Computers, Materials & Continua
DOI:10.32604/cmc.2021.013389
Article

Tech Science Press

Trade-Off between Efficiency and Effectiveness: A Late Fusion Multi-View Clustering Algorithm

Yunping Zhao[1], Weixuan Liang[1], Jianzhuang Lu[1,*], Xiaowen Chen[1] and Nijiwa Kong[2]

[1]National University of Defence Technology, Changsha, China
[2]Department of Mathematics, Faculty of Science, University of Sargodha, Sargodha, Pakistan
[*]Corresponding Author: Jianzhuang Lu. Email: lujz1977@163.com
Received: 05 August 2020; Accepted: 12 September 2020

Abstract: Late fusion multi-view clustering (LFMVC) algorithms aim to integrate the base partition of each single view into a consensus partition. Base partitions can be obtained by performing kernel k-means clustering on all views. This type of method is not only computationally efficient, but also more accurate than multiple kernel k-means, and is thus widely used in the multi-view clustering context. LFMVC improves computational efficiency to the extent that the computational complexity of each iteration is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ (where n is the number of samples). However, LFMVC also limits the search space of the optimal solution, meaning that the clustering results obtained are not ideal. Accordingly, in order to obtain more information from each base partition and thus improve the clustering performance, we propose a new late fusion multi-view clustering algorithm with a computational complexity of $\mathcal{O}(n^2)$. Experiments on several commonly used datasets demonstrate that the proposed algorithm can reach quickly convergence. Moreover, compared with other late fusion algorithms with computational complexity of $\mathcal{O}(n)$, the actual time consumption of the proposed algorithm does not significantly increase. At the same time, comparisons with several other state-of-the-art algorithms reveal that the proposed algorithm also obtains the best clustering performance.

Keywords: Late fusion; kernel k-means; similarity matrix

1 Introduction

Many real-world datasets are naturally made up of different representations or views. Given that these multiple representations often provide compatible and complementary information, it is clearly preferable to integrate them in order to obtain better performance rather than relying on a single view. However, because data represented by different views can be heterogeneous and biased, the question of how to fully utilize the multi-view information in order to obtain a consensus representation for clustering purposes remains a challenging problem in the field of multi-view clustering (MVC). In recent years, various kinds of algorithms have been developed to provide better clustering performance or higher efficiency; these include multi-view subspace clustering [1–3], multi-view spectral clustering [4–6], and multiple kernel k-means (MKKM) [7–11]. Among these, MKKM has been widely utilized due to its excellent clustering performance and strong interpretability.

Depending on the stage of multi-view data fusion involved, the existing literature on MVC can be roughly divided into two categories. The first category, which can be described as 'early fusion', fuses the information of all views before clustering. Huang et al. [7] optimize an optimal kernel matrix as a linear combination of a set of pre-specified kernels for clustering purposes. To reduce the redundancy of the pre-defined kernels, Liu et al. [10] propose an MKKM algorithm with matrix-induced regularization. To further enhance the clustering performance, a local kernel alignment criterion is adopted for MKKM in [10]; moreover, the algorithm proposed in [11] allows the optimal kernel to be found in the neighbor of the combinational kernels, which increases the search space of the optimal kernel.

Although the above-mentioned early fusion algorithms have greatly improved the clustering accuracy of MKKM in different respects, they also have two drawbacks that should not be overlooked. The first of these is the more intensive computational complexity, i.e., usually $\mathcal{O}(n^3)$per iteration (where n is the number of samples). The second relates to the over-complicated optimization processes of these methods, which increase the risk of the algorithm becoming trapped in bad local minimums and thus lead to unsatisfying clustering performance.

The second category, which is termed as 'late fusion', maximizes the alignment between the consensus clustering indicator matrix and the weighted base clustering indicator matrices with orthogonal transformation, in which each clustering indicator matrix is generated by performing clustering on each single view [12,13]. This type of algorithm can improve the efficiency and clustering performance; however, these methods also simplify the search space of the optimal clustering indicator matrix, which decreases clustering performance.

Accordingly, in this paper, we propose a new late fusion framework designed to extract more information from the base clustering indicator matrix. We first assume that each base clustering indicator matrix is a feature representation after the sample is mapped from the original space to the feature space. Compared with the original features, this involves nonlinear relations between data [14]; moreover, this approach can also filter out some noise arising from the kernel matrix. Therefore, the similarity matrix constructed by the base clustering indicator matrix is superior to the original kernel matrix for clustering purposes.

We refer to the proposed method as 'late fusion multi-view clustering with learned consensus similarity matrix' (LF-MVCS). LF-MVCS jointly optimizes the combination coefficient of the base clustering indicator matrices and the consensus similarity matrix. This consensus similarity matrix is then used as the input of spectral clustering [15] to obtain the final result. Furthermore, we design an effective algorithm for solving the resulting optimization problem, as well as analyzing its computational complexity and convergence. Extensive experiments on 10 multi-view benchmark datasets are then conducted to evaluate the effectiveness and efficiency of our proposed method.

The contributions of this paper can be summarized as follows:

- The proposed LF-MVSC integrates the information of all views in a late fusion manner. It further optimizes the combination coefficient of each base clustering indicator matrix and consensus similarity matrix. To the best of our knowledge, this is the first time that a unified form has been obtained that can better reflect the similarity between samples via late fusion.
- The proposed optimization formulation has just one parameter to tune and avoids excessive time consumption. We therefore design an alternate optimization algorithm with proved convergence that can efficiently tackle the resultant problem. Our algorithm avoids complex computation owing

to fewer optimization variables being utilized, meaning that that the objective function can converge within fewer iterations.

2 Related Work

2.1 Multiple Kernel K-means

Let $\{x_i\}_{i=1}^{n} \subseteq \mathcal{X}$ be a dataset consisting of n samples, while $\varnothing(\cdot)\ :\ x \in \mathcal{X} \rightarrow \mathcal{H}$ is the feature mapping, which transfers x into a reproducing kernel Hilbert space $\mathcal{H}$. Kernel k-means clustering aims to minimize the sum-of-squares loss over the cluster assignment matrix $Z \in \{0,1\}^{n\times k}$, which can be formulated as follows:

$$\boldsymbol{Z} \in \underset{\{0,1\}^{n\times k}}{\min} \sum\nolimits_{i=1,c=1}^{n,k} Z_{ic} \parallel \varnothing(\boldsymbol{X}_i) - \boldsymbol{\mu}_c \parallel_2^2 \ s.t.\ \sum\nolimits_{c=1}^{k} Z_{ic} = 1 \tag{1}$$

Here, $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{n} Z_{ic}\varnothing(\boldsymbol{X}_i)$ is the centroid of the $c-th(1 \leq c \leq k)$ cluster, while $n_c = \sum_{i=1}^{n} Z_{ic}$ is the number of samples in the corresponding cluster.

In order to optimize Eq. (1), we can transform it as follows:

$$\underset{\boldsymbol{Z}}{\min}\ Tr(\boldsymbol{K}) - Tr\left(\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z}^T\boldsymbol{KZ}L^{\frac{1}{2}}\right)\ s.t.\ \boldsymbol{Z}\mathbf{1}_k = \mathbf{1}_n \tag{2}$$

where K is a kernel matrix with $K_{ij} = \varnothing(\boldsymbol{X}_i)^T\varnothing(\boldsymbol{X}_j)$, $L = \text{diag}\left(\left[n_1^{-1},\ \ldots,\ n_k^{-1}\right]\right)$, while $\mathbf{1}_s \in \mathbb{R}^s$ denotes a vector with all elements being 1. In Eq. (2), the variables $\boldsymbol{Z}$ are discrete, which makes the optimization problem difficult to solve. Dhillon et al. [16] let $\boldsymbol{H} = \boldsymbol{ZL}^{\frac{1}{2}}$ and relaxed $\boldsymbol{Z}$ to take the arbitrary real values $\boldsymbol{H}$. Thus, Eq. (2) can be rewritten as follows:

$$\underset{\boldsymbol{H}}{\min}\ Tr\left(\boldsymbol{K}\left(\boldsymbol{I_n} - \boldsymbol{HH}^T\right)\right)\ s.t.\ \boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}_K \tag{3}$$

Here, $\boldsymbol{I}_K$ is an identity matrix with size $k \times k$. Due to $\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z}^T\boldsymbol{ZL}^{\frac{1}{2}} > \boldsymbol{I}_K$, we impose an orthogonality constraint on $\boldsymbol{H}$. Finally, the optimal $\boldsymbol{H}$ for Eq. (3) can be obtained by taking the k eigenvectors corresponding to the k largest eigenvalues of $\boldsymbol{K}$. The Lloyd's algorithm is usually executed on $\boldsymbol{H}$ for the clustering labels.

In multiple kernel k-means (MKKM), all samples have multiple feature representations via a group of feature mappings $\{\varnothing(\cdot)\}_{p=1}^{m}$. More specifically, each sample can be represented as $\varnothing_\alpha(\boldsymbol{X}) = \left[\alpha_1\varnothing_1(X)^T,\ \ldots,\right]\alpha_m\varnothing_m(X)^T]^T$, where $\alpha = [\alpha_1,\ \ldots,\ \alpha_m]^T$ are the coefficients of each base kernel. The kernel function over the above feature mappings can be written as:

$$\mathcal{K}_\alpha(X_i, X_j) = \varnothing_\alpha(X_i)^T\varnothing_\alpha(X_j) = \sum_{p=1}^{m}\alpha_p^2\mathcal{K}_p(X_i, X_j) \tag{4}$$

Assume that the optimal kernel matrix can be represented as $\boldsymbol{K}_\alpha = \sum_{p=1}^{m}\boldsymbol{K}_p$, we obtain the optimization objective of MKKM as follows:

$$\min_{\boldsymbol{H},\boldsymbol{\alpha}} Tr\left(\boldsymbol{K_\alpha}\left(\boldsymbol{I_n} - \boldsymbol{HH}^T\right)\right) \ s.t. \ \boldsymbol{H} = \mathbb{R}^{n\times k}, \ \boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}_K, \ \boldsymbol{\alpha}^T 1 = 1, \ \alpha > 0 \tag{5}$$

A two-step iterative algorithm can be used to optimize the Eq. (5).

i) Optimize α with fixed H. Eq. (5) is equivalent to:

$$\min_{\boldsymbol{\alpha}} \sum_{p=1}^{m} \alpha_p^2 \delta_p \ s.t. \ \boldsymbol{\alpha}^T\mathbf{1} = 1, \ \alpha \geq 0 \tag{6}$$

where $\delta_p = \ Tr\left(\boldsymbol{K}_p\left(\boldsymbol{I}_n - \boldsymbol{HH}^T\right)\right)$. According to the Cauchy-Schwartz inequality, we can obtain the closed-form solution to Eq. (6) as $\alpha_p = \frac{1/\sqrt{\delta_p}}{\sum_{p=1}^{m} 1/\sqrt{\delta_p}} \left(\forall_p \in [m]\right)$.

ii) **Optimize** H **with fixed α**. With the kernel coefficients α fixed, $\boldsymbol{H}$ can be optimized by taking the k eigenvectors corresponding to the k largest eigenvalues of $\boldsymbol{K}_\alpha$.

2.2 Multi-View Clustering via Late Fusion Alignment Maximization

Multi-view clustering via late fusion alignment maximization [12] (MVC-LFA) assumes that the clustering indicator matrix obtained from each single view is $\boldsymbol{H}_p(p \in [m])$, while a set of rotation matrices $\left\{\boldsymbol{W}_p\right\}_{p=1}^m$ are used to maximize the alignment of $\boldsymbol{H}_p(p \in [m])$ and the optimal clustering indicator matrix $\boldsymbol{H}^*$. The objective function is as follows:

$$\max_{\boldsymbol{H}^*, \left\{W_p\right\}_{p=1}^m, \boldsymbol{\mu}} Tr\left(\boldsymbol{H}^{*T}\boldsymbol{R}\right) + \lambda Tr\left(\boldsymbol{H}^{*T}\boldsymbol{F}\right) s.t. \ \boldsymbol{H}^* \in \mathbb{R}^{n\times k}, \ \boldsymbol{H}^{*T}\boldsymbol{H}^* = I_k, \ \sum_{p=1}^{m}\mu_p^2 = 1,$$
$$\mu_p \geq 0, \boldsymbol{R} = \sum_{p=1}^{m} \mu_p \boldsymbol{H}_p \boldsymbol{W}_p \tag{7}$$

where F denotes the average clustering indicator matrix, while λ is a trade-off parameter. Moreover, $Tr\left(\boldsymbol{H}^{*T}\boldsymbol{F}\right)$ is a regularization of the optimal clustering indicator matrix that prevents $\boldsymbol{H}^*$ from being too far away from the prior average partition. This formula assumes that the optimal clustering indicator matrix $\boldsymbol{H}^*$ is a linear combination of the base clustering indicator matrix $H_p(p \in [m])$. Readers are encouraged to refer to [12] for more details regarding the optimization of Eq. (7).

The drawback of MKKM is its excessive computational complexity, which is $\mathcal{O}(n^3)$. Although MVC-LFA does reduce the computational complexity from $\mathcal{O}(n^3)$to $\mathcal{O}(n)$, it also over-simplifies the search space of the optimal solution, which in turn limits the clustering performance. We therefore propose a novel late fusion multi-view clustering algorithm in Section Three, which offers a trade-off between clustering efficiency and effectiveness.

3 Multi-View Clustering Via Late Fusion Consensus Similarity Matrix

3.1 Proposed Method

By taking the eigenvectors corresponding to the first k largest eigenvalues of $\left\{\boldsymbol{K_p}\right\}_{p=1}^m$, we can obtain the clustering indicator matrices $\left\{\boldsymbol{H_p}\right\}_{p=1}^m \subseteq \mathbb{R}^{n\times k}$. Each row of $\boldsymbol{H_p}$ can be regarded as a feature representation of each sample in the k dimension space. To construct the consensus similarity matrix $\boldsymbol{S}$, we need to construct the similarity matrix of each view. After normalizing each row of $\boldsymbol{H_p}$ (i.e., dividing by the 2-norm of the row), the 2-norm of each row of the processed $\boldsymbol{H_p}$ becomes 1. We treat $S^{(p)} = \boldsymbol{H_p}\boldsymbol{H_p}^T \in \mathbb{R}^{n\times n}$ as the similarity matrix of the p-th view.

Based on the assumption that the optimal consensus similarity matrix resides in the neighbor of a linear combination of all similarity matrices, our model can be expressed as follows:

$$\min_{\boldsymbol{S},\boldsymbol{\alpha}} \sum_{p=1}^{m} \alpha_p^2 \parallel \boldsymbol{S} - \boldsymbol{H_p H_p}^T \parallel_F^2, \ s.t. \ \sum_{p=1}^{m} \boldsymbol{\alpha_p} = 1, \ \alpha \geq 0 \tag{8}$$

The solution to the above formula is clearly equal to a certain $\boldsymbol{S_p}$, such that multi-view information is not fully utilized. Accordingly, inspired by [9], we introduce a regularization item capable of improving the diversity of all views for learning purposes. We let $\mathcal{M}(p, q) = Tr(H > \boldsymbol{H_p}^T \boldsymbol{H_q})$. A larger $\mathcal{M}(p, q)$ is associated with a higher degree of correlation between the p-th and q-th points. Let $M \in \mathbb{R}^{m \times m}$ be the matrix that stores diversified information of views, while $M_{pq} = \mathcal{M}(p, q)$. Thus, when M_{pq} is large, to avoid the information redundancy caused by similar views, we can set the coefficient of view such that $\boldsymbol{\alpha_p \alpha_q} M_{pq}$ is small. Based on the above, we add the following regularization item to the formula:

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T M \boldsymbol{\alpha} \ \ s.t. \ \sum_{p=1}^{m} \alpha_p = 1, \ \alpha_p \geq 0$$

At the same time, we aim to have **S** satisfy the basic properties of the similarity matrix. Hence, we impose certain constraints on **S,** namely: i) The degree of each sample in **S** is greater than or equal to 0, i.e., $S_{ij} \geq 0$; ii) The degree of each sample in **S** is equal to 1, i.e., $\boldsymbol{S^T 1} = \boldsymbol{1}$. We can accordingly obtain the following formula:

$$\min_{\boldsymbol{S},\boldsymbol{\alpha}} \sum_{p=1}^{m} \alpha_p \parallel \boldsymbol{S} - \boldsymbol{H_p H_p}^T \parallel_F^2 + \lambda \boldsymbol{\alpha}^T M \boldsymbol{\alpha}, \ s.t. \ \sum_{p=1}^{m} \alpha_p = 1, \ \alpha \geq 0, \ S_{ij} \geq 0, \ \boldsymbol{S^T 1} = \boldsymbol{1} \tag{9}$$

Not only can the clustering indicator matrices obtained by KKM, but also the similar basic partitions constructed by any clustering algorithms, be the input of our model. The model developed thus far represents a new multi-view clustering framework based on late fusion. Compared to other late fusion algorithms, such as [12,13], our objective formula has three key advantages, namely:

Fewer variables need to be optimized, meaning that the total number of iterations is reduced and the algorithm converges quickly.

Other late fusion algorithms hold that the optimal clustering indicator matrix is a neighbor of the average clustering indicator matrix; however, this is a heuristic assumption that may not hold true in some cases. By contrast, our assumptions in Eq. (9) are more reasonable and will be effective in any multi-view clustering task.

We further introduce a regularization item to measure the correlation of views in order to increase both the diversity and the utilization rate of multi-view information.

3.2 Alternate Optimization

In the following, we construct an efficient algorithm designed to solve the optimization problem in Eq. (9). More specifically, we design a two-step algorithm to alternately solve the following:

Optimization α with fixed S.

When S is fixed, the optimization problem in Eq. (9) is equivalent to the following optimization problem:

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T(\lambda \boldsymbol{M} + \boldsymbol{Z})\boldsymbol{\alpha},\ s.t.\ \sum_{p=1}^{m} \alpha_p = 1,\ \alpha_p \geq 0 \tag{10}$$

Here, $Z = diag \parallel \left(\boldsymbol{S} - \boldsymbol{H_1H_1}^T \parallel_F^2, \ldots, \parallel \boldsymbol{S} - \boldsymbol{H}_m\boldsymbol{H}_m{}^T \parallel_F^2\right)$. Moreover, $\lambda\boldsymbol{M} + \boldsymbol{Z}$ is a nonnegative definite matrix, meaning that the optimization in Eq. (10) w.r.t α is a quadratic programming problem with linear constraint. We can solve this using the Matlab QP toolkit.

Optimization S with fixed α.

When α is fixed, the optimization problem in Eq. (9) w.r.t **S** can be rewritten as follows:

$$\min_{S} \sum_{p=1}^{m} \alpha_p \parallel \boldsymbol{S} - \boldsymbol{H}_p\boldsymbol{H}_p{}^T \parallel_F^2,\ s.t.\ S_{ij} \geq 0,\ S^T 1 = 1 \tag{11}$$

Note that the problem in Eq. (11) is independent of the value of j, meaning that we can optimize each column of S as follows:

$$\min_{S_j} \sum_{p=1}^{m} \alpha_p S_j^T S_j - \alpha_p^2 S_j^T S_j^{(p)},\ s.t.\ S_{ij} \geq 0,\ S_j^T 1 = 1 \tag{12}$$

where $S_j^{(p)}$ is the j-th column of $\boldsymbol{S}^{(p)} = \boldsymbol{H_pH_p}^T$. Eq. (12) is equivalent to the following function:

$$\min_{\boldsymbol{S}_j} \frac{1}{2} \parallel \boldsymbol{S}_j - \boldsymbol{C}_j \parallel_2^2,\ s.t.\ S_{ij} \geq 0,\ \boldsymbol{S^T1} = 1 \tag{13}$$

Here, $\boldsymbol{C_j} = \frac{\alpha_p^2 \boldsymbol{S}_j^{(p)}}{\alpha_p^2}$. Eq. (13) is a problem on the simplex space, and its Lagrangian function is as follows:

$$\mathcal{L}(S_j, \gamma, \boldsymbol{\tau}) = \frac{1}{2} \parallel \boldsymbol{S}_j - \boldsymbol{C}_j \parallel_2^2 + \gamma\left(s_j^T 1 - \mathbf{1}\right) - \boldsymbol{\tau}^T\boldsymbol{S}_j \tag{14}$$

where γ and τ are the Lagrangian multipliers.

According to its Karush-Kuhn-Tucker condition [17], we can obtain the optimal solution $\boldsymbol{S}^*$ in the form of Eq. (15):

$$\boldsymbol{S}_j^* = \left(\boldsymbol{C}_j - \gamma^* 1\right)_+ \tag{15}$$

Here, $(x)_+$ is a signal function, which sets the elements of x that are less than 0 to 0, while $\gamma^* \in$ R makes $\mathbf{1^T} > \boldsymbol{S}_j^* = 1$.

The algorithm is summarized in detail in Algorithm 1.

Algorithm 1: Late fusion multi-view clustering with learned consensus similarity matrix

```
Input: $\{H_p\}_{p=1}^m$ and ε.
Output: S.
Initialize α as $^{1_m}/_m$ and t as 1.
    while not converge do
        for j ∈ {1,2, …, n} do
            Update $s_j$, by solving (13) with fixed α.
        end for
```

Algorithm 1 (continued).

```
    $S = \frac{S + S^T}{2}$.
    Update α, by solving (10) with fixed **S**.
    t = t + 1.
**end while** $\| \alpha^{(t)} - \alpha^{(t-1)} \|_2^2 \leq \epsilon$
```

3.3 Complexity and Convergence

Computational Complexity. Obtaining the base clustering indicator matrices requires calculating the eigenvectors corresponding to the k largest eigenvalues of the m kernel matrices, leading to a complexity of $\mathcal{O}(\mathrm{kmn}^2)$. The proposed iterative algorithm further requires the optimization of the two variables α and S. In each iteration, the optimization of α via Eq. (10) requires the solving of a standard quadratic programming problem with linear constraint, the complexity of which is $\mathcal{O}(m^3)$. Updating each s_j by Eq. (13) requires $\mathcal{O}(n)$ time. Moreover, the complexity associated with obtaining S is $\mathcal{O}(n^2)$ per iteration, as n times are needed to calculate each sj. Overall, the total complexity of Algorithm 1 is $\mathcal{O}((m^3 + n^2)t)$, where *t* is the number of iterations. Finally, executing the standard spectral clustering algorithm on S takes $\mathcal{O}(n^3)$ time.

Convergence. In each of the optimization iterations of our proposed algorithm, the optimizations of α and S monotonically decrease the value of the objective Eq. (9). Moreover, this objective is also lower-bounded by zero. Consequently, our algorithm is guaranteed to converge to the local optimum of Eq. (9).

4 Experiments

4.1 Datasets and Experimental Settings

Several benchmark datasets are utilized to demonstrate the effectiveness of the proposed method: namely, Flower17 (www.robots.ox.ac.uk/□vgg/data/flowers/17/), Flower102 (www.robots.ox.ac.uk/□vgg/data/flowers/102/), PFold (mkl.ucsd.edu/dataset/protein-fold-prediction), Digit (http://ss.sysu.edu.cn/py/), and Cal (www.vision.caltech.edu/Image Datasets/Caltech101/). Six sub-datasets, which are constructed by selecting the first 5, 10, 15, 20, 25 and 30 classes respectively from the Cal data, are also used in our experiments. More detailed information regarding these datasets is presented in Tab. 1. From this table, we can observe that the number of samples, views and categories of these datasets range from 510 to 8189, 3 to 48, and 5 to 102, respectively.

Table 1: Benchmark datasets

Datasets	Samples	Kernels	Clusters
Flower17	1360	7	17
Flower102	8189	4	102
ProteinFold	694	12	27
Digit	2000	3	10
Cal-5	510	48	5
Cal-10	1020	48	10
Cal-15	1530	48	15
Cal-20	2040	48	20
Cal-25	2550	48	25
Cal-30	3060	48	30

4.2 Comparison with State-of-the-Art Algorithms

In our experiments, LF-MVCS is compared with several state-of-the-art multi-view clustering methods, as outlined below.

Average multiple kernel k-means (A-MKKM): The kernel matrix, which is a linear combination of the base kernels with the same coefficients, is taken as the input of the standard kernel $\mathcal{K}$-means algorithm.

Single best kernel k-means (SB-KKM): Stardard kernel k-means is performed on each single kernel, after which the best result is outputted.

Multiple kernel k-means (MKKM) [7]: The algorithm jointly optimizes the consensus clustering indicator matrix and the kernel coefficients.

Robust multiple kernel k-means (RMKKM) [18]: RMKC learns a robust low-rank kernel for clustering by capturing the structure of noise in multiple kernels.

Multiple kernel k-means with matrix-induced regularization (MKKM-MR) [9]: This algorithm introduces a matrix-induced regularization designed to reduce the redundancy of the kernels and enhance their diversity.

Optimal neighborhood kernel clustering with multiple kernels (ONKC) [11]: ONKC allows the optimal kernel to reside in the neighborhood of linear combination of base kernels, thereby effectively enlarging the search space of the optimal kernel.

Multi-view clustering via late fusion alignment maximization (MVC-LFA) [12]: MVC-LFA proposes maximally aligning the consensus clustering indicator matrix with the weighted base clustering indicator matrix.

Simple multiple kernel k-means (SMKKM) [19]: SimpleMKKM, or SMKKM, re-formulates the MKKM problem as a minimization-maximization problem in the kernel coefficients and the consensus clustering indicator matrix.

4.3 Experimental Settings

In our experiments, all base kernels are first centered and then normalized; therefore, for all samples $\mathrm{x_i}$ and the p-th kernel, we have $K_p\left(x_i, x_j\right) = 1$. For all datasets, the true number of clusters is known, and we set this to be the true number of classes. In addition, the parameters of RMKKM, MKKM-MR and MVC-LFA are selected by grid search according to the methods suggested in their respective papers. For the proposed algorithm, its regularization parameters λ are chosen from a large enough range $[2^{-15}, 2^{-12}, \ldots, 2^{15}]$ by means of grid search. We set the allowable error ϵ to 1×10^{-4}, which is the termination condition of Algorithm 1.

The widely used metrics of clustering accuracy (ACC), normalized mutual information (NMI) and purity are applied to evaluate the clustering performance of each algorithm. For all algorithms, we repeat each experiment 50 times with random initialization (to reduce randomness caused by k-means in the final spectral clustering) and report the average result. All experiments are performed on a desktop with Intel(R) Core(TM)-i7-7820X CPU and 64 GB RAM.

4.4 Experimental Results

Tab. 2 reports the clustering performance of the above-mentioned algorithms on all datasets. From these results, we can make the following observations:

First, the proposed algorithm consistently demonstrates the best NMI on all datasets. For example, it exceeds the second-best method (MVC-LFA) by over 8.56% on Flower17. The superiority of our algorithm is also confirmed by the ACC and purity reported in Tab. 2.

Table 2: Empirical evaluation and comparison of SimpleMKKM with eight baseline methods on 11 datasets in terms of clustering accuracy (ACC), normalized mutual information (NMI) and purity

Dataset	A-MKKM	SB-KKM	MKKM Huang et al. [7]	RMKKM Du et al. [18]	MKKM-MR Liu et al. [9]	ONKC Liu et al. [11]	MVC-LFA Wang et al. [12]	SMKKM Liu et al. [13]	Proposed
ACC(%)									
Flower17	51.29	42.41	43.84	49.78	57.66	53.40	59.86	58.40	**64.15**
Flower102	26.85	33.22	22.51	29.19	40.32	38.51	42.32	42.26	**47.21**
ProteinFold	29.05	33.53	27.56	27.81	35.15	36.03	32.47	34.84	**36.20**
Digit	88.82	73.05	47.20	42.85	90.34	90.83	88.83	90.34	**96.63**
Cal-5	35.95	35.95	27.67	31.37	36.11	35.90	37.73	36.03	**43.33**
Cal-10	31.79	33.60	22.17	25.69	32.67	32.38	34.08	32.62	**44.31**
Cal-15	30.55	33.34	19.72	21.96	31.80	30.85	33.10	31.47	**39.47**
Cal-20	29.47	33.87	18.45	21.96	31.47	30.02	32.36	31.24	**39.16**
Cal-25	29.15	33.44	17.26	22.47	30.47	29.33	31.50	30.91	**41.96**
Cal-30	28.53	33.80	16.45	20.46	30.44	29.11	31.39	30.75	**38.23**
NMI(%)									
Flower17	49.87	45.08	44.65	50.41	56.07	52.59	57.90	56.66	**66.46**
Flower102	45.94	48.76	42.69	48.68	56.75	55.19	56.88	58.55	**61.66**
ProteinFold	40.32	41.09	38.27	37.73	44.30	44.92	42.03	44.39	**46.66**
Digit	80.74	66.26	48.74	47.74	83.14	83.70	80.78	83.31	**93.52**
Cal-5	70.26	70.06	66.01	65.94	70.29	70.24	70.99	70.33	**73.35**
Cal-10	61.77	62.78	55.50	55.99	62.23	61.98	63.04	62.22	**68.74**
Cal-15	57.20	58.86	49.35	50.43	57.99	57.39	58.95	57.97	**62.63**
Cal-20	54.23	56.49	45.48	47.43	55.50	54.38	56.07	55.39	**60.11**
Cal-25	52.09	54.50	42.35	45.28	53.10	52.32	54.02	53.46	**60.57**
Cal-30	49.95	53.39	40.08	43.12	51.54	50.45	52.45	51.95	**57.00**
Purity(%)									
Flower17	52.26	44.62	45.30	51.03	59.13	54.24	61.30	59.42	**65.87**
Flower102	32.17	38.46	27.93	34.61	46.36	44.30	48.69	48.58	**52.77**
ProteinFold	37.32	39.19	34.31	33.29	42.44	**43.10**	39.37	41.62	41.20
Digit	88.82	73.40	50.05	45.85	90.34	90.83	88.83	90.34	**96.66**
Cal-5	37.32	37.36	28.42	33.14	37.63	37.27	39.48	37.50	**45.10**
Cal-10	33.62	35.83	23.53	27.16	34.83	34.37	36.26	34.54	**46.56**
Cal-15	32.27	35.34	21.16	24.12	33.71	32.73	35.12	33.27	**41.96**
Cal-20	31.50	35.80	20.07	24.07	33.41	31.93	34.40	33.28	**41.47**
Cal-25	31.12	35.35	18.79	23.92	32.64	31.58	33.93	33.11	**44.82**
Cal-30	30.49	35.71	17.83	22.55	32.42	31.03	33.58	32.75	**40.78**

As a strong baseline, the recently proposed SMKKM [19] outperforms other comparison early-fusion multiple kernel clustering methods. However, the proposed algorithm significantly and consistently outperforms SMKKM by 9.80%, 3.11%, 2.27%, 10.21%, 3.02%, 6.52%, 4.66%, 4.72%, 7.11% and 5.05% respectively in terms of NMI on all datasets in the order listed in Tab. 1.

The late fusion algorithm MVC-LFA also outperforms most other algorithms; however, the proposed algorithm outperforms MVC-LFA by 8.56%, 4.78%, 4.63%, 12.74%, 2.36%, 5.70%, 3.68%, 4.04%, 6.55% and 4.55% respectively in terms of NMI on all datasets in the order listed in Tab. 1.

We further study the clustering performance on Cal for each algorithm depending on the number of classes, as shown in Fig. 1. As can be observed from the figure, our algorithm is consistently at the top of all sub-figures when the number of classes varies, indicating that it achieves the best performance.

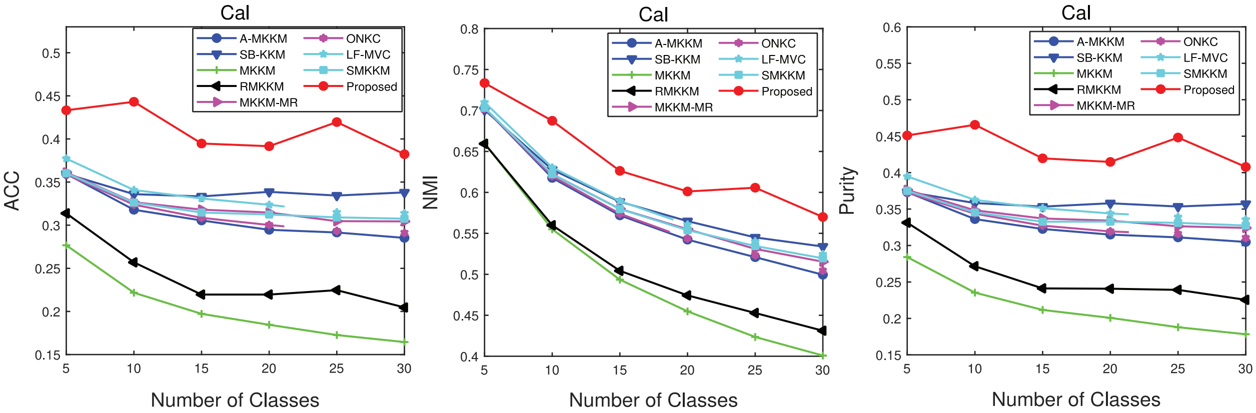

Figure 1: ACC, NMI and purity comparison with variations in number of classes on Cal. (left) ACC; (mid) NMI; (right) purity

In summary, our proposed LF-MVCS demonstrates superior clustering performance over all SOTA multi-view clustering algorithms. From the above experiments, we can therefore conclude that the proposed algorithm effectively learns the information from all base clustering indicator matrices, bringing significant improvements to the clustering performance.

4.5 Runtime, Convergence and Parameter Sensitivity

Runtime. To investigate the computational efficiency of the proposed algorithm, we record the running time of various algorithms on the benchmark datasets and report them in Fig. 2. The computational complexity of the proposed LF-MVCS is $\mathcal{O}(n^2)$; as can be observed, LF-MVCS has slightly higher computational complexity than the existing MVC-LFA (with a computational complexity of $\mathcal{O}(n)$). At the same time, however, the runtime of LF-MVCS is lower relative to with other kernel-based multi-view clustering algorithms.

Algorithm Convergence. We next theoretically analyze the convergence of our algorithm. Fig. 3 reflects the variation of the objective Eq. (9) with the number of iterations on four datasets: namely, Flower17, ProteinFold, Digit, and Cal-30. It can be seen that the objective function achieves convergence over very few iterations, thereby demonstrating that fewer optimization variables can make the algorithm converge faster, which makes our algorithm more efficient than other algorithms.

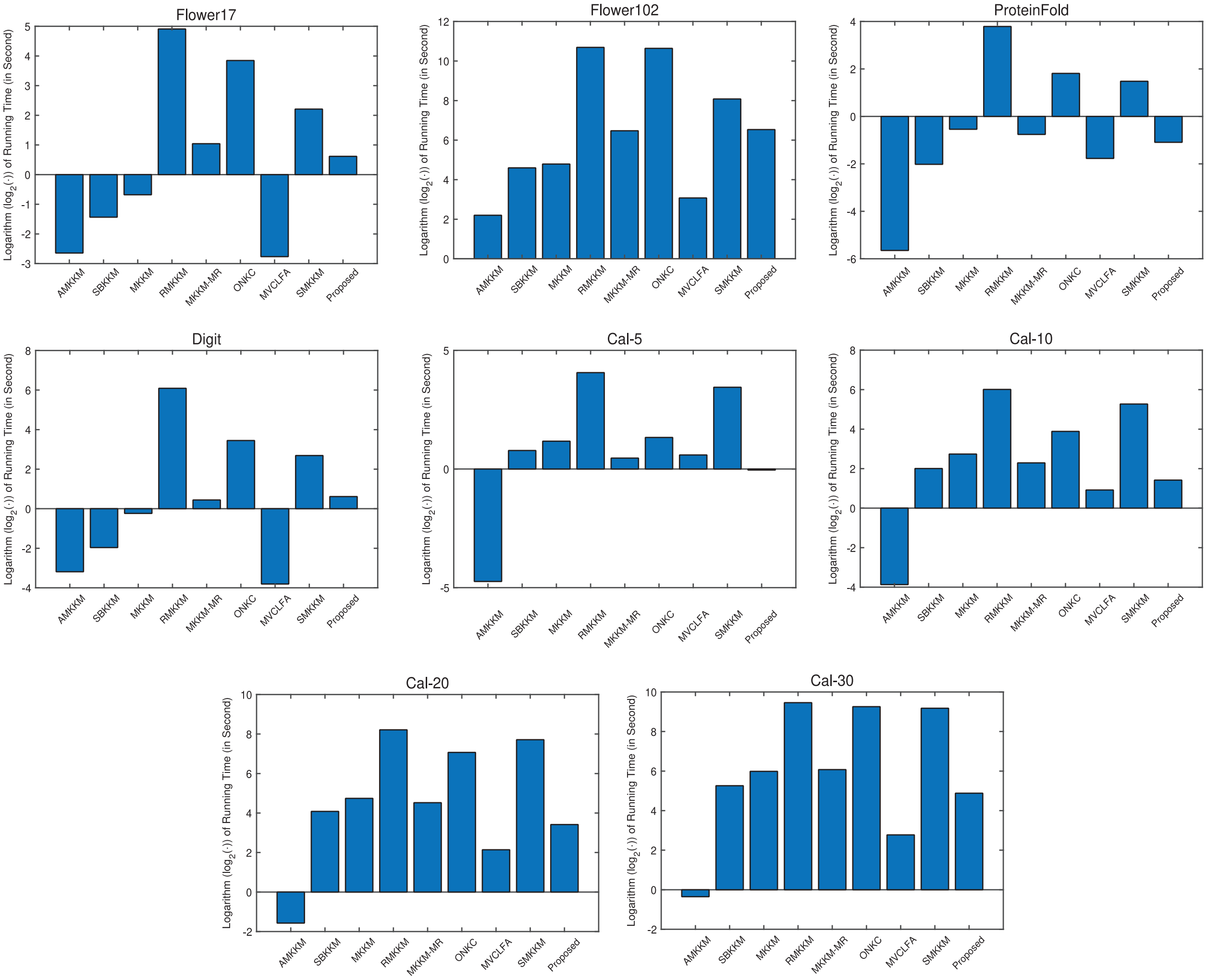

Figure 2: Runtime of different algorithms on eight benchmark datasets (in seconds). The experiments are conducted on a PC with Intel(R) Core(TM)-i7-7820X CPU and 64 GB RAM in the MATLAB environment. LF-MVCS is comparably fast relative to the alternatives while also providing superior performance when compared with most other multiple kernel k-means based algorithms. Results for other datasets are omitted due to space limitations

Parameter Sensitivity. The proposed algorithm introduces one hyper-parameter, i.e., the balancing coefficient of the diversity regularization item λ. The performance variation against λ is illustrated in Fig. 4. As the figure shows: i) λ is effective in improving the algorithm performance; ii) It also achieves good performance over a wide range of parameter settings.

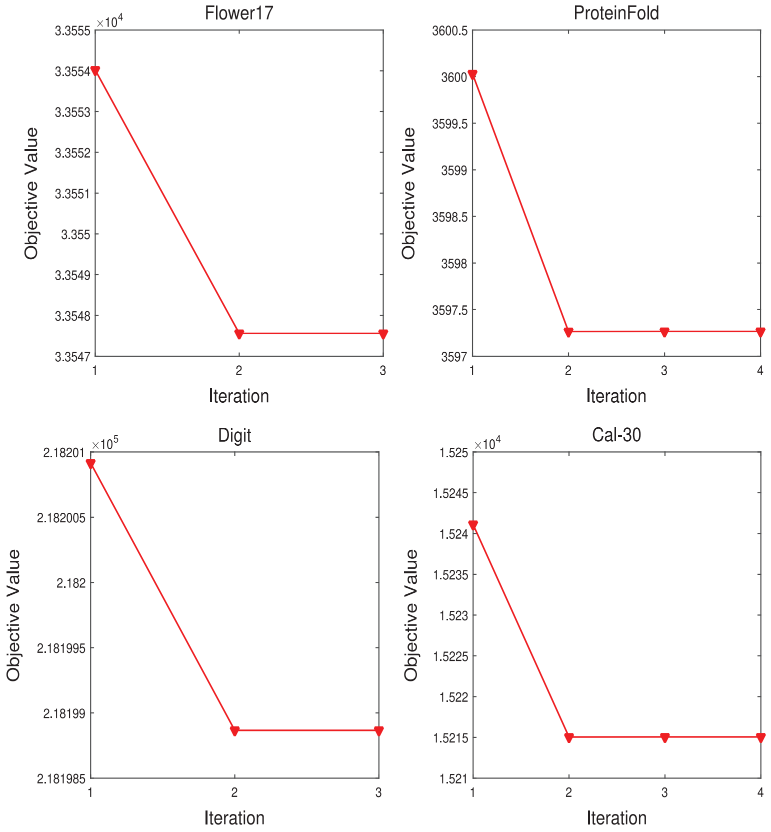

Figure 3: The objective value of our algorithm on four datasets at each iteration

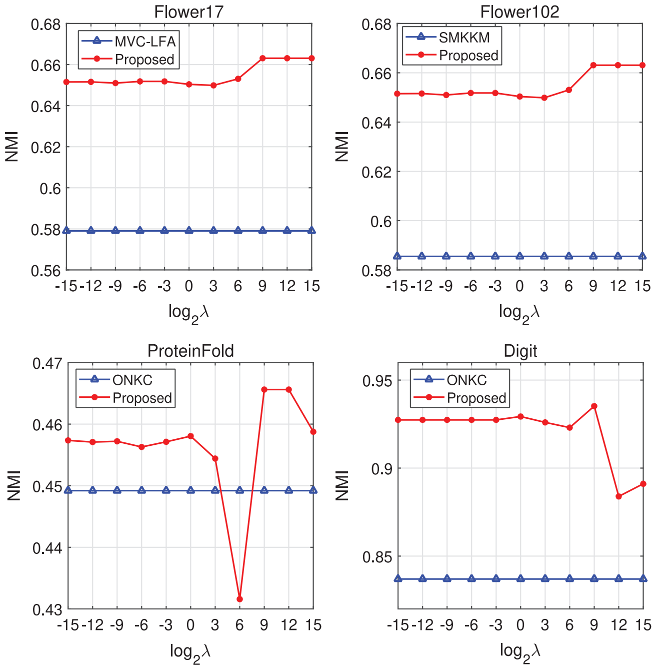

Figure 4: Illustration of parameter sensitivity against λ. The red curves represent the results of the proposed algorithm, while the blue lines indicate the second-best performance on the corresponding dataset

5 Conclusion

This paper proposes a simple but effective algorithm, LF-MVCS, which learns a consensus similarity matrix from all base clustering indicator matrices. We determine that each base clustering indicator matrix is a better feature representation than the original feature and kernel. Based on this finding, we derive a simple novel optimization goal for multi-view clustering, which learns a consensus similarity matrix to facilitate better clustering performance. The proposed algorithm is developed to efficiently solve the resultant optimization problem. LF-MVCS outperforms all SOTA methods significantly in terms of clustering performance. Moreover, our algorithm also reduces the computational cost on benchmark datasets when compared with other kernel-based multi-view clustering methods. In future work, we plan to extend our algorithm to a general framework, then use this framework as a platform to revisit existing multi-view algorithms in order to further improve the fusion method.

Funding Statement: This paper and research results were supported by the Hunan Provincial Science and Technology Plan Project. The specific grant number is 2018XK2102. Y. P. Zhao, W. X. Liang, J. Z. Lu and X. W. Chen all received this grant.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Gao, F. Nie, X. Li and H. Huang, “Multi-view subspace clustering,” in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 4238–4246, 2015.

[2] C. Zhang, Q. Hu, H. Fu, P. Zhu and X. Cao, “Latent multi-view subspace clustering,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 4333–4341, 2017.

[3] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie *et al.,* “Generalized latent multi-view subspace clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 86–99, 2020.

[4] A. Kumar, P. Rai and H. Daume, “Co-regularized multi-view spectral clustering,” in *Advances in Neural Information Processing Systems*, Granada, Spain, pp. 1413–1421, 2011.

[5] Y. Li, F. Nie, H. Huang and J. Huang, “Large-scale multi-view spectral clustering via bipartite graph,” in *Proc. of the Twenty-Ninth AAAI Conf. on Artificial Intelligence*, Austin, Texas, USA, pp. 2750–2756, 2015.

[6] S. Zhou, X. Liu, J. Liu, X. Guo, Y. Zhao *et al.,* “Multi-view spectral clustering with optimal neighborhood laplacian matrix,” in *Proc. of the Thirty-Fourth AAAI Conf. on Artificial Intelligence*, New York, NY, USA, pp. 6965–6972, 2020.

[7] H. Huang, Y. Chuang and C. Chen, “Multiple kernel fuzzy clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, 2012.

[8] M. Gonen and A. A. Margolin, “Localized data fusion for kernel k-means clustering with application to cancer biology,” in *Advances in Neural Information Processing Systems*, Ambridge, MA, USA, pp. 1305–1313, 2014.

[9] X. Liu, Y. Dou, J. Yin, L. Wang and E. Zhu, “Multiple kernel k-means clustering with matrix-induced regularization,” in *Proc. of the Thirtieth Conf. on Artificial Intelligence*, Phoenix, Arizona, USA, pp. 1888–1894, 2016.

[10] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin *et al.,* “Multiple kernel clustering with local kernel alignment maximization,” in *Proc. of the Twenty-Fifth Int. Joint Conf. on Artificial Intelligence*, New York, NY, USA, pp. 1704–1710, 2016.

[11] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou *et al.,* “Optimal neighborhood kernel clustering with multiple kernels,” in *Proc. of the Thirty-First Conf. on Artificial Intelligence*, San Francisco, California, USA, pp. 2266–2272, 2017.

[12] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu *et al.,* “Multi- view clustering via late fusion alignment maximization,” in *Proc. of the Twenty-Eighth Int. Joint Conf. on Artificial Intelligence*, Macao, China, pp. 3778–3784, 2019.

[13] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong *et al.,* "Efficient and effective regularized incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 3778–3784, 2020.

[14] B. Schölkopf, A. Smola and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[15] A. Y. Ng, M. I. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, Cambridge, MA, United States, pp. 849–856, 2002.

[16] I. Dhillon, Y. Guan and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proc. of the Tenth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 551–556, 2004.

[17] S. Boyd and L. Vandenberghe, "Basic properties and examples, operations that preserve convexity, optimization problems, & convex optimization, " in *Convex Optimization*, Sao Paolo (ed.), vol. 1. London, UK: Cambridge University Press, pp. 67–142, 2004.

[18] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan *et al.,* "Robust multiple kernel k-means using l21-norm," in *Twenty-Fourth Int. Joint Conf. on Artificial Intelligence*, Buenos Aires, Argentina, pp. 3476–3482, 2015.

[19] X. Liu, E. Zhu and J. Liu, "Simplemkkm: Simple multiple kernel k-means," in *Proc. of the 37th Int. Conf. on Machine Learning*, Vienna, Austria, pp. 1–10, 2020.