




Article

One-Step Clustering with Adaptively Local Kernels and a Neighborhood Kernel

Cuiling Chen ¹, Zhijun Hu ², Hongbin Xiao ¹, Junbo Ma ^{1,3,*} and Zhi Li ^{1,*}

¹ School of Computer Science and Engineering, Guangxi Normal University, 15 Yucai Road, Guilin 541004, China; mathchen@163.com (C.C.); hongbinxiao@stu.gxnu.edu.cn (H.X.)

² School of Mathematics and Statistics, Guangxi Normal University, 15 Yucai Road, Guilin 541004, China; huzhijun@mailbox.gxnu.edu.cn

³ Department of Psychiatry, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

* Correspondence: junbo_ma@med.unc.edu (J.M.); zhili@gxnu.edu.cn (Z.L.)

Abstract: Among the methods of multiple kernel clustering (MKC), some adopt a neighborhood kernel as the optimal kernel, and some use local base kernels to generate an optimal kernel. However, these two methods are not synthetically combined together to leverage their advantages, which affects the quality of the optimal kernel. Furthermore, most existing MKC methods require a two-step strategy to cluster, i.e., first learn an indicator matrix, then executive clustering. This does not guarantee the optimality of the final results. To overcome the above drawbacks, a one-step clustering with adaptively local kernels and a neighborhood kernel (OSC-ALK-ONK) is proposed in this paper, where the two methods are combined together to produce an optimal kernel. In particular, the neighborhood kernel improves the expression capability of the optimal kernel and enlarges its search range, and local base kernels avoid the redundancy of base kernels and promote their variety. Accordingly, the quality of the optimal kernel is enhanced. Further, a soft block diagonal (BD) regularizer is utilized to encourage the indicator matrix to be BD. It is helpful to obtain explicit clustering results directly and achieve one-step clustering, then overcome the disadvantage of the two-step strategy. In addition, extensive experiments on eight data sets and comparisons with six clustering methods show that OSC-ALK-ONK is effective.

Keywords: local kernels; neighborhood kernel; multiple kernel clustering; block diagonal representation

MSC: 68T10; 91C20; 62H30



Citation: Chen, C.; Hu, Z.; Xiao, H.; Ma, J.; Li, Z. One-Step Clustering with Adaptively Local Kernels and a Neighborhood Kernel. *Mathematics* **2023**, *11*, 3950. <https://doi.org/10.3390/math11183950>

Academic Editor: Faheim Sufi

Received: 28 August 2023

Revised: 14 September 2023

Accepted: 15 September 2023

Published: 17 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The data in real problems usually contain nonlinear structures. When clustering these data, it is necessary to use a clustering method that can capture the nonlinear structure. Multiple kernel clustering (MKC) has the advantage of not only processing nonlinear data but also fusing the information of multiple given kernels to yield an optimal kernel. Therefore, it attracts extensive attention from scholars. Recently, many MKC methods for generating an optimal kernel have been proposed.

One strategy is to use a linear combination of given kernels to form an optimal kernel. The weights of given kernels in [1,2] are learned by ℓ_1 -regular term, while the weights of given kernels in [3,4] are yielded from ℓ_2 -regular term. More generally, ℓ_p -regular term [5,6] is used to optimize the weights of given kernels and learn an optimal kernel, and it makes the selection of regular term more flexible. In addition, many research studies adopt the strategy of linear combination to learn the optimal kernel [7–11]. In particular, a mini-max model is utilized in a simple MKC method (SimpleMKKM) to learn the kernel coefficient and update the indicator matrix [12]. It is worth noting that this strategy is based on the assumption that the optimal kernel stays in a linear combination of given kernels. This

assumption may not hold according to the fact, because this strategy restricts the search scope of the optimal kernel and degrades its quality.

In order to expand the search scope of the optimal kernel, a neighborhood kernel is used in [13–15]. The optimal kernel in [13,14] is learned from a neighborhood of the consensus kernel, where a low-rank constraint [14] is applied to the neighborhood kernel to reveal the clustering structure between samples. In particular, the base neighbor kernels with block diagonal structure [15] are produced by defining the neighbor kernel of base kernels, and then an optimal kernel is obtained by combining linearly the neighbor kernels. However, the neighborhood kernels in the literature above are generated from all the base kernels. The disadvantage is that it leads to the redundancy of base kernels because of not taking into account the correlation between given kernels.

Based on the consideration of the correlation between given kernels, selecting local base kernels to generate an optimal kernel emerges. This can avoid the redundancy of given kernels and promote diversity. On the basis of simpleMKKM [12], by considering the similarity of k -nearest neighbors between samples, a local simpleMKKM is proposed [16]. By selecting subsets from the predefined kernel pool to determine local kernels, an MKC method by using representative kernels (MKKM-RK) to learn an optimal kernel is presented [17]. In [18], a matrix-induced regularization is applied in an MKC method (MKKM-MR) to measure the correlation between each pair of kernels to generate an optimal kernel, where the kernels with strong correlation are assigned smaller coefficients, and those with weak correlation are assigned larger coefficients. By constructing the index set of samples to select local base kernels, the optimal kernel is relaxed into a neighborhood of the combination of local base kernels [19].

In recent years, various kernel evaluation methods for model selection have emerged in endless succession; for example, kernel alignment [20], kernel polarization [21], kernel class separability [22], etc. Among them, kernel alignment is one of the most commonly used evaluation methods on account of its simplicity, efficiency, and theoretical support. For example, centered kernel alignment is merged in an MKC method [23]. And, in [24], a local kernel alignment strategy is proposed by requiring only one sample to align with its k -nearest neighbors. Further, the global and local structure alignment, i.e., the internal structure of the data, is preserved in [25].

The research mentioned above fully shows that MKC has been widely used. However, most of them only adopt either a neighborhood kernel or local base kernels and do not combine these two methods together. Thus, they cannot broaden the search area of the optimal kernel and promote the variety of given kernels simultaneously and therefore cannot ensure the quality of the optimal kernel. In addition, most of the above methods require two steps; that is, first obtain the indicator matrix and then perform clustering. The two-step strategy does not guarantee the reliability and optimality of the final results because of error propagation and accumulation from each step.

In an ideal state, there is only one nonzero element in each row of the indicator matrix and the column in which the nonzero element resides corresponds to the cluster to which the sample belongs. That is, the indicator matrix in the ideal state directly displays clustering results. In this state, multiplying the indicator matrix by its transpose yields a block diagonal (BD) matrix [23]. However, in the actual clustering process, the indicator matrix is usually not the ideal case. As a result, clustering results can only be obtained after clustering is performed on the indicator matrix. This is why most MKC methods adopt the two-step operation. The shortcomings of this operation have been mentioned above. In this case, the product of the indicator matrix and its transpose is not BD. Nevertheless, we can think in reverse: if the product is BD, the indicator matrix is guided towards the ideal state, and clustering results are obtained directly.

Inspired by the above idea, we impose a BD constraint on the product of the indicator matrix and its transpose to guide it be BD, which aims to obtain clustering results directly from the indicator matrix, i.e., one-step clustering. Then, we propose a one-step clustering with adaptively local kernels and a neighborhood kernel (OSC-ALK-ONK) in this paper.

This method not only merges the advantages of local base kernels and the neighborhood kernel but also achieves one-step clustering. The process of generating a neighborhood kernel can be seen in Figure 1.

Here are the main contributions of this paper.

- By considering the correlation between base kernels, a simple strategy for selecting local base kernels is used to produce a consensus kernel, which adjusts adaptively to avoid the redundancy of given kernels and promote variety.
- By selecting a neighborhood kernel of the consensus kernel as the optimal kernel, the expression capability of the optimal kernel is improved and its search scope is expanded.
- A soft BD regularizer is used to encourage the product of the indicator matrix and its transpose to be BD, which means that the clustering results are obtained from the indicator matrix directly. Therefore, one-step clustering is realized, which ensures the final clustering results are optimal.
- A four-step iterative algorithm including the Riemann conjugate gradient method in [26], is used to overcome the difficulty of solving the model.
- Extensive experiment results conducted on eight benchmark datasets and compared with six clustering methods indicate that OSC-ALK-ONK is effective.

The remaining sections of the paper are as follows. Section 2 presents the notations used and the background of MKKC. In Section 3, the proposed OSC-ALK-ONK method and the optimization process are introduced in detail. Section 4 presents the experimental results and makes some discussions. The conclusions are stated in Section 5.

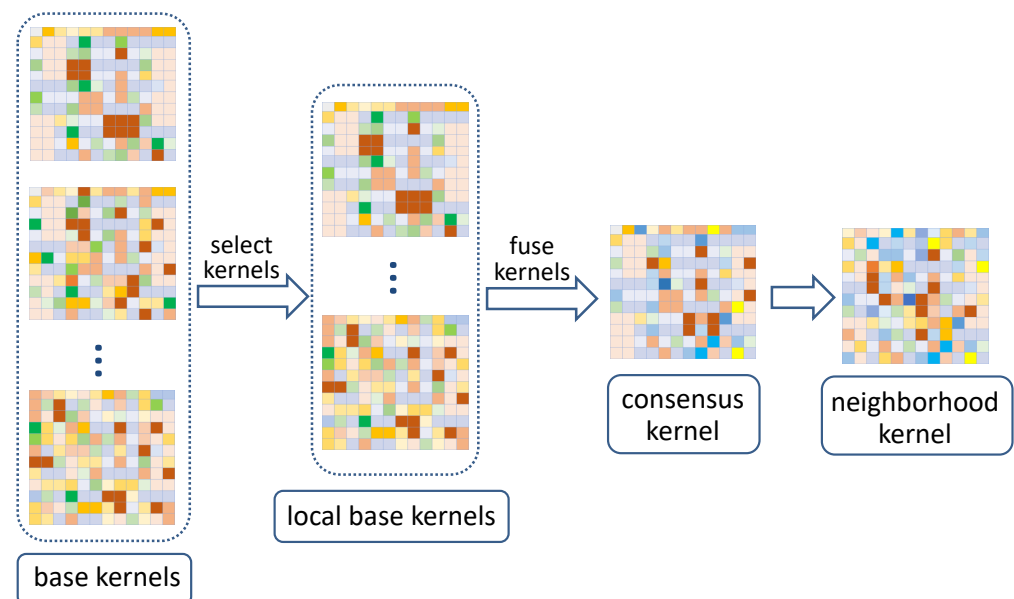


Figure 1. The process of generating a neighborhood kernel.

2. Related Work

2.1. Notations

The details of notations used in this paper are listed in Table 1.

Table 1. Details of notations.

$\ \mathbf{A}\ _F$	Frobenius norm of \mathbf{A} , i.e., $\ \mathbf{A}\ _F = \sqrt{\sum_{i,j} \mathbf{A}_{ij}^2}$
\mathbf{A}^T	transpose of \mathbf{A}
$tr(\mathbf{A})$	trace of \mathbf{A}
$Diag(\mathbf{A})$	diagonal matrix with diagonal elements of \mathbf{A}
$\mathbf{A} \succeq 0$	positive semi-definite \mathbf{A}
\mathbf{I}_k	k -order identity matrix
$\mathbf{1}_n$	all-one column vector
$\mathbf{1}_M$	all-one matrix

2.2. Kernel k -Means Clustering (KKC)

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a set of samples and $\phi(\cdot): \mathcal{X} \rightarrow \mathcal{H}$ be a kernel mapping from an original space \mathcal{X} to a reproducing Hilbert space \mathcal{H} . Kernel k -means clustering (KKC) is usually expressed as

$$\begin{aligned} \min_{\mathbf{Z} \in \{0,1\}^{n \times k}} \quad & \sum_{i=1}^n \sum_{c=1}^k Z_{ic} \|\phi(\mathbf{x}_i) - \mu_c\|_2^2 \\ \text{s.t.} \quad & \sum_{c=1}^k Z_{ic} = 1, \end{aligned} \tag{1}$$

where $\mathbf{Z} \in \{0, 1\}^{n \times k}$ is an assignment matrix, k is the number of clusters,

$$\mu_c = \frac{1}{n_c} \sum_{i=1}^n Z_{ic} \phi(\mathbf{x}_i), \quad n_c = \sum_{i=1}^n Z_{ic} \tag{2}$$

are the centroid and the number of the c -th ($1 \leq c \leq k$) cluster.

Denoting the design matrix as $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_n)] \in \mathcal{R}^{d \times n}$ and the centroid matrix as $\mathbf{U} = [\mu_1, \mu_2, \dots, \mu_k] \in \mathcal{R}^{d \times k}$, problem (1) can be rewritten as

$$\begin{aligned} \min_{\mathbf{Z} \in \{0,1\}^{n \times k}} \quad & Tr((\Phi - \mathbf{UZ}^T)^T(\Phi - \mathbf{UZ}^T)) \\ \text{s.t.} \quad & \mathbf{Z} \cdot \mathbf{1}_k = \mathbf{1}_n. \end{aligned} \tag{3}$$

Taking $\mathbf{L} = diag([n_1^{-1}, n_2^{-1}, \dots, n_k^{-1}])$, then $\mathbf{Z}^T \mathbf{Z} = \mathbf{L}^{-1}$, $\mathbf{U} = \Phi \mathbf{Z} \mathbf{L}$. And taking a kernel matrix \mathbf{K} with $\mathbf{K}_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, problem (3) can be simplified as

$$\begin{aligned} \min_{\mathbf{Z} \in \{0,1\}^{n \times k}} \quad & Tr(\mathbf{K} - \mathbf{KZLZ}^T) \\ \text{s.t.} \quad & \mathbf{Z} \cdot \mathbf{1}_k = \mathbf{1}_n. \end{aligned} \tag{4}$$

According to the matrix decomposition, problem (4) is equivalent to

$$\begin{aligned} \min_{\mathbf{Z} \in \{0,1\}^{n \times k}} \quad & Tr(\mathbf{K} - \mathbf{L}^{\frac{1}{2}} \mathbf{Z}^T \mathbf{KZL}^{\frac{1}{2}}) \\ \text{s.t.} \quad & \mathbf{Z} \cdot \mathbf{1}_k = \mathbf{1}_n. \end{aligned} \tag{5}$$

The difficulty of solving (5) is from the discreteness of \mathbf{Z} . To overcome this difficulty, the discrete \mathbf{Z} is usually relaxed to arbitrary real values, and its approximate values are treated as the solution of (5). Specifically, denoting $\mathbf{H} = \mathbf{ZL}^{\frac{1}{2}}$, the following relaxed form of (5) is derived:

$$\begin{aligned} \min_{\mathbf{H} \in \mathbb{R}^{n \times k}} \quad & Tr(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) \\ \text{s.t.} \quad & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{6}$$

where $\mathbf{H} \in \mathbb{R}^{n \times k}$, \mathbf{I}_k is a k -order identity matrix. The optimal \mathbf{H} for (6) is made up of the k eigenvectors corresponding to the k largest eigenvalues of \mathbf{K} .

2.3. Multiple Kernel k-Means Clustering (MKKC)

In MKC, a consensus kernel is computed by

$$\mathbf{K}_w = \sum_{p=1}^m w_p^2 \mathbf{K}_p, \tag{7}$$

where \mathbf{K}_p is the p -th base kernel, w_p is the p -th component of the weight vector $\mathbf{w} = [w_1, w_2, \dots, w_m]^T$, m is the number of base kernels.

Replacing \mathbf{K} in (6) with \mathbf{K}_w , the model of MKKC is:

$$\begin{aligned} \min_{\mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{w} \in \mathbb{R}_+^m} & \text{Tr}(\mathbf{K}_w(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) \\ \text{s.t.} & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \mathbf{w}^T \mathbf{1}_m = 1. \end{aligned} \tag{8}$$

Problem (8) can be solved by updating \mathbf{H} and \mathbf{w} alternately. (i) Updating \mathbf{H} with fixed \mathbf{w} , i.e., solving the similar one to problem (6). (ii) Updating \mathbf{w} with fixed \mathbf{H} , i.e., solving a quadratic programming problem:

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}_+^m} & \sum_{p=1}^m w_p^2 \text{Tr}(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) \\ \text{s.t.} & \mathbf{w}^T \mathbf{1}_m = 1. \end{aligned} \tag{9}$$

3. Proposed Method

3.1. Localized Kernel Selection

For a series of base kernels $\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_m$, considering the relationship between base kernel pairs, we define

$$y_{pq} = \begin{cases} 0, & \text{if } \|\mathbf{G} - \mathbf{K}_p\|_F^2 - \|\mathbf{G} - \mathbf{K}_q\|_F^2 < \delta, \\ 1, & \text{else.} \end{cases} \tag{10}$$

For the matrix \mathbf{G} and a given positive parameter δ , in one hand, $\|\mathbf{G} - \mathbf{K}_p\|_F^2 - \|\mathbf{G} - \mathbf{K}_q\|_F^2 < \delta$ means $\mathbf{K}_p, \mathbf{K}_q$ are both in the neighborhood of \mathbf{G} , i.e., they have large similarity. In this case, we set that $y_{pq} = 0$, which aims to discard the base kernels with high similarity. On the other hand, if $\|\mathbf{G} - \mathbf{K}_p\|_F^2 - \|\mathbf{G} - \mathbf{K}_q\|_F^2 < \delta$ does not hold, we set that $y_{pq} = 1$, which means that we select the base kernels with low similarity to yield an optimal kernel. In summary, (10) can effectively avoid the redundancy of base kernels while maintaining their variety.

Evidently, y_{pq} in (10) reflects the similarity between \mathbf{K}_p and \mathbf{K}_q , then $\sum_{q=1}^m y_{pq}$ represents the similarity between \mathbf{K}_p and all the $\mathbf{K}_q (q = 1, 2, \dots, m)$, $\text{Tr}(\mathbf{Y}^T \mathbf{1}_M)$ is the total similarity between each \mathbf{K}_p and $\mathbf{K}_q (q = 1, 2, \dots, m)$.

Let $w_p = \frac{1}{\text{Tr}(\mathbf{Y}^T \mathbf{1}_M)} \sum_{q=1}^m y_{pq}$, then

$$w_p = \frac{1}{\text{Tr}(\mathbf{Y}^T \mathbf{1}_M)} \sum_{q=1}^m y_{pq} = \frac{1}{\text{Tr}(\mathbf{Y}^T \mathbf{1}_M)} (\mathbf{Y} \cdot \mathbf{1}_m)_p \in [0, 1], \tag{11}$$

and

$$\sum_{p=1}^m w_p = \frac{1}{\text{Tr}(\mathbf{Y}^T \mathbf{1}_M)} \sum_{p=1}^m (\mathbf{Y} \cdot \mathbf{1}_m)_p = \frac{1}{\text{Tr}(\mathbf{Y}^T \mathbf{1}_M)} \text{Tr}(\mathbf{Y}^T \mathbf{1}_M) = 1. \tag{12}$$

Thereby such a w_p can balance the contribution of different given kernels to generate an optimal kernel.

3.2. Block Diagonal Regularizer

The clustering indicator matrix \mathbf{H} in (6) and (8) is not a square matrix. In the ideal case, its element can be computed as:

$$\mathbf{H}_{ij} = \begin{cases} \frac{1}{\sqrt{n_j}}, & \text{if } \mathbf{x}_i \in C_j, \\ 0, & \text{if } \mathbf{x}_i \notin C_j, \end{cases} \tag{13}$$

where x_i denotes the i -th sample, C_j denotes the j -th cluster, n_j represents the number of samples in C_j . From (13), only one element in each row of \mathbf{H} is nonzero, and this means the corresponding sample belongs to one and only one cluster. Further, if the samples are arranged from C_1 to C_k by the cluster they belong to, then $\mathbf{H}\mathbf{H}^T$ is a BD matrix as follows:

$$\mathbf{H}\mathbf{H}^T = \begin{bmatrix} \mathbf{1}_{n_1}\mathbf{1}_{n_1}^T & & & \\ & \mathbf{1}_{n_2}\mathbf{1}_{n_2}^T & & \\ & & \ddots & \\ & & & \mathbf{1}_{n_k}\mathbf{1}_{n_k}^T \end{bmatrix} \tag{14}$$

(14) prompts us to have the following idea: If $\mathbf{H}\mathbf{H}^T$ itself has the property of (14), then it will in turn induce \mathbf{H} to have the elements as (13), which means explicit clustering results are obtained directly from \mathbf{H} .

Inspired by this idea, we hope that $\mathbf{H}\mathbf{H}^T$ possesses the BD property.

Since $\mathbf{H}\mathbf{H}^T$ is a square matrix, we view $\mathbf{H}\mathbf{H}^T$ as an adjacency matrix, then according to Laplacian matrix in graph theory, its degree matrix \mathbf{D} is a diagonal matrix with $d_{ii} = \sum_{c=1}^k (\mathbf{H}\mathbf{H}^T)_{ic}$, i.e.,

$$\mathbf{D} = \text{Diag}(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n),$$

thus

$$L_{\mathbf{H}\mathbf{H}^T} = \text{Diag}(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) - \mathbf{H}\mathbf{H}^T. \tag{15}$$

There is an important conclusion between a matrix and a Laplacian matrix.

Theorem 1 ([27]). For any $\mathbf{A} \in \mathbb{R}^{n \times n} \succeq 0$, the number of connected components (blocks) in \mathbf{A} equals the multiplicity k of the eigenvalue 0 of the corresponding Laplacian matrix $\mathbf{L}_\mathbf{A}$.

Then, \mathbf{A} has k connected components if and only if

$$\lambda_i(\mathbf{L}_\mathbf{A}) \begin{cases} > 0, & i = 1, \dots, n - k, \\ = 0, & i = n - k + 1, \dots, n, \end{cases} \tag{16}$$

where $\lambda_i(\mathbf{L}_\mathbf{A}) (i = 1, \dots, n)$ are the eigenvalues of $\mathbf{L}_\mathbf{A}$ in decreasing order.

Hence, the k -BD representation of $\mathbf{H}\mathbf{H}^T$ can be given as follows.

Definition 1 ([27]). For $\mathbf{H}\mathbf{H}^T \in \mathbb{R}^{n \times n}$, the k -BD representation is defined as the sum of the k smallest eigenvalues of $\mathbf{L}_{\mathbf{H}\mathbf{H}^T}$, i.e.,

$$\|\mathbf{H}\mathbf{H}^T\|_{[k]} = \sum_{i=n-k+1}^n \lambda_i(\mathbf{L}_{\mathbf{H}\mathbf{H}^T}). \tag{17}$$

From Theorem 1, (16) and (17), $\|\mathbf{H}\mathbf{H}^T\|_{[k]} = 0$ means that $\mathbf{H}\mathbf{H}^T$ is k -BD. Then, minimizing $\|\mathbf{H}\mathbf{H}^T\|_{[k]}$ is to encourage it to be BD. Thereby, it is a natural idea that $\|\mathbf{H}\mathbf{H}^T\|_{[k]}$ is viewed as a BD regularizer. Its advantages, such as controlling the number of blocks, are softer than the BD method in [28] and be better than the alternatives of Rank ($\mathbf{L}_{\mathbf{H}\mathbf{H}^T}$) or the convex relaxation $\|\mathbf{L}_{\mathbf{H}\mathbf{H}^T}\|_*$, are stated in detail in [27].

3.3. Objective Function

Hereto, combining localized kernel selection, the block diagonal regularizer, and choosing a neighborhood kernel as the optimal kernel, we formulate the final model as follows:

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{H}, \mathbf{K}_w} & \text{Tr}(\mathbf{G}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) + \frac{\alpha}{2} \|\mathbf{G} - \mathbf{K}_w\|_F^2 + \frac{\beta}{2} \|\mathbf{H}\mathbf{H}^T\|_{[k]} \\ \text{s.t.} & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{G} \succeq 0, \mathbf{K}_w = \sum_{p=1}^m w_p \mathbf{K}_p, \end{aligned} \tag{18}$$

where w_p is computed according to (10) and (11).

The loss function of the objective function is used to executive multiple kernel clustering, the consensus term is used to choose a neighbor kernel, and the block diagonal term is used to encourage $\mathbf{H}\mathbf{H}^T$ to be block diagonal, the aim of which is to obtain an expected \mathbf{H} as Equation (13) and to implement one-step clustering.

3.4. Optimization

The regularizer $\|\mathbf{H}\mathbf{H}^T\|_{[k]}$ in problem (18) is non-convex, which leads the difficulty of solving it. For this, a theorem is introduced to reformulate $\|\mathbf{H}\mathbf{H}^T\|_{[k]}$.

Theorem 2 ([29], p. 515). *Let $\mathbf{L} \in \mathbb{R}^{n \times n}$ and $\mathbf{L} \succeq 0$. Then*

$$\sum_{i=n-k+1}^n \lambda_i(\mathbf{L}) = \min_{\mathbf{W}} \langle \mathbf{L}, \mathbf{W} \rangle \quad s.t. \quad 0 \preceq \mathbf{W} \preceq \mathbf{I}, Tr(\mathbf{W}) = k. \tag{19}$$

By (17) and (19), then

$$\|\mathbf{H}\mathbf{H}^T\|_{[k]} = \min_{\mathbf{W}} \langle \mathbf{L}_{\mathbf{H}\mathbf{H}^T}, \mathbf{W} \rangle, \quad s.t. \quad 0 \preceq \mathbf{W} \preceq \mathbf{I}, Tr(\mathbf{W}) = k.$$

From $\langle \mathbf{L}, \mathbf{W} \rangle = Tr(\mathbf{L}^T \mathbf{W})$, problem (18) is equivalent to

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{H}, \mathbf{K}_w, \mathbf{W}} \quad & Tr(\mathbf{G}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) + \frac{\alpha}{2} \|\mathbf{G} - \mathbf{K}_w\|_F^2 \\ & + \frac{\beta}{2} Tr((Diag(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) - \mathbf{H}\mathbf{H}^T)^T \mathbf{W}) \\ s.t. \quad & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{G} \succeq 0, \mathbf{K}_w = \sum_{p=1}^m w_p \mathbf{K}_p, \\ & 0 \preceq \mathbf{W} \preceq \mathbf{I}, Tr(\mathbf{W}) = k. \end{aligned} \tag{20}$$

Although problem (20) is not jointly convex on \mathbf{W} , \mathbf{G} , \mathbf{K}_w and \mathbf{H} , it is convex for each variable with the rest variables fixed. Thus, we optimize each variable alternately to solve (20).

3.4.1. Update \mathbf{W} While Fixing \mathbf{G} , \mathbf{K}_w and \mathbf{H}

While \mathbf{G} , \mathbf{K}_w and \mathbf{H} are fixed, problem (20) is

$$\begin{aligned} \min_{\mathbf{W}} \quad & Tr((Diag(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) - \mathbf{H}\mathbf{H}^T)^T \mathbf{W}) \\ s.t. \quad & 0 \preceq \mathbf{W} \preceq \mathbf{I}, Tr(\mathbf{W}) = k. \end{aligned} \tag{21}$$

For (21), $\mathbf{W}^{k+1} = \mathbf{U}\mathbf{U}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times k}$ is composed of the k eigenvectors associated with the k smallest eigenvalues of $Diag(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) - \mathbf{H}\mathbf{H}^T$ [27].

3.4.2. Update \mathbf{G} While Fixing \mathbf{W} , \mathbf{K}_w and \mathbf{H}

While \mathbf{W} , \mathbf{K}_w and \mathbf{H} are fixed, problem (20) is the following form:

$$\begin{aligned} \min_{\mathbf{G}} \quad & Tr(\mathbf{G}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) + \frac{\alpha}{2} \|\mathbf{G} - \mathbf{K}_w\|_F^2 \\ s.t. \quad & \mathbf{G} \succeq 0. \end{aligned} \tag{22}$$

Problem (22) can be expressed as

$$\begin{aligned} \min_{\mathbf{G}} \quad & \frac{1}{2} \|\mathbf{G} - \mathbf{B}\|_F^2 \\ s.t. \quad & \mathbf{G} \succeq 0, \end{aligned} \tag{23}$$

where $\mathbf{B} = \mathbf{K}_w - \frac{1}{\alpha}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)$.

The optimal solution of problem (23) is $\mathbf{G} = \mathbf{U}_B \Sigma_B^+ \mathbf{V}_B^T$, where $\mathbf{B} = \mathbf{U}_B \Sigma_B \mathbf{V}_B^T$ is SVD of \mathbf{B} , Σ_B^+ is a diagonal matrix where the diagonal elements are the positive elements of Σ_B and zeros [13].

3.4.3. Update \mathbf{K}_w While Fixing \mathbf{W} , \mathbf{G} and \mathbf{H}

With fixed \mathbf{W} , \mathbf{G} and \mathbf{H} , problem (20) reduces to

$$\begin{aligned} \min_{\mathbf{K}_w} & \frac{\alpha}{2} \|\mathbf{G} - \mathbf{K}_w\|_F^2 \\ \text{s.t.} & \mathbf{K}_w = \sum_{p=1}^m w_p \mathbf{K}_p. \end{aligned} \tag{24}$$

By introducing a parameter γ , problem (24) can be turned into

$$\min_{\mathbf{K}_w} \frac{\alpha}{2} \|\mathbf{G} - \mathbf{K}_w\|_F^2 + \frac{\gamma}{2} \|\mathbf{K}_w - \sum_{p=1}^m w_p \mathbf{K}_p\|_F^2. \tag{25}$$

The closed-form solution of \mathbf{K}_w in (25) is computed by taking its derivative with respect to \mathbf{K}_w to zero:

$$\mathbf{K}_w = \frac{1}{\alpha + \gamma} (\alpha \mathbf{G} + \gamma \sum_{p=1}^m w_p \mathbf{K}_p). \tag{26}$$

where w_p is updated according to newly generated \mathbf{Y} that is learned from new \mathbf{G} .

3.4.4. Update \mathbf{H} While Fixing \mathbf{G} , \mathbf{K}_w and \mathbf{W}

Here, problem (20) is

$$\begin{aligned} \min_{\mathbf{H}} & -\text{Tr}(\mathbf{G} \cdot \mathbf{H}\mathbf{H}^T) + \beta \text{Tr}((\text{Diag}(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) - \mathbf{H}\mathbf{H}^T)\mathbf{W}) \\ \text{s.t.} & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \mathbf{H} \in \mathbb{R}^{n \times k}. \end{aligned} \tag{27}$$

The term $\beta \text{Tr}((\text{Diag}(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) - \mathbf{H}\mathbf{H}^T)\mathbf{W})$ leads to the difficulty of solving (27) directly. By means of matrix operations and the properties of the trace, $\text{Tr}(\text{Diag}(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) \cdot \mathbf{W}) = \text{Tr}((\mathbf{1}_M \text{Diag}(\mathbf{W})) \cdot \mathbf{H}\mathbf{H}^T)$, then (27) can be changed into

$$\begin{aligned} \min_{\mathbf{H}} & \text{Tr}(\beta(\mathbf{1}_M \text{Diag}(\mathbf{W}) - \mathbf{G}) \cdot \mathbf{H}\mathbf{H}^T) \\ \text{s.t.} & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \mathbf{H} \in \mathbb{R}^{n \times k}. \end{aligned} \tag{28}$$

Because $\mathbf{1}_M \text{Diag}(\mathbf{W})$ in (28) is not symmetric, the same solution as a kernel k -means clustering is not suitable for (28). Notably, (28) is similar to the problem on the Stiefel manifold in [26]; thus, the Riemann conjugate gradient method in [26] can be used to solve it.

These are the main steps of our proposed algorithm.

4. Experiments

4.1. Data Sets

Eight real data sets are used in our experiments, and their sizes and classes are summarized in Table 2.

Table 2. Summaries of data sets.

Data Sets	# (Samples)	# (Features)	# (Classes)
AR	840	768	120
BA	1404	320	36
CCUDS10	1944	101	10
GLIOMA	50	4434	4
ISOLET	1560	617	2
LYMPHOMA	96	4026	9
ORL	400	1024	40
YALE	165	1024	15

4.2. Comparison Methods

To demonstrate the clustering performance, we compare OSC-ALK-ONK with six clustering methods. Among them, KKM is a single kernel clustering, MKKM and RMKKM are two classic MKC methods, and MKKM-MR, SimpleMKKM, and MKKM-RK are three MKC methods recently proposed.

- KKM integrates integral operator kernel functions in principal component analysis to deal with nonlinear data [30].
- MKKM combines the fuzzy k -means clustering with multiple kernel learning, where the weights of base kernels are automatically updated to produce the optimal kernel [31].
- RMKKM is an extension based on MKKM, and its robustness is ensured by an $\ell_{2,1}$ -norm in kernel space [7].
- MKKM-MR uses a matrix-induced regularization to measure the correlation between all the kernel pairs and implements MKC [18].
- SimpleMKKM adopts a min-max model to minimize kernel alignment on the kernel coefficient and maximize kernel alignment on the clustering matrix, and is a simple MKC [12].
- MKKM-RK is an MKC method by selecting representative kernels from the base kernel pool to generate the optimal kernel [17].

4.3. Multiple Kernels' Construction

In this paper, we construct a kernel pool by selecting twelve base kernels (i.e., $m = 12$), which consists of seven radial basis function kernels with $ker(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / (2\tau\sigma^2))$, where the value of τ is selected from $\{0.01, 0.05, 0.1, 1, 10, 50, 100\}$ and σ is the maximum distance between samples; four polynomial kernels with $ker(\mathbf{x}_i, \mathbf{x}_j) = (a + \mathbf{x}_i^T \mathbf{x}_j)^b$, where a and b are chosen from $\{0, 1\}$ and $\{2, 4\}$, respectively; and a cosine kernel with $ker(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j) / (\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|)$. And all the kernels $\{\mathbf{K}_p\}_{p=1}^m$ are normalized to the range of $[0, 1]$.

4.4. Experimental Results and Analysis

To obtain better and more stable clustering performance, we utilize the ten-fold cross-validation method with the five-fold cross-validation embedded in OSC-ALK-ONK. To this end, at first, we randomly partition all the samples into ten subsets without repetition, where nine subsets are viewed as training sets and the rest are regarded as testing sets. Further, the nine training sets are partitioned into five subsets without repetition, where four subsets are utilized as training sets and the rest one is the validation set. In order to lose generality, the values of two parameters α, β change from $\in [10^{-2}, 10^{-1}, \dots, 10^1, 10^2]$. Five-fold cross validation aims to select the optimal combination of parameters α, β . The obtained optimal combinations are used in the test set to produce the final clustering results. And the number of cluster k in each data set is set as the true value of the cluster.

For each method used for comparison, we set the parameters according to the corresponding literature.

The final experimental results of each method on each data set, namely the average ACC, NMI, and Purity of 15 experiments, are reported in Table 3. The best ACC, NMI, and Purity on each data set are highlighted in boldface. The last three rows in Table 3 are the mean ACC, NMI, and Purity of each method on all the data sets. Evidently, the proposed OSC-ALK-ONK performs best. The detailed analyses are as follows.

(1) OSC-ALK-ONK outperms KKM by 54.41%, 62.44%, 53.26% according to ACC, NMI and Purity. This verifies that multiple kernel clustering is prior to single kernel clustering. (2) OSC-ALK-ONK exceeds MKKM and RMKKM by 58.62%, 67.01%, 60.53% and 31.29%, 34.36%, 32.89% in terms of ACC, NMI, and Purity. The reason should be that the combination of the local kernel method and the neighborhood kernel method is used to avoid the redundancy of the base kernel and expand the search range of the optimal kernel. And the clustering results of OSC-ALK-ONK are better than SimpleMKKM, which should also give credit to the combination of the two methods. (3) Although MKKM-MR and MKKM-RK

exceed KKM, MKKM, and RMKKM, they are inferior to OSC-ALK-ONK. The reason should be the localized kernel strategy in OSC-ALK-ONK ensures the sparsity of base kernels and successfully avoids the redundancy of base kernels. In a word, OSC-ALK-ONK improves the quality of the optimal kernel and promotes the clustering performance by combining local kernels and a neighborhood kernel. In addition, the BD representation ensures the reliability of clustering results and further promotes the clustering performance of OSC-ALK-ONK.

Overall, the experiment results show that OSC-ALK-ONK is an effective clustering method.

Table 3. Clustering results of different methods.

Dataset	Metric	KKM	MKKM	RMKKM	MKKM -MR	Simple -MKKM	MKKM -RK	Proposed
AR	ACC	0.3000	0.3167	0.3168	0.4863	0.5150	0.5047	0.6686
	NMI	0.6360	0.6350	0.6608	0.7615	0.7644	0.7608	0.8890
	Purity	0.3190	0.3437	0.3358	0.5398	0.5304	0.5305	0.7826
BA	ACC	0.2863	0.3868	0.4088	0.4177	0.4496	0.3708	0.4211
	NMI	0.4365	0.5301	0.5639	0.5882	0.5919	0.5194	0.6716
	Purity	0.3226	0.4010	0.4329	0.4619	0.4780	0.3962	0.4773
CCUDS10	ACC	0.1280	0.1214	0.1285	0.1345	0.1287	0.1287	0.2031
	NMI	0.0093	0.0081	0.0091	0.0083	0.0102	0.0073	0.1054
	Purity	0.1318	0.1234	0.1331	0.1357	0.1327	0.1310	0.2182
GLIOMA	ACC	0.5032	0.4880	0.5760	0.4955	0.5120	0.5640	0.7900
	NMI	0.3256	0.2943	0.4818	0.3083	0.2957	0.4077	0.7178
	Purity	0.5357	0.5400	0.6460	0.5341	0.5320	0.5787	0.8420
ISOLET	ACC	0.5659	0.5269	0.5643	0.5282	0.5801	0.5558	0.6489
	NMI	0.0224	0.0021	0.0121	0.0023	0.0192	0.0090	0.0862
	Purity	0.5659	0.5269	0.5643	0.5282	0.5801	0.5558	0.6500
LYMPHOMA	ACC	0.4982	0.5085	0.6135	0.5437	0.5932	0.5639	0.6929
	NMI	0.5105	0.5070	0.6172	0.6495	0.6099	0.5963	0.7070
	Purity	0.7163	0.7036	0.8031	0.7826	0.8266	0.8125	0.8350
ORL	ACC	0.4308	0.3475	0.5521	0.6357	0.6391	0.5860	0.7127
	NMI	0.6383	0.5378	0.7406	0.8163	0.8073	0.7581	0.8898
	Purity	0.4797	0.3525	0.6001	0.6908	0.6860	0.6188	0.8107
YALE	ACC	0.4182	0.3515	0.5218	0.5341	0.5512	0.5939	0.6962
	NMI	0.4330	0.4152	0.5558	0.5614	0.5826	0.5986	0.8262
	Purity	0.4424	0.3636	0.5364	0.5495	0.5555	0.5988	0.7691
Avg	ACC	0.3913	0.3809	0.4602	0.4720	0.4961	0.4835	0.6042
	NMI	0.3765	0.3662	0.4552	0.4620	0.4602	0.4572	0.6116
	Purity	0.4392	0.4193	0.5065	0.5278	0.5402	0.5278	0.6731

In order to further substantiate the effectiveness of OSC-ALK-ONK, we present the visualization of clustering results for all methods on ISOLET (for convenience, only a fifth of samples in ISOLET are chosen). As can be seen from Figure 2, OSC-ALK-ONK achieves a good clustering effect.

4.5. Ablation Study

In OSC-ALK-ONK, the weights of base kernels are adjusted adaptively, which aims to choose base kernels with small correlation and discard those with large correlation. These weights are automatically updated during the optimization process of the model. To verify the effectiveness of the localized kernel selection strategy, we adopt the uniform weight strategy as a contrast, i.e., $w_p = \frac{1}{m}$, $p = 1, 2, \dots, m$, to perform ablation study. For

convenience's sake, this model is denoted as OSC-ONK-UW. That is, all the base kernels are selected in OSC-ONK-UW.

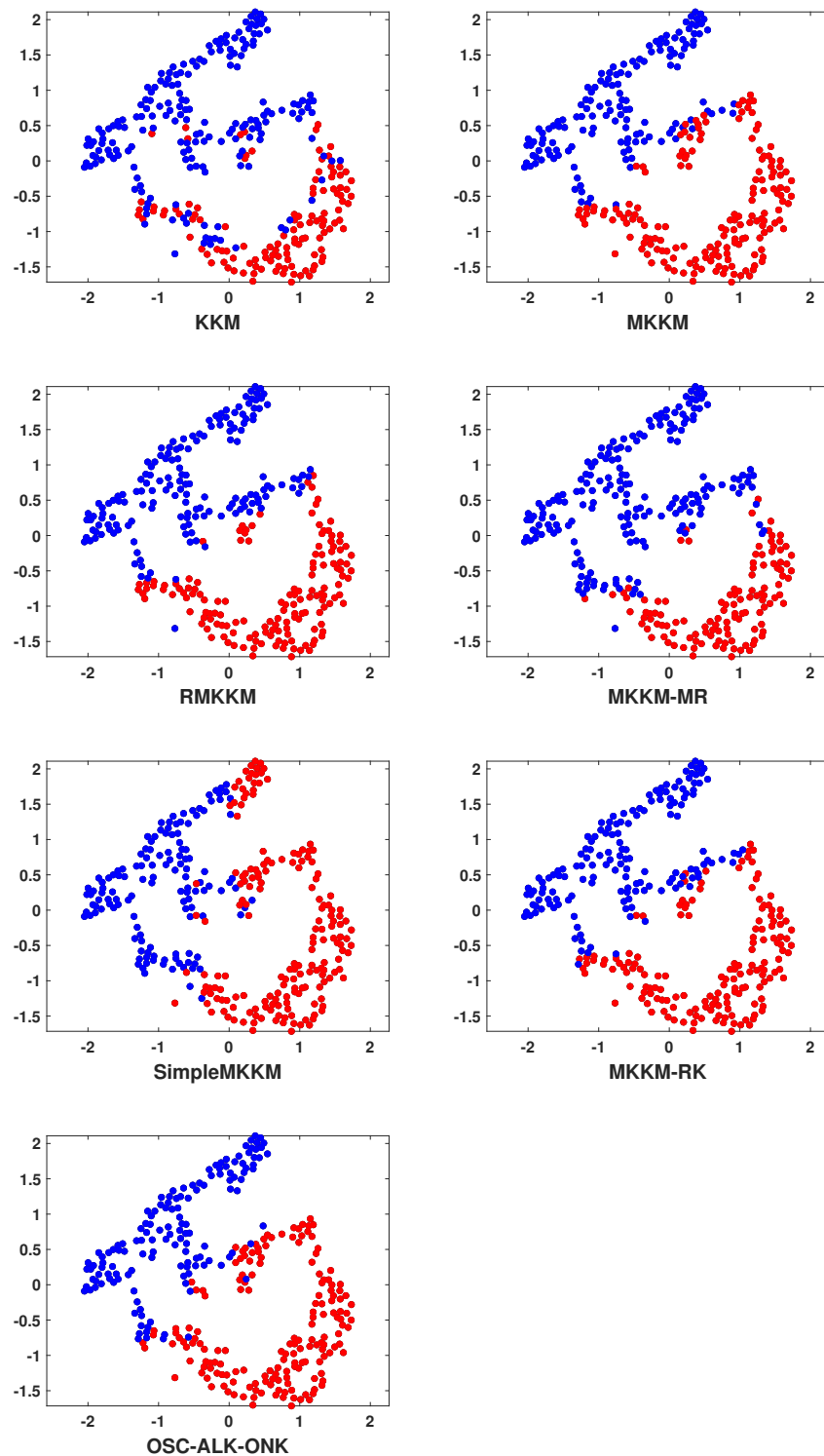


Figure 2. The visualization of clustering results for OSC-ALK-ONK and comparison methods on ISOLET.

In addition, the BD regularization term is used in our OSC-ALK-ONK. To validate its effect, we also conduct an ablation study on the model not including this term, i.e., we only consider the following model (ALK-ONK-NoBD):

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{H}, \mathbf{K}_w} & \text{Tr}(\mathbf{G}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) + \frac{\alpha}{2} \|\mathbf{G} - \mathbf{K}_w\|_F^2 \\ \text{s.t.} & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{G} \geq 0, \mathbf{K}_w = \sum_{i=1}^m w_i \mathbf{K}_i, \end{aligned} \tag{29}$$

where w_p is computed according to (10) and (11).

The results of ablation studies on eight data sets, namely OSC-ONK-UW, ALK-ONK-NoBD, and OSC-ALK-ONK, are shown in Figure 3, which indicates that OSC-ALK-ONK outperforms OSC-ONK-UW and ALK-ONK-NoBD. Accordingly, OSC-ALK-ONK improves the clustering performance through the strategy of localized kernel selection and BD regularizer.

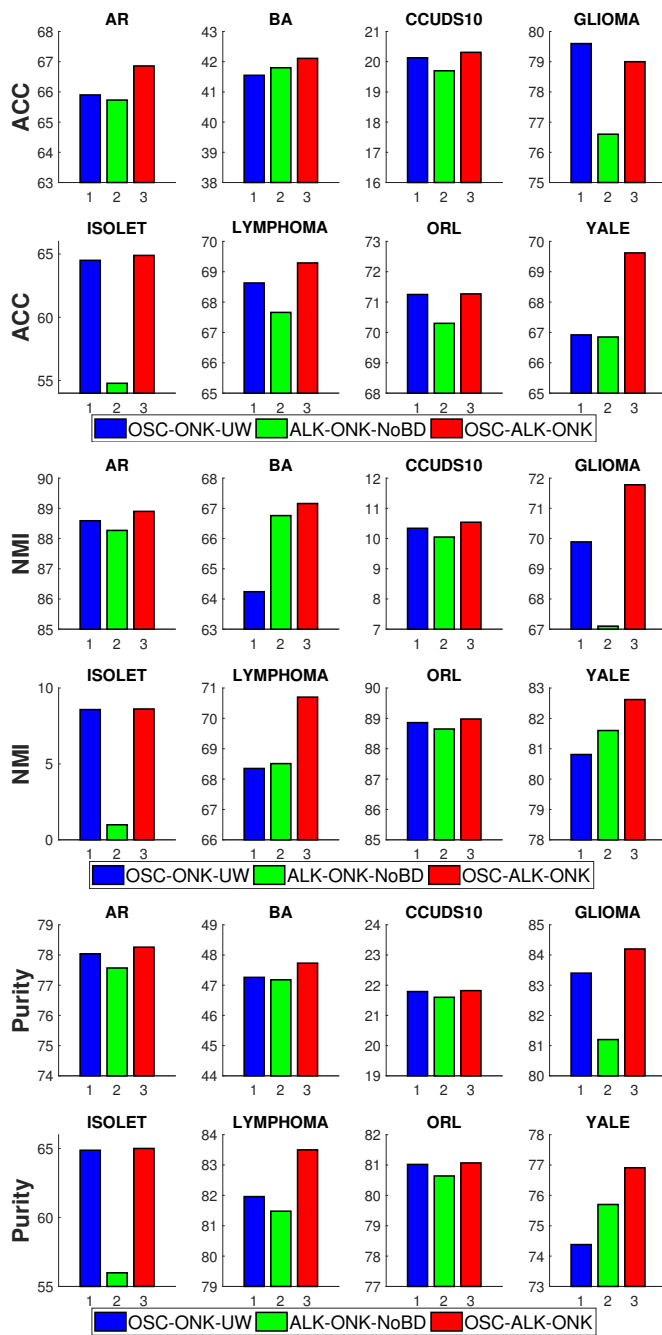


Figure 3. Comparison of clustering results among OSC-ONK-UW, ALK-ONK-NoBD and OSC-ALK-ONK on eight datasets.

4.6. Parameters' Sensitivity

The model of OSC-ALK-ONK involves the parameters α , β , and a penalty parameter γ . We set γ to be 0.1 in experiments. To verify the sensitivity of OSC-ALK-ONK to α and β , they are tuned in the ranges $[10^{-2}, 10^{-1}, \dots, 10^1, 10^2]$ via leveraging a grid search technique. Figure 4 shows the clustering performance of OSC-ALK-ONK corresponding to varying α and β , which indicates that OSC-ALK-ONK is data-driven.

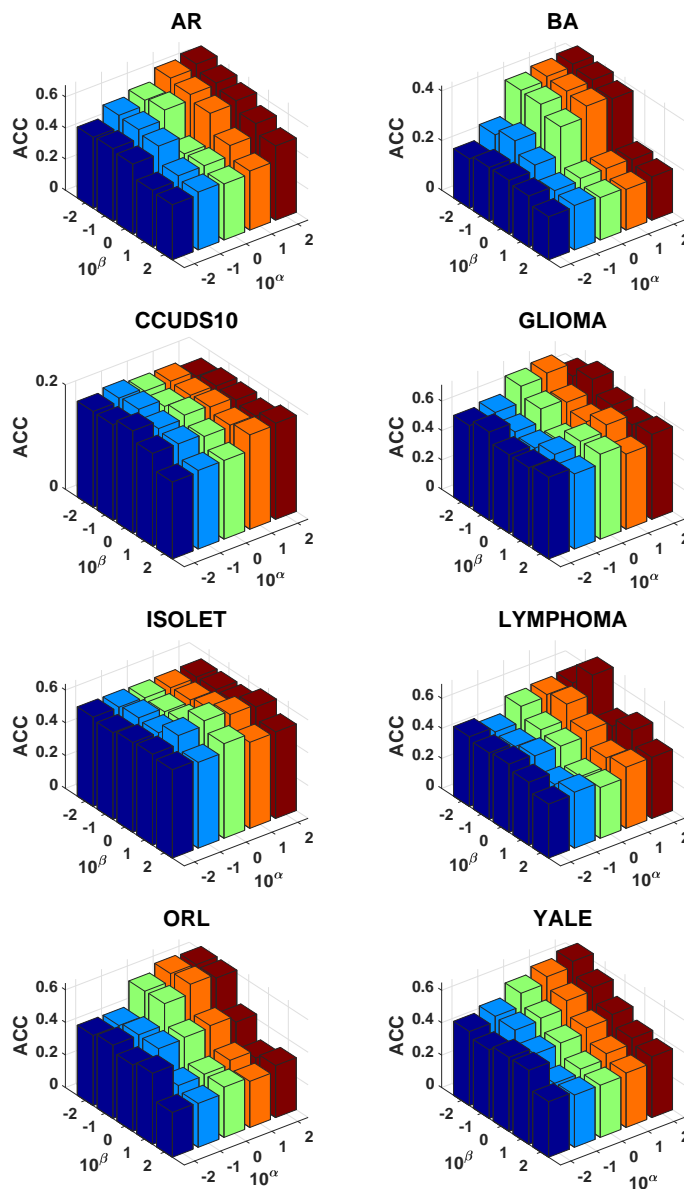


Figure 4. ACC of OSC-ALK-ONK with different parameter's settings.

4.7. Convergence

In this section, we first prove the convergence of the objective function of (20). For convenience, we express the objective function of problem (20) as

$$\begin{aligned}
 \mathcal{J}(\mathbf{W}, \mathbf{G}, \mathbf{K}_w, \mathbf{H}) = & \left\{ \min_{\mathbf{G}, \mathbf{H}, \mathbf{K}_w, \mathbf{W}} \text{Tr}(\mathbf{G}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^T)) + \frac{\alpha}{2} \|\mathbf{G} - \mathbf{K}_w\|_F^2 \right. \\
 & \left. + \frac{\beta}{2} \text{Tr}((\text{Diag}(\mathbf{H}\mathbf{H}^T \cdot \mathbf{1}_n) - \mathbf{H}\mathbf{H}^T)^T \mathbf{W}) \right. \\
 \text{s.t. } & \mathbf{H}^T \mathbf{H} = \mathbf{I}_k, \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{G} \succeq 0, \mathbf{K}_w = \sum_{q=1}^r w_q \mathbf{K}_q, \\
 & \left. 0 \preceq \mathbf{W} \preceq \mathbf{I}, \text{Tr}(\mathbf{W}) = k \right\}.
 \end{aligned}
 \tag{30}$$

When updating \mathbf{W} with fixed $\mathbf{G}, \mathbf{K}_w, \mathbf{H}$, problem (21) is a convex programming problem [27], so it can converge to the global optimal solution. We denote the optimal solution as \mathbf{W}^{t+1} , then

$$\mathcal{J}(\mathbf{W}^{(t+1)}, \mathbf{G}^t, \mathbf{K}_w^t, \mathbf{H}^t) \leq \mathcal{J}(\mathbf{W}^t, \mathbf{G}^t, \mathbf{K}_w^t, \mathbf{H}^t). \tag{31}$$

When updating \mathbf{G} with fixed $\mathbf{W}, \mathbf{K}_w, \mathbf{H}$, problem (22) is convex and the global optimal solution can be obtained, which we denote as \mathbf{G}^{t+1} , then

$$\mathcal{J}(\mathbf{W}^{t+1}, \mathbf{G}^{t+1}, \mathbf{K}_w^t, \mathbf{H}^t) \leq \mathcal{J}(\mathbf{W}^{t+1}, \mathbf{G}^t, \mathbf{K}_w^t, \mathbf{H}^t). \tag{32}$$

When updating \mathbf{K}_w with fixed $\mathbf{W}, \mathbf{G}, \mathbf{H}$, problem (25) is convex, then the global optimal solution can be obtained. It is denoted as \mathbf{K}_w^{t+1} , then

$$\mathcal{J}(\mathbf{W}^{t+1}, \mathbf{G}^{t+1}, \mathbf{K}_w^{t+1}, \mathbf{H}^t) \leq \mathcal{J}(\mathbf{W}^{t+1}, \mathbf{G}^{t+1}, \mathbf{K}_w^t, \mathbf{H}^t). \tag{33}$$

When updating \mathbf{H} with fixed $\mathbf{W}, \mathbf{G}, \mathbf{K}_w$, since $\mathbf{1}_M \text{Diag}(\mathbf{W})$ is not symmetric, it is difficult to prove that problem (28) for \mathbf{H} is convex. Nevertheless, the global convergence of the conjugate gradient method after finite step iteration has been proved in [26], i.e., the conjugate gradient method ensures that problem (28) can converge to the global optimal solution when updating \mathbf{H} . The optimal solution is denoted as \mathbf{H}^{t+1} , then

$$\mathcal{J}(\mathbf{W}^{t+1}, \mathbf{G}^{t+1}, \mathbf{K}_w^{t+1}, \mathbf{H}^{(t+1)}) \leq \mathcal{J}(\mathbf{W}^{(t+1)}, \mathbf{G}^{t+1}, \mathbf{K}_w^{t+1}, \mathbf{H}^t). \tag{34}$$

Combining (31)–(34), it is concluded that

$$\mathcal{J}(\mathbf{W}^{t+1}, \mathbf{G}^{t+1}, \mathbf{K}_w^{t+1}, \mathbf{H}^{t+1}) \leq \mathcal{J}(\mathbf{W}^t, \mathbf{G}^t, \mathbf{K}_w^t, \mathbf{H}^t). \tag{35}$$

Therefore, $\mathcal{J}(\mathbf{W}^t, \mathbf{G}^t, \mathbf{K}_w^t, \mathbf{H}^t)$ monotonically decreases at each iteration, until it converges to the global optimal solution.

The above proof shows that Algorithm 1 can monotonically reduce the value of the objective function at each iteration, i.e., the objective function is monotonically decreasing. The convergence graphs of OSC-ALK-ONK on all the data sets are shown in Figure 5, where the stopping criteria of the algorithm are $\frac{|obj(t+1)-obj(t)|}{|obj(t)|} \leq 10^{-3}$, and $obj(t)$ denotes the objective function value at the t -th iteration. Evidently, the changing trend of the objective function value with respect to the iteration number in Figure 5 shows the monotone descent. Further, they converge within 10 iterations on all the data sets, which demonstrates the rapid convergence of OSC-ALK-ONK.

Algorithm 1: Pseudo code of solving problem (18).

Input: m base kernels $\{K_p\}_{p=1}^m$ and parameters α, β, γ .

Initialize: $(\mathbf{K})^1 = \frac{1}{m} \sum_{p=1}^m \mathbf{K}_p, (\mathbf{H})^1 = rand(n, k), \{(w_p)^1\}_{p=1}^m = \frac{1}{m}$.

While not converged **do**.

(1) Update \mathbf{W}^{k+1} by solving (21).

(2) Update \mathbf{G}^{k+1} by solving (22).

(3) Update \mathbf{K}_w^{k+1} via (26).

(4) Update \mathbf{H}^{k+1} via (27), compute \mathbf{w} via (10) and (11).

end while

Obtain the optimal $\mathbf{G}^*, \mathbf{H}^*, \mathbf{K}_w^*, \mathbf{W}^*$.

Output: ACC, NMI and Purity.

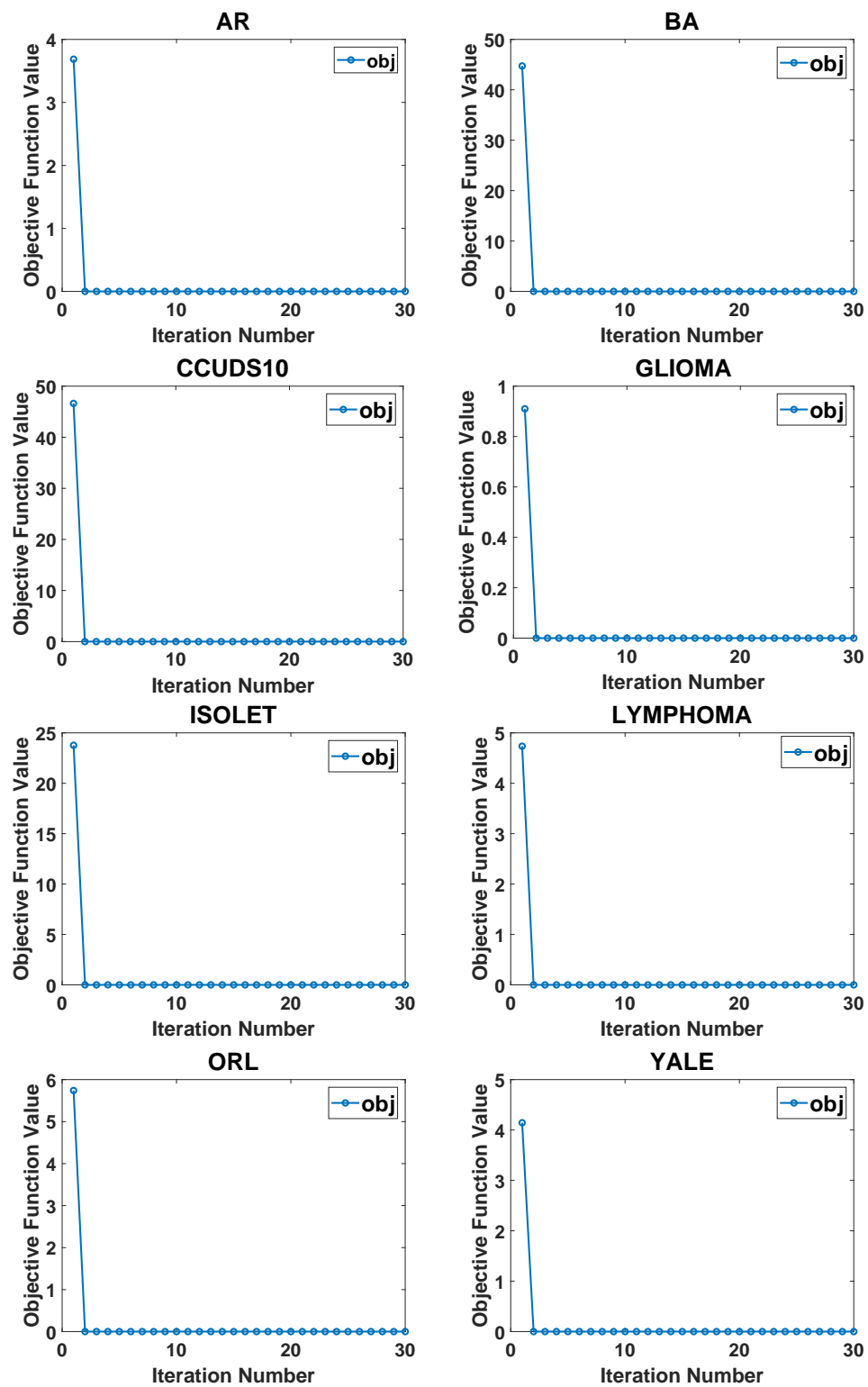


Figure 5. Objective function value of OSC-ALK-ONK at each iteration.

5. Conclusions

In this paper, we proposed a novel MKC method called OSC-ALK-ONK. It selects adaptively local given kernels to generate a consensus kernel and uses a neighborhood kernel of this consensus kernel as an optimal one. The combination of these two methods promotes the quality of the optimal kernel by enlarging its search area while avoiding the redundancy of base kernels. Furthermore, a BD regularizer is utilized on the indicator

matrix to execute one-step clustering and avoid two-step operations. In addition, sufficient experiment results indicate the effectiveness of OSC-ALK-ONK.

In real applications, a lot of data are multi-view data, which may be incomplete for some objective reasons. Due to the effectiveness of the local kernel selection strategy in this paper, it can be considered to combine this strategy with the neighborhood kernel in the future to obtain a high-quality optimal kernel in multi-view data. In addition, on account of the advantages of the BD regular term in this paper, it is also used in multi-view data, even incomplete multi-view data. All these are worth studying in the future.

Author Contributions: Conceptualization, C.C., J.M. and Z.L.; methodology, C.C.; software, Z.H. and H.X.; validation, C.C.; formal analysis, J.M. and Z.L.; resources, C.C. and Z.H.; writing—original draft preparation, C.C.; writing—review and editing, J.M. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research fund of Guangxi Key Lab of Multi-source Information Mining and Security (No. MIMS22-03, No. MIMS21-M-01), the Guangxi Natural Science Foundation (No. 2023GXNSFBA026010).

Data Availability Statement: The datasets used in the experiments are available in the corresponding website at <http://featureselection.asu.edu/datasets.php> (accessed on 12 September 2022) (i.e., AR), <http://www.cs.nyu.edu/roweis/data.html> (accessed on 3 September 2022) (i.e., BA), <https://jundongli.github.io/scikit-feature/datasets.html> (accessed on 5 October 2022) (i.e., GLIOMA, LYMPHOMA, ORL, YALE), and <https://archive-beta.ics.uci.edu/ml/datasets> (accessed on 1 November 2022) (i.e., CCUDS10, ISOLET).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Valizadegan, H.; Jin, R. Generalized maximum margin clustering and unsupervised kernel learning. In Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2006; pp. 1417–1424.
2. Zeng, H.; Cheung, Y.M. Feature selection and kernel learning for local learning-based clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1532–1547. [[CrossRef](#)] [[PubMed](#)]
3. Cortes, C.; Mohri, M.; Rostamizadeh, A. L_2 regularization for learning kernels. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 109–116.
4. Zhao, B.; Kwok, J.T.; Zhang, C.S. Multiple Kernel Clustering. In Proceedings of the SIAM International Conference on Data Mining, Sparks, NV, USA, 30 April–2 May 2009; pp. 638–649.
5. Kloft, M.; Brefeld, U.; Sonnenburg, S.; Laskov, P.; Müller, K.R.; Zien, A.; Sonnenburg, S. Efficient and accurate l_p -norm multiple kernel learning. In Proceedings of the 23rd Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 7–10 December 2009; pp. 997–1005.
6. Xu, Z.L.; Jin, R.; Yang, H.Q.; King, I.; Lyu, M.R. Simple and efficient multiple kernel learning by group lasso. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 1175–1182.
7. Du, L.; Zhou, P.; Shi, L.; Wang, H.M.; Fan, M.Y.; Wang, W.J.; Shen, Y.D. Robust multiple kernel k-means using l_2 -norm. In Proceedings of the TwentyFourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 3476–3482.
8. Kang, Z.; Lu, X.; Yi, J.F.; Xu, Z.L. Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial, Stockholm, Sweden, 13–19 July 2018; pp. 2312–2318.
9. Kang, Z.; Peng, C.; Cheng, Q.; Xu, Z.L. Unified spectral clustering with optimal graph. In Proceedings of the Thirty-Second Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 3366–3373.
10. Kang, Z.; Wen, L.J.; Chen, W.Y.; Xu, Z.L. Low-rank kernel learning for graph-based clustering. *Knowl. Based Syst.* **2019**, *163*, 510–517. [[CrossRef](#)]
11. Zhou, S.H.; Zhu, E.; Liu, X.W.; Zheng, T.M.; Liu, Q.; Xia, J.Y.; Yin, J.P. Subspace segmentation-based robust multiple kernel clustering. *Inf. Fusion* **2020**, *53*, 145–154. [[CrossRef](#)]
12. Liu, X.W. Simplemkkm: Simple multiple kernel k-means. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 5174–5186. [[PubMed](#)]
13. Liu, X.W.; Zhou, S.H.; Wang, Y.Q.; Li, M.M.; Dou, Y.; Zhu, E.; Yin, J.P. Optimal neighborhood kernel clustering with multiple kernels. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 2266–2272.
14. Ou, Q.Y.; Gao, L.; Zhu, E. Multiple kernel k-means with low-rank neighborhood kernel. *IEEE Access* **2021**, *9*, 3291–3300. [[CrossRef](#)]
15. Zhou, S.H.; Liu, X.W.; Li, M.M.; Zhu, E.; Liu, L.; Zhang, C.W.; Yin, J.P. Multiple kernel clustering with neighbor-kernel subspace segmentation. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 1351–1362. [[CrossRef](#)] [[PubMed](#)]

16. Liu, X.W.; Zhou, S.H.; Liu, L.; Tang, C.; Wang, S.W.; Liu, J.Y.; Zhang, Y. Localized simple multiple kernel k-means. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 9273–9281.
17. Yao, Y.Q.; Li, Y.; Jiang, B.B.; Chen, H.H. Multiple kernel k-means clustering by selecting representative kernels. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4983–4996. [[CrossRef](#)] [[PubMed](#)]
18. Liu, X.W.; Dou, Y.; Yin, J.P.; Wang, L.; Zhu, E. Multiple kernel k-means clustering with matrix-induced regularization. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1888–1894.
19. Liu, J.Y.; Liu, X.W.; Xiong, J.; Liao, Q.; Zhou, S.H.; Wang, S.W.; Yang, Y.X. Optimal neighborhood multiple kernel clustering with adaptive local kernels. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2872–2885. [[CrossRef](#)]
20. Afkanpour, A.; Szepesvári, C.; Bowling, M. Alignment based kernel learning with a continuous set of base kernels. *Mach. Learn.* **2013**, *91*, 305–324. [[CrossRef](#)]
21. Wang, T.H.; Tian, S.F.; Huang, H.K.; Deng, D.Y. Learning by local kernel polarization. *Neurocomputing* **2009**, *72*, 3077–3084. [[CrossRef](#)]
22. Wang, L. Feature selection with kernel class separability. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1534–1546. [[CrossRef](#)] [[PubMed](#)]
23. Lu, Y.T.; Wang, L.T.; Lu, J.F.; Yang, J.Y.; Shen, C.H. Multiple kernel clustering based on centered kernel alignment. *Pattern Recognit.* **2014**, *47*, 3656–3664. [[CrossRef](#)]
24. Li, M.M.; Liu, X.W.; Wang, L.; Dou, Y.; Yin, J.P.; Zhu, E. Multiple kernel clustering with local kernel alignment maximization. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 1704–1710.
25. Wang, C.L.; Zhu, E.; Liu, X.W.; Gao, L.; Yin, J.P.; Hu, N. Multiple kernel clustering with global and local structure alignment. *IEEE Access* **2018**, *6*, 77911–77920. [[CrossRef](#)]
26. Li, J.F.; Qin, S.J.; Zhang, L.; Hou, W.T. An efficient method for solving a class of matrix trace function minimization problem in multivariate statistical. *Math. Numer. Sin.* **2021**, *43*, 70–86.
27. Lu, C.Y.; Feng, J.S.; Lin, Z.C.; Mei, T.; Yan, S.C. Subspace clustering by block diagonal representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 487–501. [[CrossRef](#)] [[PubMed](#)]
28. Feng, J.S.; Lin, Z.C.; Xu, H.; Yan, S.C. Robust subspace segmentation with block-diagonal prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3818–3825.
29. Dattorro, J. Convex Optimization and Euclidean Distance Geometry. 2016. Available online: <http://meboo.convexoptimization.com/Meboo.html> (accessed on 10 October 2022).
30. Schölkopf, B.; Smola, A.J.; Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **1998**, *10*, 1299–1319. [[CrossRef](#)]
31. Huang, H.C.; Chuang, Y.Y.; Chen, C.S. Multiple kernel fuzzy clustering. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 120–134. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.