One-step Multiple Kernel k−means Clustering Based on Block Diagonal Property

Cuiling Chen
Guangxi Normal University

Jian Wei
Guangxi Normal University

Zhi Li (zhili@gxnu.edu.cn)
Guangxi Normal University

Research Article

Keywords: Multiple kernel clustering, Kernel k-means, Block diagonal property, One-step clustering

Posted Date: January 6th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2437949/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.
One-step Multiple Kernel $k-$means Clustering Based on Block Diagonal Property

Cuiling Chen$^1$, Jian Wei$^2$ and Zhi Li$^1$*

$^1$School of Computer Science and Engineering, Guangxi Normal University, 15 Yucai Road, Guilin, 541004, Guangxi, China.

$^2$Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, 15 Yucai Road, Guilin, 541004, Guangxi, China.

*Corresponding author(s). E-mail(s): zhili@gxnu.edu.cn; Contributing authors: mathchen@163.com; jweimath@163.com;

Abstract

Multiple kernel k-means clustering (MKKC) can efficiently incorporate multiple base kernels to generate an optimal kernel. However, many existing MKKC methods all need two-step operation: learning clustering indicator matrix and performing clustering on it. These results are not necessarily optimal because the optimal values of two steps are not equivalent to those of original problem. To address this issue, in this paper we propose a novel algorithm named one-step multiple kernel k-means clustering based on block diagonal property(OSMKKC-BD). By imposing a block diagonal constraint on the product of indicator matrix and its transpose matrix and enhancing its block diagonalization, this algorithm can promote the indicator matrix to get explicit clustering indicator, so as to implement one-step clustering. Furthermore, a simple kernel weighting strategy is used to obtain an optimal kernel, and boosts the quality of optimal kernel. In addition, a three-step iterative algorithm is designed to solve the corresponding optimization problem, in which the Riemann conjugate gradient iterative method is used to solve the optimization problem about the indicator matrix. Finally, by extensive experiments on ten real data sets and comparison of clustering results with the state-of-the-art MKKC methods, it is concluded that OSMKKC-BD is effective.

Keywords: Multiple kernel clustering · Kernel k-means · Block diagonal property · One-step clustering
1 Introduction

As a powerful tool for mining the potential structure of data, clustering is widely used in data mining and data processing. During the past decades, many clustering methods have been proposed, such as $k$-means clustering (KM) [1], spectral clustering (SC) [2], clustering with adaptive neighbors (CAN) [3], and their variants, etc. As a method with conceptual simplicity, easy-implementation and high efficiency, KM stands out from these methods. However, most of the data in real problems are nonlinear structures, and $k$-means cannot capture this structure. In order to cluster these data efficiently, a kernel version of KM is put forward [4].

The clustering performance of kernel methods depends on the selected kernel to a large extent, so it is particularly critical to select the appropriate kernel. In order not to be affected by a single kernel, multiple kernel $k$-means clustering (MKKC) has been proposed. It can effectively incorporate multiple base kernels to generate an optimal kernel. Most existing MKKC algorithms usually need two-step processes: 1) obtaining the clustering indicator matrix; 2) performing $k$-means clustering on this matrix [5]. This two-step operation can effectively handle clustering tasks. However, they can not ensure that the ultimate clustering results are optimal. This is because the optimal results of the two-step strategy are obtained by solving the optimal problem for each step, and they are not equivalent to those of original problem.

To address this issue, we hope to propose a MKKC algorithm for one-step clustering. However, how to realize one-step clustering?

As is known to all, each row of an ideal clustering indicator matrix has only one non-zero element, so we can judge which category the corresponding sample belongs to. At this time, the product of the indicator matrix and its transpose is in block diagonal (BD) form [6]. But in practical application, usually the indicator matrix cannot present the ideal state, and the product of it and its transpose does not have a BD structure. Therefore, it is necessary to perform the clustering task on the indicator matrix to obtain the clustering results. This is also the reason why most MKKC algorithms need to adopt a two-step strategy, and their shortcomings have been mentioned above. Nevertheless, we can assume the reverse: if the product of the indicator matrix and its transpose has a BD structure, the clustering indicator matrix can be promoted to approach the ideal state. Thus the clustering results can be obtained directly from the indicator matrix and one-step clustering can be realized.

Inspired by this idea, by using the BD representation method in [7], we put a BD constraint on the product of cluster indicator matrix and its transpose, and propose a novel algorithm named one-step multiple kernel $k$-means clustering based on BD property (OSMKKC-BD).

Here, we briefly introduce the main contributions of this paper.

• By using the Laplace matrix, we get the BD representation of the product of the indicator matrix and its transpose. This BD regularization term can
promote the indicator matrix to produce explicit clustering results, so that the clustering task can be completed in one step.

- A strategy of calculating kernel weights is presented to obtain an optimal kernel, which automatically adjusts important kernels to be assigned larger weight than unimportant kernels, and vice versa. This avoids the difficulty of choosing suitable base kernels and tuning kernel weights.
- The Riemann conjugate gradient iterative method in [6] is combined in the three-step iterative algorithm, which effectively overcomes the difficulty that the algorithm model is not easy to solve.
- Extensive experiments are conducted on ten benchmark data sets, and the effectiveness of OSMKKC-BD is on full display from the comparison of the experimental results with six state-of-the-art MKKC methods.

The paper consists of six sections. In Section II, the related works of MKKC method are briefly over-viewed. Section III presents the methodology of KKC and MKKC, and then introduces the proposed OSMKKC-BD algorithm in detail. Section IV is the optimization algorithm of OSMKKC-BD. Section V shows the experimental results under the given experimental settings and some discussions on experimental results. The conclusions are presented in Section VI.

2 Related Works

To capture non-linear structure of data, kernel $k$-means method (KKM) [8] extends KM into kernel space via kernel trick and performs clustering on arbitrary shape. The clustering performance based on kernel trick is closely relevant to selected kernels. However, how to determine an optimal kernel for a specific clustering in a real problem is usually unknown in advance, and it is very difficult. Multiple kernel $k$-means (MKKM) incorporates the information of multiple kernels by extending the fuzzy KM to a multiple kernel setting and avoids the difficulty of selecting the single kernel [9]. Further, robust MKKM (RMKKM) proposes an extension of MKKM by using $\ell_{2,1}$ norm in kernel space [10], which shows stronger robustness than MKKM since it can resist the interference of noise and outliers.

In recent years, many MKKC algorithms focus on how to generate an optimal kernel. A linear combination of all the base kernels [10–12] is used to generate an optimal kernel. In particular, a min-max model about the kernel coefficient and indicator matrix is given in a simple MKKC algorithm (SimpleMKKM) [11]. Its further version, i.e., localized SimpleMKKM [12], selects the combination of some base kernels to produce the consensus kernel. However, the ideal optimal kernel might not be in the linear combination. That is to say, this combination representation limits the search area of optimal kernel and reduces its representation capability.

Neighborhood kernel strategy [13, 14] allows the optimal kernel selected from the neighborhood of a linear combination of base kernels. It expands the
search area of optimal kernel and boosts its representation capability. Furthermore, it is verified that existing MKKC algorithms are the special cases in [13]. And the local density of individual samples is considered in [14], which makes it different from many existing MKKC algorithms. However, they usually use all the base kernels to learn the optimal kernel, which could lead to the redundance of base kernels.

As we see from [14], on account of considering the correlation among base kernels, the technique of local kernels is used to reduce the redundance of base kernels and promote the diversity of the selected kernels, and then the clustering performance is improved. Similar techniques can be also seen in [15], where the correlation between all kernel pairs is measured by the designed matrix-induced regularization, then MKKC with matrix-induced regularization (MKKM-MR) is proposed. In [16], MKKC by selecting representative kernels (MKKM-RK) is proposed by a strategy to select the representative kernels. It incorporates the process of selecting diverse subsets from the predefined kernels into the multiple k-means clustering framework, thus reduces the redundance of base kernels and enhances the high quality of optimal kernel.

Kernel alignment strategy [6, 17] is often used to preserve the internal structure of data and to enhance the clustering performance. Recently, in order to combine the advantages of both MKKC and SC and overcome their disadvantage, Ren et al. [18] proposes a MKKC algorithm coupled graph tensor learning, then builds a bridge between MKKC and SC. And in some practical cases, such as disease prediction or discrimination [19, 20], some views of some samples may be lost or missing, then the corresponding rows and columns of base kernels are absent. This leads to difficulty in getting good clustering results. A remedy for this issue [21–23] is to impute the absent part of base kernels and then obtain the optimal kernel on the imputed kernels.

As we see that lots of strategies have been proposed in MKKC methods to address different problems from different aspects and obtain good clustering results. However, among the above MKKC methods, most of them suffer from one drawback in common, i.e., the optimization procedures contain two stages, learning the indicator matrix and performing clustering on this matrix. The shortcomings of this two-step operation have been mentioned in Section 1.

3 Methodology

3.1 Notations

For convenience, the details of notations in this paper are listed in Table 1.

3.2 Kernel $k$–means Clustering (KKC)

Let $X = \{x_i\}_{i=1}^n$ be a set of samples, and $\phi(\cdot): \mathcal{X} \to \mathcal{H}$ be a kernel mapping from an original space $\mathcal{X}$ to a reproducing Hilbert space $\mathcal{H}$. Kernel $k$–means
One-step Multiple Kernel $k-$means Clustering Based on Block Diagonal Property

Table 1 Details of Notations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_i$</td>
<td>the $i$-th column of $A$</td>
</tr>
<tr>
<td>$A_{ij}$</td>
<td>the $i,j$-th entry of $A$</td>
</tr>
<tr>
<td>$|A|_F$</td>
<td>Frobenius norm of $A$, i.e., $A_F = \sqrt{\sum_{i,j} A_{i,j}^2}$</td>
</tr>
<tr>
<td>$A^T$</td>
<td>transpose of $A$</td>
</tr>
<tr>
<td>$tr(A)$</td>
<td>trace of $A$</td>
</tr>
<tr>
<td>$Diag(A)$</td>
<td>diagonal matrix with diagonal elements of $A$</td>
</tr>
<tr>
<td>$A \succeq 0$</td>
<td>positive semi-definite $A$</td>
</tr>
<tr>
<td>$I_k$</td>
<td>$k$ order identity matrix</td>
</tr>
<tr>
<td>$1_v$</td>
<td>all-one column vector</td>
</tr>
<tr>
<td>$1_M$</td>
<td>all-one matrix</td>
</tr>
</tbody>
</table>

Clustering (KKC) is usually expressed as

$$\min_{Z \in \{0,1\}^{n \times k}} \sum_{i=1}^{n} \sum_{c=1}^{k} Z_{ic} \| \phi(x_i) - \mu_c \|^2$$

subject to

$$\sum_{c=1}^{k} Z_{ic} = 1,$$

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

$$\mu_c = \frac{1}{n_c} \sum_{i=1}^{n} Z_{ic} \phi(x_i), \quad n_c = \sum_{i=1}^{n} Z_{ic},$$

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

By denoting the design matrix and the centroid matrix as $\Phi = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)] \in \mathbb{R}^{d \times n}$ and $U = [\mu_1, \mu_2, \ldots, \mu_k] \in \mathbb{R}^{d \times k}$, respectively, problem (1) is as follows

$$\min_{Z \in \{0,1\}^{n \times k}} Tr((\Phi - UZ^T)^T(\Phi - UZ^T))$$

subject to

$$Z \cdot 1_k = 1_n.$$  \hspace{1cm} (3)

Letting $L = \text{diag}([n_1^{-1}, n_2^{-1}, \cdots, n_k^{-1}])$, then $U = \Phi Z L$. Noting that $Z^T Z = L^{-1}$, thus

$$Tr((\Phi - UZ^T)^T(\Phi - UZ^T)) = Tr(K - KZLZ^T),$$

where $K$ is a kernel matrix with $k_{ij} = \phi(x_i)^T \phi(x_j)$. Then the problem (3) turns into

$$\min_{Z \in \{0,1\}^{n \times k}} Tr(K - KZLZ^T)$$

subject to

$$Z \cdot 1_k = 1_n.$$  \hspace{1cm} (4)

According to matrix decomposition, problem (4) becomes

$$\min_{Z \in \{0,1\}^{n \times k}} Tr(K - L^{\frac{1}{2}}Z^T KZL^{\frac{1}{2}})$$

subject to

$$Z \cdot 1_k = 1_n.$$  \hspace{1cm} (5)
We note that $Z$ in (5) is discrete, then it is difficult to solve (5). The usual way to overcome this difficulty is to relax $Z$ to arbitrary real values and take approximate value of discrete $Z$. Specifically, letting $H = ZL^{1/2}$, a relaxed form of (5) is obtained:

$$\min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(K(I_n - HH^T))$$

s.t. $H^T H = I_k,$

where $H \in \mathbb{R}^{n \times k}$. The orthogonality constraint $H^T H = I_k$ on $H$ is derived from $H = ZL^{1/2}$ and $Z^T Z = L^{-1}$. The optimal $H$ for (6) is made up of the $k$ eigenvectors corresponding to the $k$ largest eigenvalues of $K$.

### 3.3 Multiple Kernel $k-$means Clustering (MKKC)

In the scenario of multiple kernel clustering, each sample is represented as $\phi_w(x) = [w_1 \phi_1(x)^T, w_2 \phi_2(x)^T, \cdots, w_m \phi_m(x)^T]^T$ via a group of kernel mappings $\{\phi_p(\cdot)\}_{p=1}^m$, where $w_p$ of $w = [w_1, w_2, \cdots, w_m]^T$ corresponds to the coefficient of $p-$th base kernel. Usually, $w$ is learned by optimizing in clustering process. Thus, the kernel function of kernel mappings can be computed by

$$k_w(x_i, x_j) = \phi_w(x_i)^T \phi_w(x_j) = \sum_{p=1}^m w_p^2 k_p(x_i, x_j).$$

As a substitute for the kernel matrix $K$ in (6), $K_w$ computed via (7) is used for MKKC, thus the optimization problem is as follows:

$$\min_{H \in \mathbb{R}^{n \times k}, w \in \mathbb{R}^m_+} \text{Tr}(K_w(I_n - HH^T))$$

s.t. $H^T H = I_k, w^T 1_m = 1.$

The solution of problem (8) is obtained by updating $H$ and $w$ alternately: Optimizing $H$ with fixed $w$ by solving the same problem as that in (6); Optimizing $w$ with fixed $H$ via solving the quadratic programming problem:

$$\min_{w \in \mathbb{R}^m_+} \sum_{p=1}^m w_p^2 \text{Tr}(K_p(I_n - HH^T))$$

s.t. $w^T 1_m = 1.$

### 3.4 One-step Multiple Kernel $k-$means Clustering Based on Block Diagonal Property (OSMKKC-BD)

#### 3.4.1 Block Diagonal Regularizer

We know that $H$ is a clustering indicator matrix. According to the idea of Laplacian matrix in graph theory, we view the square matrix $HH^T$ as an adjacency matrix, then the degree matrix $D$ of $HH^T$ is a diagonal matrix, of which the diagonal element is $d_{ii} = \sum_{c=1}^k (HH^T)_{ic}$, i.e.,

$$D = \text{Diag}(HH^T \cdot 1_v),$$
then
\[ L_{HH^T} = \text{Diag}(HH^T \cdot 1_v) - HH^T. \] (10)

The following theorem reveals the relationship between a matrix and its Laplacian matrix.

**Theorem 1** [7] For any \( A \in \mathbb{R}^{n \times n} \succeq 0 \), the number of connected components (blocks) in \( A \) equals the multiplicity \( k \) of the eigenvalue 0 of the corresponding Laplacian matrix \( L_A \).

For any matrix \( A \in \mathbb{R}^{n \times n} \succeq 0 \), let \( \lambda_i(A) \) represent the eigenvalues of \( A \) in decreasing order, then \( A \succeq 0 \) and \( \lambda_i(A) \geq 0 \) for all \( i \). Therefore, \( A \) has \( k \) connected components by Theorem 1 if and only if

\[
\lambda_i(A) \begin{cases} 
> 0, & i = 1, \ldots, n - k, \\
= 0, & i = n - k + 1, \ldots, n.
\end{cases} \] (11)

Therefore, the \( k \)-block diagonal representation of \( HH^T \) can be defined as follows.

**Definition 1** [7] (\( k \)-block diagonal representation) For the matrix \( HH^T \in \mathbb{R}^{n \times n} \), the \( k \)-block diagonal representation is defined as the sum of the \( k \) smallest eigenvalues of \( L_{HH^T} \), i.e.,

\[
\|HH^T\|_k = \sum_{i=n-k+1}^{n} \lambda_i(L_{HH^T}). \] (12)

It can be seen from Definition 1 that \( \|HH^T\|_k = 0 \) is equivalent to that the matrix \( HH^T \) is \( k \)-block diagonal, hence \( \|HH^T\|_k \) can be regarded as the BD induced regularizer. It can control the number of blocks and enhance \( HH^T \) to be BD directly [7]. Further, it is softer and more flexible than the hard constraint in [24]. In addition, \( \|HH^T\|_k \) is superior to the alternative of Rank \( (L_{HH^T}) \) or the convex relaxation \( \|L_{HH^T}\|_* \) [7].

**3.4.2 Kernel Weight Strategy**

One strategy of Multiple kernel learning (MKL) is to tune the weight of every base kernel automatically and learn an optimal consensus kernel. The MKL model in [25] is given by

\[
\min_K \sum_{p=1}^{m} w_p \|K_p - K\|^2_{F}, \] (13)

where \( m \) is the number of base kernels, \( K_p \) is the \( p \)-th base kernel, \( K \) is the consensus kernel, \( w_p \) is the weight value of \( p \)-th base kernel \( K_p \).
Here we propose a weight strategy as follows:

\[ w_p = \frac{1}{m-1} \left( 1 - \frac{\|K_p - K\|_F^2}{\sum_{p=1}^m \|K_p - K\|_F^2} \right). \tag{14} \]

In (14), \( w_p \) has the following characteristics: 1) It is decreasing monotonically with respect to \( \|K_p - K\|_F^2 \). This ensures it is large when the contribution of \( K_p \) to \( K \) is great, vice versa. 2) \( 0 \leq w_p \leq 1 \) and \( \sum_{p=1}^m w_p = 1 \). This shows that \( w_p \) can well balance the contribution of each base kernel \( K_p \) to learn \( K \). 3) It is simpler than the formula of \( w_p \) in [25]:

\[ w_p = \exp \left( -\frac{\delta e_p}{\bar{e}} \right) / \sum_{p=1}^m \exp \left( -\frac{\delta e_p}{\bar{e}} \right), \tag{15} \]

where \( \delta \) is a scalar, \( e_p = \|K_p - K\|_F^2 \), and \( \bar{e} \) is the mean value of \( [e_1, e_2, \ldots, e_m]^T \) (i.e., \( \bar{e} = \frac{1}{m} \sum_{p=1}^m e_p \)).

### 3.4.3 Objective Function

Combining the block diagonal regularizer of (12) and the kernel weight strategy of (14), the final objective function of OSMKKC-BD is as follows:

\[
\begin{align*}
\min_{H,K} & \quad \text{Tr}(K(I_n - HH^T)) + \alpha \sum_{p=1}^m w_p \|K_p - K\|_F^2 + \beta \|HH^T\|_F^4 \\
\text{s.t.} & \quad H^T H = I_k, H \in \mathbb{R}^{n \times k},
\end{align*}
\tag{16}
\]

where \( K = \sum_{p=1}^m w_p K_p \).

### 4 Optimization Algorithm

A disadvantage of the problem (16) is that the regularizer \( \|HH^T\|_F^4 \) is non-convex, then how to solve (16) is a challenge. For this, we introduce a theorem to reformulate \( \|HH^T\|_F^4 \) by Ky Fan.

**Theorem 2** [[26], p.515] Let \( L \in \mathbb{R}^{n \times n} \) and \( L \succeq 0 \). Then

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

From Definition 1 and Theorem 2, it is easy to obtain the following convex program:

\[
\|HH^T\|_F^4 = \min_{W} \langle L_{HH^T}, W \rangle, \quad \text{s.t.} \quad 0 \preceq W \preceq I, Tr(W) = k.
\]
Because of \( \langle L, W \rangle = Tr(L^T W) \) and the symmetry of \( \text{Diag}(HH^T \cdot 1_v) - HH^T \), then (16) is equivalent to
\[
\min_{H,K,W} \quad Tr(K(I_n - HH^T)) + \alpha \sum_{p=1}^{m} w_p \|K_p - K\|_F^2 + \beta Tr((\text{Diag}(HH^T \cdot 1_v) - HH^T)W)
\]
\[
s.t. \quad H^TH = I_k, H \in \mathbb{R}^{n \times k}, 0 \preceq W \preceq I, Tr(W) = k.
\]
(17)

It is evident that (17) is not jointly convex on \( H, K \) and \( W \). However, for each variable, it is convex when the other variables are fixed. Thus for (17), we adopt the alternative optimization strategy, i.e., each variable is iteratively optimized with the others fixed, until convergence.

(1) Update \( W \) while fixing \( H \) and \( K \)

While \( H \) and \( K \) are fixed, problem (17) becomes
\[
\min_{W} \quad Tr((\text{Diag}(HH^T \cdot 1_v) - HH^T)W)
\]
\[
s.t. \quad 0 \preceq W \preceq I, Tr(W) = k.
\]
(18)

For (18), \( W^{k+1} = GG^T \), where \( G \in \mathbb{R}^{n \times k} \) is composed of the \( k \) eigenvectors associated with the \( k \) smallest eigenvalues of \( \text{Diag}(HH^T \cdot 1_v) - HH^T \). [7]

(2) Update \( K \) while fixing \( W \) and \( H \)

With fixed \( W \) and \( H \), the optimization problem from (17) is as follows:
\[
\min_{K} \quad Tr(K(I_n - HH^T)) + \alpha \sum_{p=1}^{m} w_p \|K_p - K\|_F^2.
\]
(19)
The closed-form solution of \( K \) is solved by taking the derivative of (19) w.r.t. \( K \) to zero:
\[
K = \frac{(HH^T - I_n) + 2\alpha \sum_{p=1}^{m} w_p K_p}{2\alpha \sum_{p=1}^{m} w_p}.
\]
(20)

(3) Update \( H \) while fixing \( W \) and \( K \)

Here, the optimization subproblem from (17) is
\[
\min_{H} \quad -Tr(K \cdot HH^T) + \beta Tr((\text{Diag}(HH^T \cdot 1_v) - HH^T)W)
\]
\[
s.t. \quad H^TH = I_k, H \in \mathbb{R}^{n \times k}.
\]
(21)

It is difficult to solve (21) directly. By the properties of matrix operations and its trace, \( Tr(Diag(HH^T \cdot 1_v) \cdot W) = Tr((1_M \text{Diag}(W)) \cdot HH^T) \), (21) can be changed into
\[
\min_{H} \quad Tr(\beta(1_M \text{Diag}(W) - W - K) \cdot HH^T)
\]
\[
s.t. \quad H^TH = I_k, H \in \mathbb{R}^{n \times k}.
\]
(22)

Because \( 1_M \text{Diag}(W) \) is not a symmetric matrix, (22) cannot be solved as a kernel \( k \)-means clustering problem. Noting it is similar to the optimization problem on Stiefel manifold in [27], thus we adopt the Riemann conjugate gradient method proposed in [27] to solve (22).

Therefore, the main steps of our proposed algorithm are as follows.
Algorithm 1 Pseudo code of solving problem (16)

1: Input: $m$ base kernels $\{K_p\}_{p=1}^{m}$ and parameters $\alpha, \beta$.
2: Initialize: $(K)^1 = \frac{1}{m} \sum_{p=1}^{m} K_p, \{(w_p)^1\}_{p=1}^{m} = \frac{1}{m}$.
3: While not converged do.
4: (1) Update $W^{k+1}$ by solving (18).
5: (2) Update $K^{k+1}$ by solving (20).
6: (3) Update $H^{k+1}$ via (22), compute $w$ via (14).
7: end while
8: Obtain the optimal $W^*, H^*, K^*$.
9: Output: ACC, NMI and Purity.

5 Experimental Analysis

In this section, the used data sets, comparison method, evaluation metrics and multiple kernels’ construction are listed. And under these given experimental settings, the experimental results and some analysis are presented, which demonstrate the effectiveness of our proposed method.

5.1 Data sets

In order to evaluate the effectiveness of OSMKKC-BD, we choose ten real data sets from the public websites \(^1\) \(^2\) \(^3\). They have different sizes and classes, and their summaries can be seen in Table 2.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>#(Samples)</th>
<th>#(Features)</th>
<th>#(Classes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>840</td>
<td>768</td>
<td>120</td>
</tr>
<tr>
<td>BA</td>
<td>1404</td>
<td>320</td>
<td>36</td>
</tr>
<tr>
<td>COIL20</td>
<td>1440</td>
<td>1024</td>
<td>20</td>
</tr>
<tr>
<td>GLIOMA</td>
<td>50</td>
<td>4434</td>
<td>4</td>
</tr>
<tr>
<td>LYMPHOMA</td>
<td>96</td>
<td>4026</td>
<td>8</td>
</tr>
<tr>
<td>MSRA25</td>
<td>1799</td>
<td>256</td>
<td>12</td>
</tr>
<tr>
<td>ORL</td>
<td>400</td>
<td>1024</td>
<td>40</td>
</tr>
<tr>
<td>TR11</td>
<td>414</td>
<td>6429</td>
<td>9</td>
</tr>
<tr>
<td>USPS</td>
<td>1854</td>
<td>256</td>
<td>10</td>
</tr>
<tr>
<td>YALE</td>
<td>165</td>
<td>1024</td>
<td>15</td>
</tr>
</tbody>
</table>

5.2 Comparison Algorithms

This paper focuses on MKKC method. In order to demonstrate its clustering performance, we compare OSMKKC-BD with six MKKC algorithms.

\(^1\)http://archive.ics.uci.edu/ml/,
\(^2\)http://featureselection.asu.edu/datasets.php,
\(^3\)http://trec.nist.gov
Among of them, KKM, MKKM, RMKKM are three classic MKKC algorithms, MKKM-MR, SimpleMKKM and MKKM-RK are the state-of-the-art MKKC algorithms recently proposed.

- KKM is a kernel version of KM to handle nonlinear data of principal component analysis via using integral operator kernel functions [8].
- MKKM extends the fuzzy KM into a MKL setting, where kernel weights are automatically adjusted to incorporate multiple kernels [9].
- RMKKM is an extension of MKKM, which is a robust MKKM by using ℓ2,1 norm in kernel space [10].
- MKKM-MR is a MKKC method with matrix-induced regularization, where the correlation between all kernel pairs is measured via this regularization [15].
- SimpleMKKM is a MKKC algorithm by optimizing a min-max problem, which minimizes kernel alignment on kernel coefficient and maximizes it on clustering matrix, until gets clustering results. [11].
- MKKM-RK is a MKKC algorithm by utilizing a representative-kernels regularizer to choose diverse subsets of kernels from predefined base kernels and to generate the optimal combination of kernels [16].

### 5.3 Evaluation Metrics

We adopt the widely used metrics, i.e., clustering accuracy (ACC), normalized mutual information (NMI), and Purity, to quantitatively evaluate the performance of our proposed OSMKKC-BD and the comparison methods.

Usually, ACC, NMI and Purity are the percentage of correctly clustered data to the total data, the measure of the similarity between the truth labels and the predicted labels, and the percentage of the correctly clustered data contained in each cluster. Their calculated formulas are given below:

\[
ACC = \frac{N_{\text{correct}}}{N}, \quad NMI = \frac{H(A) + H(B)}{\max(H(A), H(B))}, \quad Purity = \sum_{i=1}^{k} \left( \frac{M_i}{N} \right) P_i,
\]

where \(N\) is the total amount of all clustered data, \(N_{\text{correct}}\) is the amount of correctly clustered data;

\[
H(A) = -\sum_{a} P_A(a)\log P_A(a), \quad H(B) = -\sum_{b} P_B(b)\log P_B(b),
\]

\[
H(A, B) = -\sum_{a,b} P_{AB}(a, b)\log P_{AB}(a, b),
\]

where \(k\) is the number of clusters, \(M_i\) and \(P_i\) correspond to the amount of data and the distribution of correctly clustered data in the \(i\)-th cluster, respectively.

The larger value of ACC, NMI and Purity means better performance. The detailed information on them can be found in [28] and [29].
5.4 Experiment Setup

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

The number of clusters \( k \) is set to the true number of cluster in each data set. The two parameters \( \alpha, \beta \) are selected from \([10^{-3}, 10^{-2}, \cdots, 10^2, 10^3]\). On one hand, to avoid the randomness of sampling, the ten-fold cross-validation method embedded the five-fold cross-validation in our experiment is used to test the effectiveness of OSMKKC-BD. That is, we randomly divided all the samples into ten subsets without repetition, nine of which are used as training sets and the other one as a test set. The nine training sets are further divided into five subsets, four of which are used as training sets and the rest as a validation set. The purpose of the five-fold cross-validation method is to train the model and determine the optimal combination of hyper-parameters \( \alpha, \beta \) from \([10^{-3}, 10^{-2}, \cdots, 10^2, 10^3]\). Then the determined combination of \( \alpha, \beta \) is used in the test set to perform clustering. In the other hand, for the comparison methods, we set the parameters according to the corresponding literatures.

5.5 Experimental Results and Analysis

The average ACC, NMI and Purity of all the methods are reported in Table 3. The last three rows of Table 3 correspond to the mean ACC, NMI and Purity of each method on all the data sets. The best results of ACC, NMI and Purity on each data set are highlighted in boldface.

Table 3 shows that the proposed OSMKKC-BD performs best. More precise analysis is as follows. First, as a MKKC method, OSMKKC-BD surpasses KKM and MKKM by far and improves by 52.31\%, 61.65\%, 54.68\% and 63.26\%, 65.99\%, 63.13\%, respectively, in terms of ACC, NMI and Purity. Second, OSMKKC-BD outperforms RMKKM by 17.49\%, 18.09\%, 18.76\% according to the three evaluation metrics. Third, as we can see from Table 3 that the three state-of-the-art MKKC methods recently proposed, i.e., MKKM-MR, SimpleMKKM and MKKM-RK, all obtain good cluster results, but OSMKKC-BD transcends all of them by more than 13\% whether from ACC, NMI or Purity. This shows that OSMKKC-BD obtain better clustering performance than MKKM-MR, SimpleMKKM and MKKM-RK.

In addition, we illustrate the block diagonal property of \( HH^T \) from MSRA25 and USPS data sets by OSMKKC-BD. As shown in Figs. 1 and 2, our proposed algorithm encourage to obtain \( HH^T \) with block diagonal property.

Overall, the experiment results show that OSMKKC-BD is effective.
Table 3  Clustering results of different methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>KKM</th>
<th>MKKM</th>
<th>RMKKM</th>
<th>SimpleMKKM</th>
<th>MKKM&lt;sub&gt;<em>MR</em>&lt;/sub&gt;</th>
<th>MKKM&lt;sub&gt;<em>RK</em>&lt;/sub&gt;</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>0.3000</td>
<td>0.1286</td>
<td>0.3168</td>
<td>0.4863</td>
<td>0.3995</td>
<td>0.5047</td>
<td>0.6656</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.6360</td>
<td>0.4163</td>
<td>0.6608</td>
<td>0.7615</td>
<td>0.7119</td>
<td>0.7608</td>
<td>0.8829</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.3190</td>
<td>0.1298</td>
<td>0.3558</td>
<td>0.5398</td>
<td>0.4162</td>
<td>0.5305</td>
<td>0.7806</td>
</tr>
<tr>
<td>AR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.2803</td>
<td>0.3808</td>
<td>0.4088</td>
<td>0.4177</td>
<td>0.4496</td>
<td>0.3708</td>
<td>0.4186</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4365</td>
<td>0.5301</td>
<td>0.5639</td>
<td>0.5882</td>
<td>0.5919</td>
<td>0.5194</td>
<td>0.6702</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.3226</td>
<td>0.4010</td>
<td>0.4329</td>
<td>0.4619</td>
<td>0.4780</td>
<td>0.3962</td>
<td>0.4652</td>
</tr>
<tr>
<td>BA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.5514</td>
<td>0.3076</td>
<td>0.6390</td>
<td>0.6375</td>
<td>0.6387</td>
<td>0.6127</td>
<td>0.6435</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.6328</td>
<td>0.4681</td>
<td>0.7616</td>
<td>0.7677</td>
<td>0.7598</td>
<td>0.7281</td>
<td>0.7763</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.5986</td>
<td>0.3229</td>
<td>0.6750</td>
<td>0.6667</td>
<td>0.6904</td>
<td>0.6300</td>
<td>0.7018</td>
</tr>
<tr>
<td>COIL20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.5032</td>
<td>0.4880</td>
<td>0.5760</td>
<td>0.4955</td>
<td>0.5120</td>
<td>0.5640</td>
<td>0.7286</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.3256</td>
<td>0.2943</td>
<td>0.4818</td>
<td>0.3083</td>
<td>0.2957</td>
<td>0.4077</td>
<td>0.6276</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.5357</td>
<td>0.5400</td>
<td>0.6460</td>
<td>0.5341</td>
<td>0.5320</td>
<td>0.5787</td>
<td>0.7743</td>
</tr>
<tr>
<td>GIIOMA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.4982</td>
<td>0.5085</td>
<td>0.6135</td>
<td>0.5437</td>
<td>0.5932</td>
<td>0.5639</td>
<td>0.6790</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.5105</td>
<td>0.5070</td>
<td>0.6172</td>
<td>0.6495</td>
<td>0.6099</td>
<td>0.5963</td>
<td>0.6842</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.7163</td>
<td>0.7036</td>
<td>0.8031</td>
<td>0.7826</td>
<td>0.8266</td>
<td>0.8125</td>
<td>0.8170</td>
</tr>
<tr>
<td>LYMPHOMA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.4243</td>
<td>0.3385</td>
<td>0.5526</td>
<td>0.5323</td>
<td>0.5552</td>
<td>0.5570</td>
<td>0.5890</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4222</td>
<td>0.3516</td>
<td>0.6193</td>
<td>0.6030</td>
<td>0.6262</td>
<td>0.6031</td>
<td>0.6809</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.4600</td>
<td>0.3424</td>
<td>0.5790</td>
<td>0.5563</td>
<td>0.5875</td>
<td>0.5722</td>
<td>0.6455</td>
</tr>
<tr>
<td>MSRA25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.4308</td>
<td>0.3475</td>
<td>0.5521</td>
<td>0.6357</td>
<td>0.6391</td>
<td>0.5860</td>
<td>0.6985</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.6383</td>
<td>0.5378</td>
<td>0.7406</td>
<td>0.8163</td>
<td>0.8073</td>
<td>0.7581</td>
<td>0.8877</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.4797</td>
<td>0.3525</td>
<td>0.6001</td>
<td>0.6908</td>
<td>0.6860</td>
<td>0.6188</td>
<td>0.8040</td>
</tr>
<tr>
<td>ORL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.3213</td>
<td>0.3237</td>
<td>0.5605</td>
<td>0.5005</td>
<td>0.5722</td>
<td>0.5564</td>
<td>0.5731</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.5551</td>
<td>0.1763</td>
<td>0.5499</td>
<td>0.4199</td>
<td>0.5664</td>
<td>0.5764</td>
<td>0.5989</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.3551</td>
<td>0.4517</td>
<td>0.7217</td>
<td>0.5461</td>
<td>0.7609</td>
<td>0.7799</td>
<td>0.7254</td>
</tr>
<tr>
<td>TR11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.4260</td>
<td>0.5119</td>
<td>0.6515</td>
<td>0.6357</td>
<td>0.5862</td>
<td>0.6456</td>
<td>0.6512</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4085</td>
<td>0.4856</td>
<td>0.6351</td>
<td>0.6351</td>
<td>0.6103</td>
<td>0.6045</td>
<td>0.6697</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.4983</td>
<td>0.5912</td>
<td>0.7324</td>
<td>0.7194</td>
<td>0.7090</td>
<td>0.7155</td>
<td>0.7192</td>
</tr>
<tr>
<td>USPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.4182</td>
<td>0.3515</td>
<td>0.5218</td>
<td>0.5341</td>
<td>0.5512</td>
<td>0.5814</td>
<td>0.6888</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4330</td>
<td>0.4152</td>
<td>0.5558</td>
<td>0.5614</td>
<td>0.5826</td>
<td>0.5879</td>
<td>0.8271</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.4424</td>
<td>0.3636</td>
<td>0.5364</td>
<td>0.5495</td>
<td>0.5555</td>
<td>0.5899</td>
<td>0.7656</td>
</tr>
<tr>
<td>YALE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.4160</td>
<td>0.3881</td>
<td>0.5393</td>
<td>0.5419</td>
<td>0.5612</td>
<td>0.5543</td>
<td>0.6336</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4519</td>
<td>0.4401</td>
<td>0.6186</td>
<td>0.6111</td>
<td>0.6215</td>
<td>0.6142</td>
<td>0.7305</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.4728</td>
<td>0.4413</td>
<td>0.6062</td>
<td>0.6047</td>
<td>0.6356</td>
<td>0.6224</td>
<td>0.7199</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACC</td>
<td>0.4160</td>
<td>0.3881</td>
<td>0.5393</td>
<td>0.5419</td>
<td>0.5612</td>
<td>0.5543</td>
<td>0.6336</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4519</td>
<td>0.4401</td>
<td>0.6186</td>
<td>0.6111</td>
<td>0.6215</td>
<td>0.6142</td>
<td>0.7305</td>
</tr>
<tr>
<td></td>
<td>Purity</td>
<td>0.4728</td>
<td>0.4413</td>
<td>0.6062</td>
<td>0.6047</td>
<td>0.6356</td>
<td>0.6224</td>
<td>0.7199</td>
</tr>
</tbody>
</table>

Fig. 1  Visualization of $HH^T$ on the MSRA25 data set from MKKM-MR, SimpleMKKM, MKKM-RK and OSMKKC-BD, respectively.

5.6 Ablation Study

In order to test the role of BD regular term of OSMKKC-BD during clustering, by removing the BD regular term, we only consider the following optimization...
One-step Multiple Kernel \(k\)-means Clustering Based on Block Diagonal Property

![Fig. 2](image)

**Fig. 2** Visualization of \(HH^T\) on the USPS data set from MKKM-MR, SimpleMKKM, MKKM-RK and OSMKKC-BD, respectively.

problem (OSMKKC-NoBD)

\[
\min_{H, K} \quad Tr(K(I_n - HH^T)) + \alpha \sum_{p=1}^{m} w_p \|K_p - K\|_F^2 \\
\text{s.t.} \quad H^T H = I_k, H \in \mathbb{R}^{n \times k}.
\]  

(23)

and perform ablation studies on the ten data sets. The clustering results are shown in Fig. 3, which indicates that the clustering results of OSMKKC-BD outperform those of OSMKKC-NoBD. Accordingly, the BD regular term in OSMKKC-BD improves the clustering performance.

5.7 Parameters’ Sensitivity

In OSMKKC-BD, the parameters \(\alpha\) and \(\beta\) needs to be set properly. In addition, we set the penalty parameter \(\gamma = 0.1\). In order to verify if OSMKKC-BD is sensitive to the parameters \(\alpha\) and \(\beta\), we tune \(\alpha\) and \(\beta\) in the ranges \([10^{-2}, 10^{-1}, \ldots, 10^1, 10^2]\) by leveraging a grid search technique. The clustering performance with different values of \(\alpha\) and \(\beta\) is presented in Fig. 4, which shows that our proposed OSMKKC-BD is data-driven.

5.8 Convergence

The stopping criteria of Algorithm 1 is set to \(\frac{\|\text{obj}(t+1) - \text{obj}(t)\|_2^2}{\text{obj}(t)} \leq 10^{-6}\), where \(\text{obj}(t)\) is the objective function value of (16) at the \(t\)-th iteration. And the trend of objective value in our proposed OSMKKC-BD method with respect to iterations is presented in Fig. 5. The curve graphs of Fig. 5 can show that the objective function of OSMKKC-BD decreases to convergence, and the convergence rate is relatively fast since it converges to the optimal value within 10 iterations on all the data sets.

6 Conclusions

In this paper, a novel MKKC algorithm called OSMKKC-BD is proposed by applying block diagonal constraints on the product of indicator matrix and its transpose. This algorithm can promote the indicator matrix to get explicit clustering results, so as to achieve one-step clustering. Furthermore, a simple kernel weight strategy is used to generate an optimal kernel, which promotes
the clustering performance. The experimental results on 10 data sets also fully demonstrate the effectiveness of OSMKKC-BD.

However, in this paper, we selected all the base kernels to produce an optimal kernel, which may lead to redundancy of base kernels, thus affecting the clustering effect. In the future, we will extend the block diagonal representation to the neighborhood and local multiple kernel clustering methods, obtain the clustering results directly from the indicator matrix and execute one-step clustering. In addition, the block diagonal representation in this paper can also be used to incomplete multiple kernel clustering method, and achieve one-step clustering.

Acknowledgments. This work is partially supported by Research fund of Guangxi Key Lab of Multi-source Information Mining and Security (Grant No: MIMS22-03, MIMS21-M-01), the National Natural Science Foundation of China (Grant No: 61862009), Guangxi ”Bagui Scholar” Teams for Innovation and Research.

Declarations

Conflicts of interests The authors declare that they have no conflict of interest or personal relationships related to the work in this paper.

References


One-step Multiple Kernel $k$-means Clustering Based on Block Diagonal Property


[18] Ren, Z., Sun, Q., Wei, D.: Multiple kernel clustering with kernel $k$-means
One-step Multiple Kernel $k$-means Clustering Based on Block Diagonal Property


Fig. 3 Comparison of clustering results between OSMKKC-BD and OSMKKC-NoBD.
Fig. 4 ACC of OSMKKC-BD with different parameter's settings.
Fig. 5  Objective function value of OSMKKC-BD at each iteration.