Multi-view Clustering via Late Fusion Alignment Maximization

Siwei Wang[1] , **Xinwang Liu**[1*] , **En Zhu**[1] , **Chang Tang**[2] , **Jiyuan Liu**[1] , **Jingtao Hu**[1] , **Jingyuan Xia**[3] and **Jianping Yin**[4]

[1]College of Computer, National University of Defense Technology, Changsha, China
[2]School of Computer Science, China University of Geosciences, Wuhan, China
[3]Department of Electric and Electronic Engineering, Imperial College London
[4]School of Cyberspace Science, Dongguan University of Technology, Guangdong 523808, China
xinwangliu@nudt.edu.cn

Abstract

Multi-view clustering (MVC) optimally integrates complementary information from different views to improve clustering performance. Although demonstrating promising performance in many applications, we observe that most of existing methods directly combine multiple views to learn an optimal similarity for clustering. These methods would cause intensive computational complexity and over-complicated optimization. In this paper, we theoretically uncover the connection between existing k-means clustering and the alignment between base partitions and consensus partition. Based on this observation, we propose a simple but effective multi-view algorithm termed *Multi-view Clustering via Late Fusion Alignment Maximization (MVC-LFA)*. In specific, MVC-LFA proposes to maximally align the consensus partition with the weighted base partitions. Such a criterion is beneficial to significantly reduce the computational complexity and simplify the optimization procedure. Furthermore, we design a three-step iterative algorithm to solve the new resultant optimization problem with theoretically guaranteed convergence. Extensive experiments on five multi-view benchmark datasets demonstrate the effectiveness and efficiency of the proposed MVC-LFA.

1 Introduction

Multi-view clustering (MVC) has been intensively studied recently by integrating the available multi-view information to categorize data items with similar structures or patterns into the same group [Chen *et al.*, 2007; Gönen and Alpaydin, 2008; Yu *et al.*, 2012; Huang *et al.*, 2012; Liu *et al.*, 2013; Gönen and Margolin, 2014; Liu *et al.*, 2016; Li *et al.*, 2016; Liu *et al.*, 2017; Liu *et al.*, 2018]. Existing research in this filed can be summarized into two categories. The first one learns a latent consensus matrix via low-rank optimization [Xia *et al.*, 2014; Zhou *et al.*, 2015]. In [Xia *et al.*, 2014], it is proposed to learn a latent low-rank transition probability matrix shared from multiple views as the input to the standard

---

*Corresponding author

Markov chain for clustering. The work in [Zhou *et al.*, 2015] captures the noises in each kernel and integrates them into a low-rank framework. By following multiple kernel learning framework, the other category optimizes the optimal kernel matrix as a linear combination of known kernel matrices from a given library [Chen *et al.*, 2007; Yu *et al.*, 2012; Liu *et al.*, 2013; Gönen and Margolin, 2014; Liu *et al.*, 2016; Li *et al.*, 2016; Liu *et al.*, 2017; Liu *et al.*, 2019]. In [Chen *et al.*, 2007], a three-step alternate algorithm is proposed to jointly optimize clustering, kernel coefficients and dimension reduction. The work in [Liu *et al.*, 2016] proposes a multiple kernel k-means clustering algorithm with matrix-induced regularization to reduce the redundancy of the pre-defined kernels. Furthermore, local kernel alignment criterion has been applied to multiple kernel learning to enhance the clustering performance in [Li *et al.*, 2016]. In [Liu *et al.*, 2017], a multiple kernel algorithm is proposed to allow the optimal kernel to reside in the neighborhood of the combinational kernels. Our method in this paper belongs to the second category.

Although the aforementioned algorithms have improved multi-view clustering from different aspects, we observe that they suffer from the following drawbacks. i) The intensive computational complexity, i.e., usually $\mathcal{O}(n^3)$ per iteration with n the number of samples, prevents them from being applied into medium or large-scale clustering tasks. ii) As the work in [Liu *et al.*, 2018] mentions, the optimization processes of these methods are usually over-complicated, which could increase the risk of being trapped into local minimums, leading to unsatisfying clustering performance.

In this paper, we theoretically show that maximizing the alignment between base partitions and consensus partition is conceptually equivalent to minimize the loss function of existing k-means clustering. Based on this observation, we propose a novel algorithm named *Multi-view Clustering via Late Fusion Alignment Maximization* (MVC-LFA). It maximizes the alignment between the consensus partition matrix and the weighted base partition matrices with orthogonal transformation, where each base partition is generated by performing clustering on each single view. The proposed MVC-LFA jointly optimizes the orthogonal transformation matrices, weight coefficients and the optimal consensus partition. Moreover, we develop an efficient algorithm to solve the resultant optimization problem, and theoretically analyze its computational complexity and convergence. Extensive ex-
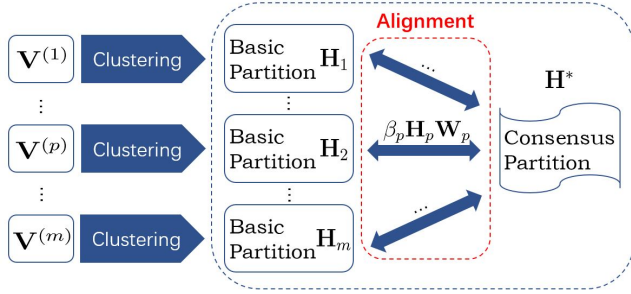
Late Fusion Alignment

Figure 1: Model pipeline for the proposed *Multi-view Clustering via Late Fusion Alignment* (MVC-LFA). Firstly, we obtain the base partitions independently from each view by kernel k-means clustering. Then, by maximizing alignment between the weighted basic partitions generated from different views and the consensus partition, our algorithm significantly reduces the time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ per iteration and also gives rise to clustering performance.

periments on five multiple-view benchmark datasets are conducted to evaluate the effectiveness and efficiency of our proposed method, including the clustering performance, the running time, the evolution of the learned consensus matrix and the objective value with iterations. As demonstrated, the proposed algorithm enjoys superior clustering performance with significant reduction in computational cost, in comparison with several state-of-the-art multi-view clustering methods.

The contributions of this paper are summarized as follows,

- We theoretically prove that maximizing the alignment between base partitions and the consensus partition is conceptually equivalent to minimize the loss function of k-means clustering. This observation motivates us to design a simple but effective multi-view clustering algorithm.

- The proposed MVC-LFA integrates multiple views in a late fusion manner. It simultaneously optimizes the consensus partition, transformation matrices and the weight coefficients of each view in a joint framework. *To the best of our knowledge, the proposed MVC-LFA is the first attempt to address multiple view clustering via maximizing alignment between consensus clustering matrix and weighted base partitions.*

- An alternate optimization algorithm with proved convergence is designed to efficiently tackle the resultant problem. By the virtue of it, MVC-LFA shows clearly superior clustering performance in comparison with state-of-the-art methods. More importantly, MVC-LFA requires significantly less computational time, especially critical for large datasets with limited computing sources.

2 Related Work

2.1 *K*-means Clustering

As one of the classical and widely-used algorithms, k-means provides an intuitive and effective way to perform clustering. To be specific, the clustering loss function of existing k-means algorithms can be written as mentioned in [Boutsidis *et al.*, 2015],

$$\min_{\mathbf{H}^*} \ \mathrm{Tr}(\mathbf{X}\mathbf{X}^{\mathrm{T}}) - \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{H}^*),$$
$$s.t. \ \mathbf{H}^* \in \mathbb{R}^{n \times k}, \mathbf{H}^{*\mathrm{T}}\mathbf{H}^* = \mathbf{I_k}, \qquad (1)$$

where $\mathbf{X}$ and $\mathbf{H}^*$ are the data and partition matrix respectively. We can obtain the optimal $\mathbf{H}^*$ for Eq. (1) by taking the k eigenvectors that correspond to the k largest eigenvalues of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$.

Eq. (1) could not directly apply to data with multi views. To overcome this limitation, many novel algorithms have been proposed to extend the k-means into multi-kernel k-means to handle multi-view clustering, which will be introduced in the next section.

2.2 Multiple Kernel *K*-means Clustering (MKKM)

For the multi-view based setting, we suppose that $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ is a collection of n samples with m views, and $\phi_p(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}_p$ be the p-th feature mapping, which transfers $\mathbf{x}$ into a reproducing kernel Hilbert space $\mathcal{H}_p$ ($1 \leq p \leq m$). Hence each sample $\mathbf{x}_i$ is represented as $\phi_{\boldsymbol{\beta}}(\mathbf{x}) = [\beta_1 \phi_1(\mathbf{x})^\top, \cdots, \beta_m \phi_m(\mathbf{x})^\top]^\top$ from m views, where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_m]^\top$ consists of the coefficients of the m base kernels $\{\kappa_p(\cdot, \cdot)\}_{p=1}^m$. These coefficients will be optimized during clustering. Based on the definition of $\phi_{\boldsymbol{\beta}}(\mathbf{x})$, a kernel function can be expressed as,

$$\kappa_{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\beta}}(\mathbf{x}_i)^{\mathrm{T}} \phi_{\boldsymbol{\beta}}(\mathbf{x}_j) = \sum\nolimits_{p=1}^m \beta_p^2 \kappa_p(\mathbf{x}_i, \mathbf{x}_j). \quad (2)$$

By denoting the optimal kernel matrix $\mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^m \beta_p^2 \mathbf{K}_p$, the optimization goal of MKKM algorithm can be formulated as,

$$\min_{\mathbf{H},\boldsymbol{\beta}} \ \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^{\mathrm{T}}))$$
$$s.t. \ \mathbf{H} \in \mathbb{R}^{n \times k}, \ \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}_k, \ \boldsymbol{\beta}^{\mathrm{T}}\mathbf{1}_m = 1, \ \beta_p \geq 0, \ \forall p. \quad (3)$$

The optimization problem in Eq. (3) can be solved by alternately updating $\mathbf{H}$ and $\boldsymbol{\beta}$. i) **Optimizing $\mathbf{H}$ by fixed $\boldsymbol{\beta}$.** With the kernel coefficients $\boldsymbol{\beta}$ fixed, $\mathbf{H}$ can be obtained by solving a kernel k-means clustering optimization problem,

$$\max_{\mathbf{H}} \ \mathrm{Tr}(\mathbf{H}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\beta}}\mathbf{H}) \ \ s.t. \ \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}_k. \quad (4)$$

The optimal $\mathbf{H}$ for Eq. (4) can be obtained by taking the k eigenvectors corresponding to the largest k eigenvalues of $\mathbf{K}_{\boldsymbol{\beta}}$. ii) **Optimizing $\boldsymbol{\beta}$ by fixed $\mathbf{H}$.** With $\mathbf{H}$ fixed, $\boldsymbol{\beta}$ can be optimized via solving the following quadratic programming with linear constraints,

$$\min_{\boldsymbol{\beta}} \ \sum\nolimits_{p=1}^m \beta_p^2 \mathrm{Tr}(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^{\mathrm{T}}))$$
$$s.t. \ \boldsymbol{\beta}^{\mathrm{T}}\mathbf{1}_m = 1, \ \beta_p \geq 0. \quad (5)$$

Along with this line, many variants of MKKM have been proposed in the literature. The work in [Liu *et al.*, 2016] proposes a multiple kernel k-means clustering algorithm with matrix-induced regularization to reduce the redundancy and

enhance the diversity of the pre-defined kernels. Furthermore, local kernel alignment criterion has been applied to multiple kernel learning to enhance the clustering performance in [Li *et al.*, 2016].

Although demonstrating promising performance, we observe that most of the existing multi-view clustering methods extend the Eq. (1) into multi-view by assuming that $\mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^{m} \beta_p \mathbf{K}_p$ for instance and learn an optimal similarity representation $\mathbf{K}^*$ to perform kernel k-means clustering. As mentioned in [Liu *et al.*, 2018], this category causes intensive computational complexity , i.e., usually $\mathcal{O}(n^3)$ per iteration with n the number of samples and over-complicated optimization increasing the risk of trapped into low-quality local minimum. In the next section, we propose *Multi-view Clustering via Late Fusion Alignment Maximization(MVC-LFA)* to address this issue.

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

3.1 Proposed Formulation

According to the aforementioned discussions, we firstly show our theoretical result on the connection between the widely-used k-means loss function and our proposed late fusion alignment. In the following, we derive a new and related optimization goal of Eq. (1) for multi-view clustering based on the following Theorem 1.

Lemma 1. *For any given orthogonal matrix $\mathbf{P}$ with the size of $k \times k$, the inequality $\mathrm{Tr}^2(\mathbf{P}) \leq k \cdot \mathrm{Tr}(\mathbf{P}^{\mathrm{T}}\mathbf{P})$ always holds.*

Theorem 1. *Based on Lemma 1, minimizing Eq. (1) is conceptually equivalent to maximize $\mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X})$.*

Proof. By taking $\mathbf{P} = \mathbf{H}^{*\mathrm{T}}\mathbf{X}$ and according to Lemma 1, we can see that $\mathrm{Tr}^2(\mathbf{H}^{*\mathrm{T}}\mathbf{X}) \leq k \cdot \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{H}^*)$. As a result, $\mathrm{Tr}(\mathbf{X}\mathbf{X}^{\mathrm{T}}) - \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{H}^*) \leq 2k - \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{H}^*) \leq 2k - \frac{1}{k}\mathrm{Tr}^2(\mathbf{H}^{*\mathrm{T}}\mathbf{X})$. Therefore, minimizing $\mathrm{Tr}(\mathbf{X}\mathbf{X}^{\mathrm{T}}) - \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}\mathbf{Y}^{\mathrm{T}}\mathbf{H}^*)$ is conceptually equivalent to maximize $\mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X})$. This completes the proof. $\square$

Based on Theorem 1, we derive a simple but effective clustering loss function for clustering. This new formulation can be readily extended to handle multi-view clustering. To be specific, after obtaining the basic partitions $\{\mathbf{H}_p\}_{p=1}^{m}$ from each single view, we conduct the new optimal combinational data partition $\mathbf{X} = \sum_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p$ to maximally align with the optimal clustering partition. Compared with minimizing Eq. (1), maximizing $\mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X})$ is much easier to optimize since it only requires the singular value decomposition of $\mathbf{X}$ in contrast to $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ so that significantly reduces the time-complexity and simplifies the optimization procedure.

As a result, we obtain the objective function of our proposed algorithm as follows,

$$\max_{\mathbf{H}^*,\{\mathbf{W}_p\}_{p=1}^{m},\beta} \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}) + \lambda \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{M}),$$
$$s.t. \ \mathbf{H}^{*\mathrm{T}}\mathbf{H}^* = \mathbf{I}_k, \mathbf{W}_p^{\mathrm{T}}\mathbf{W}_p = \mathbf{I}_k, \quad (6)$$
$$\sum_{p=1}^{m} \beta_p{}^2 = 1, \beta_p \geq 0, \mathbf{X} = \sum_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p,$$

where $\{\mathbf{W}_p\}_{p=1}^{m}$ are a set of rotation matrices, $\mathbf{M}$ denotes the average partition region and λ is a trade-off parameter. The latter $\mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{M})$ is a regularization on the consensus partition to prevent $\mathbf{H}^*$ from being too far way from prior average partition. It is worth noting that we not only set an optimization goal for the multi-view clustering with late fusion, but also offer a new framework to fuse various clustering methods, which implies that any kind of ensemble clustering results can be applied to our framework. Moreover, as the following optimization process shows, the proposed function could be easily solved by an alternate algorithm with proved convergence.

3.2 Alternate Algorithm

Although the problem in Eq. (6) is a relaxed version, it is still troublesome to be solved by existing packages. In order to solve it, we design a three-step alternate algorithm with proved convergence, where each step could be easily solved by the existing off-the-shelf packages.

Optimization $\mathbf{H}^*$ with fixed $\{\mathbf{W}_p\}_{p=1}^{m}$ and β

With $\{\mathbf{W}_p\}_{p=1}^{m}$ and β being fixed, the optimization Eq. (6) could be rewritten as follows,

$$\max_{\mathbf{H}^*} \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{U}) \ s.t. \ \mathbf{H}^{*\mathrm{T}}\mathbf{H}^* = \mathbf{I}_k, \quad (7)$$

where $\mathbf{U} = \sum_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p + \lambda \mathbf{M}$. The problem in Eq. (7) could be easily solved by taking the singular value decomposition (SVD) of the given matrix $\mathbf{U}$. Here the following Theorem gives a closed-form solution for the problem in Eq. (7).

Theorem 2. *Suppose that the matrix $\mathbf{U}$ in Eq. (7) has the economic rank-k singular value decomposition form as $\mathbf{U} = \mathbf{S}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^{\top}$, where $\mathbf{S}_k \in \mathbb{R}^{n \times k}, \boldsymbol{\Sigma}_k \in \mathbb{R}^{k \times k}, \mathbf{V}_k \in \mathbb{R}^{k \times k}$. The optimization in Eq. (7) has a closed-form solution as follows,*

$$\mathbf{H}^* = \mathbf{S}_k \mathbf{V}_k^{\mathrm{T}}, \quad (8)$$

Proof. By taking the the normal singular value decomposition $\mathbf{U} = \mathbf{S}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}$, the Eq. (7) could be rewritten as,

$$\mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{S}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}) = \mathrm{Tr}(\mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}\boldsymbol{\Sigma}), \quad (9)$$

Considering that $\mathbf{Q} = \mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}$, we have $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}\mathbf{S}^{\mathrm{T}}\mathbf{H}^*\mathbf{V} = \mathbf{I}_k$. Therefore we can take $\mathrm{Tr}(\mathbf{V}^{\mathrm{T}}\mathbf{H}^{*\mathrm{T}}\mathbf{S}\boldsymbol{\Sigma}) = \mathrm{Tr}(\mathbf{Q}\boldsymbol{\Sigma}) \leq \sum_{i=1}^{k} \sigma_i$. Hence in order to maximize the value of Eq. (7), the solution should be given as Eq. (8). $\square$

Optimization $\{\mathbf{W}_p\}_{p=1}^{m}$ with fixed $\mathbf{H}^*$ and β

With $\mathbf{H}^*$ and β being fixed, for each single $\mathbf{W}_p$, the optimization problem in Eq. (6) is equivalent to Eq. (10) as follows,

$$\max_{\mathbf{W}_p} \ \mathrm{Tr}(\mathbf{W}_p^{\mathrm{T}}\mathbf{A}) \ s.t. \ \mathbf{W}_p^{\mathrm{T}}\mathbf{W}_p = \mathbf{I}_k, \quad (10)$$

where $\mathbf{A} = \beta_p \mathbf{H}_p^{\mathrm{T}}\mathbf{H}^*$. And this problem in Eq. (10) could be easily solved by taking the singular value decomposition (SVD) of the given matrix $\mathbf{V}$. Like the closed-form expressed

in Theorem 2, if the matrix $\mathbf{V}$ has the singular value decomposition form as $\mathbf{A} = \mathbf{S}\mathbf{\Sigma}\mathbf{G}^{\mathrm{T}}$, the optimization in Eq. (10) has a closed-form solution as $\mathbf{W}_p = \mathbf{S}\mathbf{G}^{\mathrm{T}}$. Therefore, we optimize one $\mathbf{W}_p$ with other $\mathbf{W}_{i \neq p}$ fixed at each iteration. As a result, we can obtain a set of optimized $\{\mathbf{W}_p\}_{p=1}^m$.

Optimization β with fixed $\mathbf{H}^*$ and $\{\mathbf{W}_p\}_{p=1}^m$

With $\mathbf{H}^*$ and $\{\mathbf{W}_p\}_{p=1}^m$ being fixed, the optimization problem in Eq. (6) is equivalent to the optimization problem as follows,

$$\max_\beta \sum\nolimits_{p=1}^m \beta_p \delta_p \quad s.t. \sum\nolimits_{p=1}^m {\beta_p}^2 = 1, \beta_p \geq 0, \quad (11)$$

where $\delta_p = \mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{H}_p\mathbf{W}_p)$. This could be easily solved with closed-form solution as follows,

$$\beta_p = \delta_p / \sqrt{\sum\nolimits_{p=1}^m \delta_p^2}. \quad (12)$$

In sum, our algorithm is outlined in Algorithm 1, where $\mathrm{obj}^{(t)}$ denotes the objective value at the t-th iteration. The following Theorem 3 shows our algorithm is guaranteed to converge.

Theorem 3. *The proposed algorithm 1 is proved to converge to a local optimum,*

Proof. Note that for $\forall p, q$, $\mathrm{Tr}[(\beta_p\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\beta_q\mathbf{H}_q\mathbf{W}_q)] \leq \mathrm{Tr}[(\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\mathbf{H}_q\mathbf{W}_q)] \leq \frac{1}{2}(\mathrm{Tr}[(\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\mathbf{H}_p\mathbf{W}_p)] + \mathrm{Tr}[(\mathbf{H}_q\mathbf{W}_q)^{\mathrm{T}}(\mathbf{H}_q\mathbf{W}_q)]) = k$. As a result, we could derive the upper bound of the optimization goal in Eq. (6). We obtain that $\mathrm{Tr}(\mathbf{H}^{*\mathrm{T}}\mathbf{X}) \leq \frac{1}{2}(\mathrm{Tr}[\mathbf{H}^{*\mathrm{T}}\mathbf{H}^*] + \mathrm{Tr}[\mathbf{X}^{\mathrm{T}}\mathbf{X}]) = \frac{1}{2}(\mathrm{Tr}[\mathbf{H}^{*\mathrm{T}}\mathbf{H}^*] + \mathrm{Tr}(\sum_{p,q=1}^m (\beta_p\mathbf{H}_p\mathbf{W}_p)^{\mathrm{T}}(\beta_q\mathbf{H}_q\mathbf{W}_q))) \leq \frac{k}{2}(m^2 + 1)$. Meanwhile, the $(\mathbf{H}^{*\mathrm{T}}\mathbf{M}) \leq \frac{1}{2}(\mathrm{Tr}[\mathbf{H}^{*\mathrm{T}}\mathbf{H}^*] + \mathrm{Tr}[\mathbf{M}^{\mathrm{T}}\mathbf{M}]) = k$. Consequently, the whole optimization function is upper bounded. As the three subproblem is strictly convex when optimizing one variable and keeping the others fixed. The objective of Algorithm 1 is monotonically increased when optimizing one variable with the others fixed at each iteration. At the same time, the whole optimization problem is upper-bounded. As a result, the proposed algorithm can be verified to be convergent. This completes the proof. $\square$

3.3 Discussion and Extensions

In this section, we analyze the computational complexity and potential extensions.

Computational Complexity: With the optimization process outlined in Algorithm 1, the computational complexity of MVC-LFA is $\mathcal{O}(nk^2 + mk^3)$ per iteration, where n, m, k represents the number of samples, views and clusters respectively. This implies that our algorithm has a linearly growing complexity with the number of samples, making it efficiently to handle large-scale tasks comparing to the state-of-the-art multiple-kernel clustering algorithms.

Algorithm 1 Multi-view Clustering via Late Fusion Alignment Maximization

Input: $\{\mathbf{H}_p\}_{p=1}^m$, k, λ and ϵ_0.
Output: $\mathbf{H}^*$ and β.
1: Initialize $\{\mathbf{W}_p\}_{p=1}^m = \mathbf{I_k}$, $\beta = \frac{1}{\sqrt{m}}$ and $t = 1$.
2: Calculate $\mathbf{M}$ by kernel k-means with average kernel.
3: **while** not converged **do**
4:     Update $\mathbf{H}^*$ by solving Eq. (7) with fixed $\{\mathbf{W}_p\}_{p=1}^m$ and β.
5:     Update $\{\mathbf{W}_p\}_{p=1}^m$ with fixed $\mathbf{H}^*$ and β by solving Eq. (10).
6:     Update β by solving Eq. (12) with fixed $\mathbf{H}^*$ and $\{\mathbf{W}_p\}_{p=1}^m$.
7:     $t = t + 1$.
8: **end while** $\left(\mathrm{obj}^{(t-1)} - \mathrm{obj}^{(t)}\right)/\mathrm{obj}^{(t)} \leq \epsilon_0$
9: **return** $\mathbf{H}^*$ and β.

Extensions: MVC-LFA can be easily extended with the following considerations. Firstly, MVC-LFA could be further improved by capturing the noises or low-quality partitions existing in basic partitions. For example, we could integrate the basic partitions $\{\mathbf{H}_p\}_{p=1}^m$ into the optimization procedure to capture more advanced base partitions. By doing so, the high-quality basic partitions are further used to guide the generation of consensus partition. Secondly, we could apply more similarity-based clustering methods to generate basic partitions. Exploring other generating methods and evaluating their clustering performance will be an interesting future work.

4 Experiments and Analysis

In this section, we evaluate the effectiveness and efficiency of the proposed MVC-LFA for five widely used multi-view benchmark datasets from the aspects of clustering performance, computational efficiency and convergence.

4.1 Experimental Settings

Datasets
The datasets used in our experiments are Oxford Flower17 and Flower102[1], and Protein fold prediction[2] and Columbia Consumer Video (CCV)[3] and Caltech 101[4]. The detailed information of the several datasets are listed in Table 1.

Compared Algorithm
In the experiments, MVC-LFA is compared with the following state-of-the-art multi-view clustering methods. (1) **Average multiple kernel k-means (A-MKKM)**: All kernels are averagely weighted to conduct the optimal kernel, which is used as the input of kernel k-means algorithm. (2) **Single best**

---

[1] http://www.robots.ox.ac.uk/~vgg/data/flowers/
[2] http://mkl.ucsd.edu/dataset/protein-fold-prediction
[3] http://www.ee.columbia.edu/ln/dvmm/CCV/
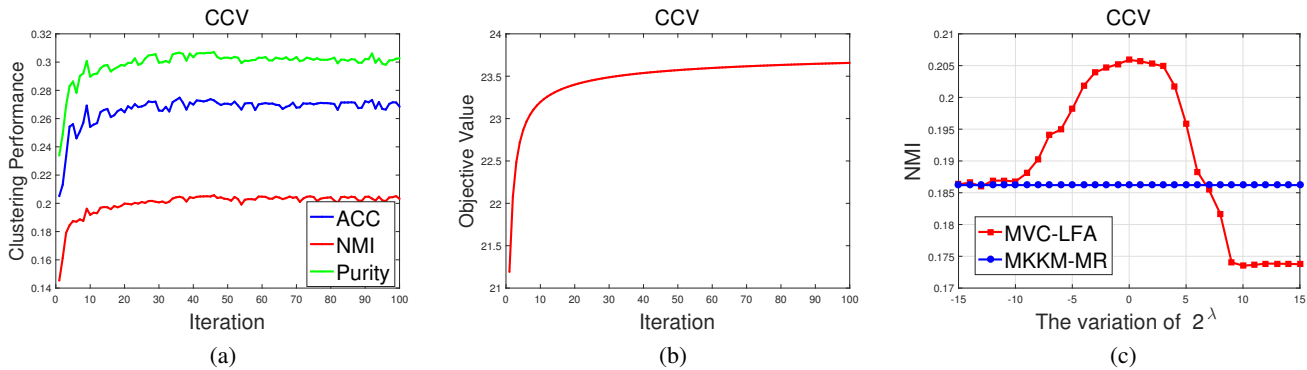[4] http://www.vision.caltech.edu/Image_Datasets/Caltech101

Figure 2: (a) The clustering performance at each iteration performed on CCV dataset. (b) The objective value at each iteration performed on CCV dataset. (c) The effect of λ on NMI on CCV dataset. The curves of other datasets are similar and are omitted due to space limit.

Dataset	#Samples	#Views	#Clusters
Flower17	1360	7	17
ProteinFold	694	12	27
Flower102	8189	4	102
Caltech	1530	25	102
CCV	6773	3	20

Table 1: Datasets used in our experiments.

kernel k-means (SB-KKM): Kernel k-means is performed on each single kernel and the best result is outputted. (3) **Multiple kernel k-means (MKKM)** ([Huang *et al.*, 2012]): The algorithm alternatively performs kernel k-means and updates the kernel coefficients. (4) **Optimized data fusion for kernel k-means clustering (OKKC)** ([Yu *et al.*, 2012]): The algorithm propose to jointly optimize clustering, the kernel coefficients and dimension reduction based on the metric of Mahalanobis distance. (5) **Co-regularized spectral clustering (CRSC)** ([Kumar *et al.*, 2011]): CRSC provides a co-regularization way to perform spectral clustering on multiple views. (6) **Multiple kernel k-means with matrix-induced regularization (MKKM-MR)** ([Liu *et al.*, 2016]): The algorithm applies the multiple kernel k-means clustering with a matrix-induced regularization to reduce the redundancy and enhance the diversity of the kernels. (7) **Multiple kernel clustering with local kernel alignment maximization (MKC-LKA)**([Li *et al.*, 2016]): The algorithm maximizes the local kernel with multiple kernel clustering and focuses on closer sample pairs that they shall stay together. (8) **Optimal neighborhood kernel clustering with multiple kernels (ONKC)**([Liu *et al.*, 2017]): ONKC allows the optimal kernel to reside in the neighborhood of linear combination of base kernels and effectively enlarges the region from which an optimal kernel can be chosen.

In all our experiments, all base kernels are first centered and then normalized so that for all sample x_i and p, we have $K_p(x_i, x_i) = 1$ by following [Cortes *et al.*, 2012]. For all data sets, it is assumed that the true number of clusters is known and set as the true number of classes. For the proposed algorithm, the trade-off parameter λ is chosen from $\left[2^{-15}, 2^{-14}, \cdots, 2^{15}\right]$ by grid search.

The widely used clustering accuracy (ACC), normalized mutual information (NMI) and purity are applied to evalu-

ate the clustering performance. For all algorithms, we repeat each experiment for 50 times with random initialization to reduce the effectiveness of randomness caused by k-means, and report the best result. All the experiments are performed on a desktop with Intel core i7-5820k CPU and 16G RAM.

4.2 Clustering Performance

The ACC, NMI and Purity of the compared algorithms on the five benchmark datasets are displayed in Table 2, where the best results are presented in red. From the results, we have the following observations: i) As a strong baseline, the recently proposed MKC-LKA ([Li *et al.*, 2016]) outperforms other early-fusion multiple-kernel clustering methods in comparison. For example, it exceeds the second best early-fusion method (MKKM-MR) by $1.6\%, 6.6\%, 1.4\%, 0.6\%, 4.5\%$ in terms of ACC on Flower17, ProteinFold, Flower102, Caltech and CCV respectively. These results verify the effectiveness of maximizing the kernel alignment in a local way. ii) The proposed algorithm MKC-LFA significantly and consistently outperforms MKC-LKA by $\mathbf{0.7\%}, \mathbf{2.9\%}, \mathbf{3.2\%}, \mathbf{6.5\%}, \mathbf{17.3\%}$ in terms of ACC on Flower17, ProteinFold, Flower102, Caltech and CCV, respectively. Table 2 also reports the comparison of NMI and purity. As can be seen, we also observe that our proposed algorithm outperforms all other methods in both NMI and purity.

In summary, the above experimental results have well demonstrated the effectiveness of our proposed MVC-LFA comparing to other state-of-the-art methods. We attribute the superiority of MVC-LFA as two aspects: i) MVC-LFA employs joint fusion to update weighted basic partitions and the consensus one and gets slightly higher performance than other methods. To be specific, when a higher quality consensus partition is obtained, we could further make full use of the high-quality partition to guide the weighted basics ones and improve the performance. ii) Comparing with the existing early-fusion methods, the proposed MVC-LFA fuses multiple kernel information in the partition level, which demonstrates the benefits of fusing high-level information. These two factors contributes to the significant improvements on clustering performance.

Datasets	A-MKKM	SB-KKM	MKKM [Huang *et al.*, 2012]	OKKC [Yu *et al.*, 2012]	CSRC [Kumar *et al.*, 2011]	MKC-LKA [Li *et al.*, 2016]	MKKM-MR [Liu *et al.*, 2016]	ONKC [Liu *et al.*, 2017]	Proposed
ACC(%)									
Flower17	51.03	42.06	45.37	44.85	51.76	60.69	59.69	60.88	**61.16**
ProteinFold	30.69	34.58	27.23	37.10	35.59	39.34	36.89	37.90	**40.49**
Flower102	27.29	33.13	21.96	22.32	38.60	40.84	40.24	37.32	**42.16**
Caltech	35.56	33.14	34.77	33.92	34.38	36.06	35.82	35.32	**38.39**
CCV	19.74	20.08	18.01	20.54	23.06	23.49	22.47	24.18	**27.56**
NMI(%)									
Flower17	50.19	45.14	45.35	45.85	53.19	57.27	57.11	58.58	**60.79**
ProteinFold	40.96	42.33	37.16	40.75	45.66	47.55	45.13	46.93	**48.96**
Flower102	46.32	48.99	42.30	43.28	54.95	57.60	57.27	58.13	**60.48**
Caltech	59.90	59.07	59.64	57.22	58.35	60.98	60.18	60.41	**61.65**
CCV	17.16	17.73	15.52	16.28	18.89	17.11	18.62	18.24	**20.59**
Purity(%)									
Flower17	51.99	44.63	46.84	45.00	53.68	61.79	60.03	61.64	**62.32**
ProteinFold	37.18	41.21	33.86	39.91	42.07	45.97	43.80	45.24	**46.85**
Flower102	32.28	38.74	27.61	28.12	45.04	48.21	46.39	47.64	**50.44**
Caltech	37.12	35.10	37.25	36.27	35.95	38.08	37.65	39.08	**40.28**
CCV	23.98	23.48	22.25	24.17	26.80	22.93	25.69	23.34	**30.71**

Table 2: ACC, NMI and purity comparison of different clustering algorithms on five benchmark data sets.

4.3 Running Time

To compare the computational efficiency of the proposed algorithms, we record the running time of various algorithms on these benchmark datasets and report them in Figure 3. As can be seen, MVC-LFA in red has the shortest running time on all datasets comparing to the-state-of-art multi-view methods (MKKM, OKKC, CRSC, MKC-LKA (in green), MKKM-MR and ONKC), demonstrating the computational efficiency of the proposed method. As theoretically demonstrated, MVC-LFA reduces the time complexity from $\mathcal{O}(n)^3$ to $\mathcal{O}(n)$ per iteration and avoid complicated optimization procedure. In sum, both the theoretical and the experimental results in Figure 3 have well demonstrated the computational advantage of MVC-LFA, making it efficient to handle with multi-view clustering.
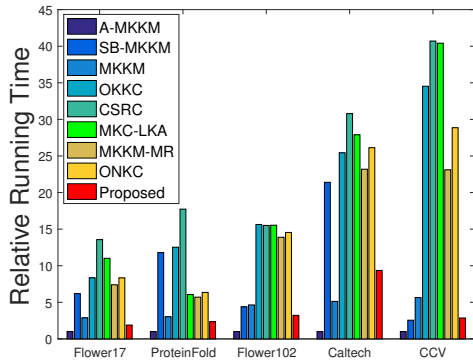

Figure 3: The running time comparison of different algorithms on five benchmark datasets.

4.4 Convergence and Parameter Sensitivity Study

Our algorithm is theoretically guaranteed to converge according to Theorem 3. For the experimental study, we conduct an experiment on CCV dataset and set the hyperparameters λ as 2^2. Specifically, we evaluate the ACC, NMI and Purity of consensus partition $\mathbf{H}^*$ learned at each iteration, as shown in Figure 2a. As can be observed, the clustering performance of MVC-LFA shown in Figure 2a gradually increases and has a slight variation with the increasing of iterations. This has clearly demonstrated the effectiveness of the learned consensus matrix $\mathbf{H}^*$. Furthermore, as shown in Figure 2b, the objective value of MVC-LFA does monotonically increase at each iteration and it usually converges in less than 10 iterations.

We also conduct the parameter sensitivity study on CCV dataset and report the clustering performance by ranging λ within the set of $\left[2^{-15}, 2^{-14}, \cdots, 2^{15}\right]$ shown in Figure 2c. From the observation, NMI first increases to a high value and generally maintains it up to slight variation with the increasing value of λ. The curves with other datasets are similar and committed due to space limit. Comparing to MKKM-MR, MVC-LFA demonstrates stable performance across a wide range of λ.

5 Conclusion

In this article, we propose a simple but effective method MVC-LFA, which maximizes the alignment between the consensus partition matrix and the weighted base partition matrices with orthogonal transformation. We theoretically uncover the connection between the existing k-means and the proposed alignment. Based on this, we derive a simple novel optimization goal for multi-view clustering, which significantly reduces the computational complexity and simplifies the optimization procedure. The proposed algorithm is developed to efficiently solve the resultant optimization. By the virtue of it, MVC-LFA shows clearly superior clustering performance with significant reduction in computational cost on benchmark datasets, in comparison with state-of-the-art methods. In the future, we will explore to capture the noises or bad partition existing in basic partitions and further improve the fusion method.

Acknowledgements

References

[Boutsidis *et al.*, 2015] Christos Boutsidis, Anastasios Zouzias, Michael W Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *IEEE Transactions on Information Theory*, 61(2):1045–1062, 2015.

[Chen *et al.*, 2007] Jianhui Chen, Zheng Zhao, Jieping Ye, and Huan Liu. Nonlinear adaptive distance metric learning for clustering. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–132. ACM, 2007.

[Cortes *et al.*, 2012] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(Mar):795–828, 2012.

[Gönen and Alpaydin, 2008] Mehmet Gönen and Ethem Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25th international conference on Machine learning*, pages 352–359. ACM, 2008.

[Gönen and Margolin, 2014] Mehmet Gönen and Adam A Margolin. Localized data fusion for kernel k-means clustering with application to cancer biology. In *Advances in Neural Information Processing Systems*, pages 1305–1313, 2014.

[Huang *et al.*, 2012] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1):120–134, 2012.

[Kumar *et al.*, 2011] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*, pages 1413–1421, 2011.

[Li *et al.*, 2016] Miaomiao Li, Xinwang Liu, Lei Wang, Yong Dou, Jianping Yin, and En Zhu. Multiple kernel clustering with local kernel alignment maximization. In *International Joint Conference on Artificial Intelligence*, pages 1704–1710, 2016.

[Liu *et al.*, 2013] Xinwang Liu, Lei Wang, Jianping Yin, En Zhu, and Jian Zhang. An efficient approach to integrating radius information into multiple kernel learning. *IEEE transactions on cybernetics*, 43(2):557–569, 2013.

[Liu *et al.*, 2016] Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, and En Zhu. Multiple kernel k -means clustering with matrix-induced regularization. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 1888–1894, 2016.

[Liu *et al.*, 2017] Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, and Jianping Yin.

Optimal neighborhood kernel clustering with multiple kernels. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[Liu *et al.*, 2018] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, Chang Tang, Jianping Yin, Dinggang Shen, Huaimin Wang, and Wen Gao. Late fusion incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

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

[Xia *et al.*, 2014] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[Yu *et al.*, 2012] Shi Yu, Leon Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan AK Suykens, Bart De Moor, and Yves Moreau. Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039, 2012.

[Zhou *et al.*, 2015] Peng Zhou, Liang Du, Lei Shi, Hanmo Wang, and Yi-Dong Shen. Recovery of corrupted multiple kernels for clustering. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.