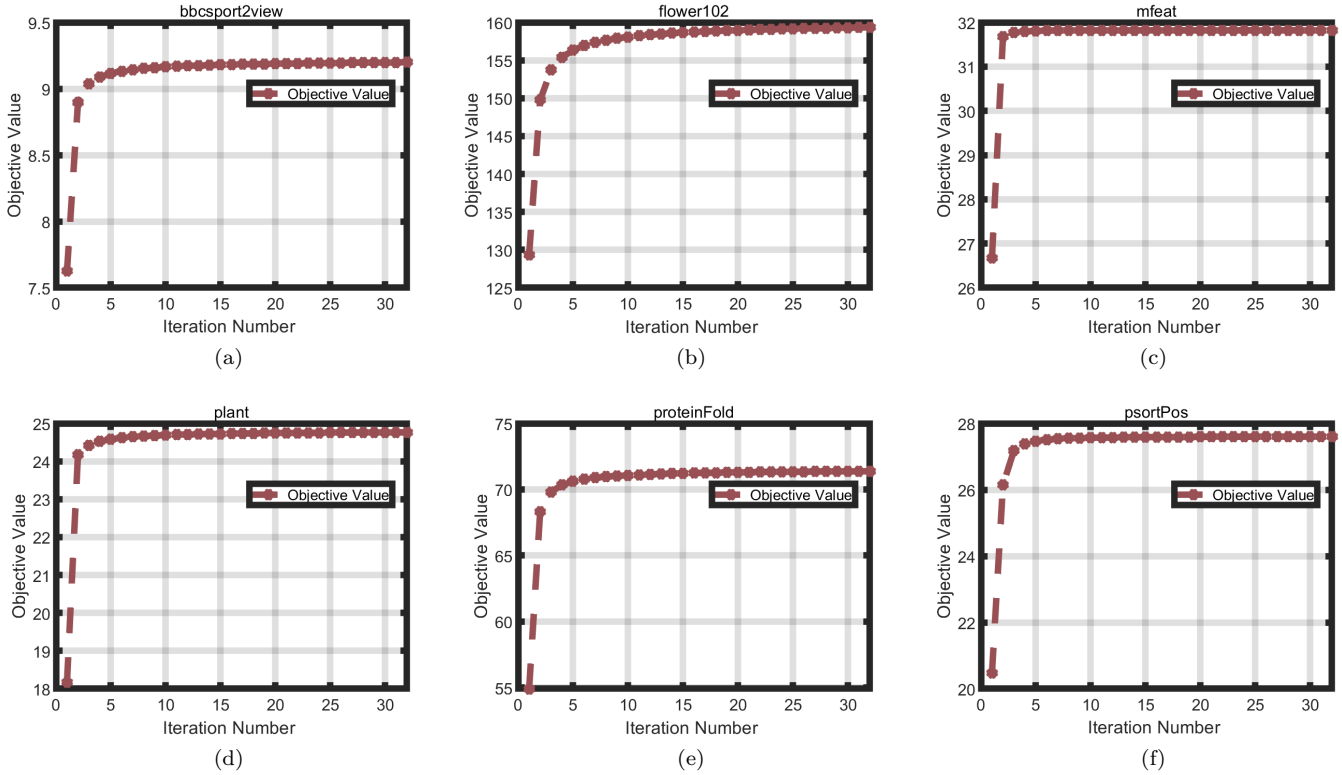


Figure 9: The convergence of the proposed LF-MKC-LKA on 6 benchmark datasets.

[7] J. Liu, X. Liu, J. Xiong, Q. Liao, S. Zhou, S. Wang, and Y. Yang, "Optimal neighborhood multiple kernel clustering with adaptive local kernels," IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2020.

[8] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, "Deep multi-view subspace clustering with unified and discriminative learning," IEEE Transactions on Multimedia, pp. 1–1, 2020.

[9] X. Xiao, Y.-J. Gong, Z. Hua, and W.-N. Chen, "On reliable multi-view affinity learning for subspace clustering," IEEE Transactions on Multimedia, 2020.

[10] H. Wang, Y. Yang, and B. Liu, "Gmc: Graph-based multi-view clustering," IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 6, pp. 1116–1129, 2019.

[11] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in Proceedings of the eleventh annual conference on Computational learning theory, 1998, pp. 92–100.

[12] A. Kumar, P. Rai, and H. Daumé, "Co-regularized multi-view spectral clustering," Curran Associates Inc., 2011.

[13] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in Proceedings of the 26th International Joint Conference on Artificial Intelligence, ser. IJCAI'17. AAAI Press, 2017, p. 2843–2849.

[14] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 5, pp. 1031–1039, 2011.

[15] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, "Multiple kernel k-means with incomplete kernels," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–14, 2019.

[16] S. Zhou, Q. Ou, X. Liu, S. Wang, L. Liu, S. Wang, E. Zhu, J. Yin, and X. Xu, "Multiple kernel clustering with compressed subspace alignment," IEEE Transactions on Neural Networks and Learning Systems, pp. 1–12, 2021.

[17] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization." in IJCAI, 2019, pp. 3778–3784.

[18] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 10, pp. 2410–2423, 2018.

[19] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized dimensionality reduction for k-means clustering," IEEE Transactions on Information Theory, vol. 61, no. 2, pp. 1045–1062, 2014.

[20] J. Chen, Z. Zhao, J. Ye, and H. Liu, "Nonlinear adaptive distance metric learning for clustering," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007, pp. 123–132.

[21] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," IEEE transactions on pattern analysis and machine intelligence, 2020.

[22] S. Jegelka, A. Gretton, B. Schölkopf, B. Sriperumbudur, U. V. Luxburg, M. Hund, and Z. Aziz, "Generalized clustering via kernel embeddings," Springer-Verlag, 2009.

[23] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 6, pp. 1083–1095, 2013.

[24] P. Wei, Y. Ke, and C. K. Goh, "Feature analysis of marginalized stacked denoising autoenconder for unsupervised domain adaptation," IEEE transactions on neural networks and learning systems, vol. 30, no. 5, pp. 1321–1334, 2018.

[25] X. Li, H. Zhang, R. Zhang, Y. Liu, and F. Nie, "Generalized uncorrelated regression with adaptive graph for unsupervised feature selection," IEEE transactions on neural networks and learning systems, vol. 30, no. 5, pp. 1587–1595, 2018.

[26] H. Dou, D. Ming, Z. Yang, Z. Pan, Y. Li, and J. Tian, "Object-based visual saliency via laplacian regularized kernel regression," IEEE Transactions on Multimedia, vol. 19, no. 8, pp. 1718–1729, 2017.

[27] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," IEEE transactions on neural networks and learning systems, vol. 30, no. 4, pp. 1265–1271, 2018.

[28] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, and J. Yin, "Multiple kernel clustering with neighbor-kernel sub-

space segmentation," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 4, pp. 1351–1362, 2020.
[29] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in Proceedings of the 28th international conference on machine learning (ICML-11), 2011, pp. 393–400.
[30] J. Zhang, Y. Liu, H. Luan, J. Xu, and M. Sun, "Prior knowledge integration for neural machine translation using posterior regularization," arXiv preprint arXiv:1811.01100, 2018.
[31] B. Schölkopf, P. Simard, A. J. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," Advances in neural information processing systems, pp. 640–646, 1998.
[32] C. A. Micchelli and M. Pontil, "Learning the kernel function via regularization," Journal of Machine Learning Research, vol. 6, no. 38, pp. 1099–1125, 2005. [Online]. Available: http://jmlr.org/papers/v6/micchelli05a.html
[33] C. Ding, H. D. Simon, R. Jin, and T. Li, "A learning framework using green's function and kernel regularization with application to recommender system," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 260–269.
[34] Y. Yao, Y. Li, B. Jiang, and H. Chen, "Multiple kernel k-means clustering by selecting representative kernels," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 4983–4996, 2021.
[35] D.-W. Kim, K. Y. Lee, D. Lee, and K. H. Lee, "Evaluation of the performance of clustering algorithms in kernel-induced feature space," Pattern Recognition, vol. 38, no. 4, pp. 607–611, 2005.
[36] C. Martin, R. Rousser, and D. Brabec, "Development of a single-kernel wheat characterization system," Transactions of the ASAE, vol. 36, no. 5, pp. 1399–1404, 1993.
[37] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Multiple kernel fuzzy clustering," IEEE Transactions on Fuzzy Systems, vol. 20, no. 1, pp. 120–134, 2011.
[38] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means using l21-norm," in Twenty-fourth international joint conference on artificial intelligence, 2015.
[39] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in Proceedings of the AAAI conference on artificial intelligence, vol. 28, no. 1, 2014.
[40] M. Chen, L. Huang, C. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020, pp. 3513–3520. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/5756
[41] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in Proceedings of the 26th International Joint Conference on Artificial Intelligence, ser. IJCAI'17. AAAI Press, 2017, p. 2564–2570.
[42] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," IEEE transactions on neural networks and learning systems, vol. 32, no. 5, pp. 1839–1851, 2020.
[43] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," The Journal of Machine Learning Research, vol. 13, no. 1, pp. 795–828, 2012.
[44] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," Neural, Parallel & Scientific Computations, vol. 11, no. 4, pp. 351–368, 2003.

Tiejian Zhang is a student of College of Computer Science and Technology in National University of Defense Technology, Hunan, China. He is pursuing a master's degree in Science and Technology on Parallel and Distributed Processing Laboratory (PDL). What he wants to do most is to work hard to get the qualification for the doctor's degree. His current research interests include multi-view clustering, kernel learning, subspace partition as well as incomplete clustering. Zhang was involved in a T-KDE paper in 2020.

Xinwang Liu (SM' 20) received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc. He serves as the associated editor of Information Fusion Journal. More information can be found at https://xinwangliu.github.io/.

Lei Gong graduated from Ocean University of China, Qingdao, China. She is now a student in College of Computer, National University of Defense Technology (NUDT), Hunan, China. She is working hard for pursuing her master degree. Her current research interests include unsupervised graph learning, deep graph clustering, multi-view clustering and kernel learning.

Siwei Wang is pursuing his P.H.D degree in National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering and deep unsupervised learning. He has published several papers and served ass PC member/ Reviewer in top journals and conferences such as IEEE TKDE, IEEE TNNLS, CVPR, AAAI, IJCAI, etc.

Xin Niu received the B.S. degree in computer science and technology from the National University of Defense Technology, Changsha, China and the Ph.D degree in Geoinformatics from the Royal Institute of Technology–KTH, Stockholm, Sweden. He has been an Associate Professor in Science and Technology on Parallel and Distributed Processing Laboratory (PDL) in College of Computer Science and Technology in National University of Defense Technology. His research interests include machine learning, computer architecture and remote sensing.

LiShen received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer Science, NUDT, China. His main research interests are image super resolution, machine learning and performance optimization of machine learning systems. Dr. Shen has published 40 research papers, including IEEE TC, IEEE TPDS, Micro, HPCA, DAC, etc.