One Pass Late Fusion Multi-view Clustering

Xinwang Liu [1] Li Liu [1] Qing Liao [2] Siwei Wang [1] Yi Zhang [1] Wenxuan Tu [1] Chang Tang [3]
Jiyuan Liu [1] En Zhu [1]

Abstract

Existing late fusion multi-view clustering (LFMVC) optimally integrates a group of pre-specified base partition matrices to learn a consensus one. It is then taken as the input of the widely used k-means to generate the cluster labels. As observed, the learning of the consensus partition matrix and the generation of cluster labels are separately done. These two procedures lack necessary negotiation and can not best serve for each other, which may adversely affect the clustering performance. To address this issue, we propose to unify the aforementioned two learning procedures into a single optimization, in which the consensus partition matrix can better serve for the generation of cluster labels, and the latter is able to guide the learning of the former. To optimize the resultant optimization problem, we develop a four-step alternate algorithm with proved convergence. We theoretically analyze the clustering generalization error of the proposed algorithm on unseen data. Comprehensive experiments on multiple benchmark datasets demonstrate the superiority of our algorithm in terms of both clustering accuracy and computational efficiency. It is expected that the simplicity and effectiveness of our algorithm will make it a good option to be considered for practical multi-view clustering applications.

1. Introduction

Multi-view clustering (MVC) maximally utilizes a group of pre-calculated complementary views to improve the clustering performance (Peng et al.; Wang et al.). It has been intensively studied and successfully applied into various applications (Huang et al.; Wang et al., 2019). According to different ways in fusing views, existing MVC can be roughly grouped into three categories: feature concatenation, multiple kernel clustering and late fusion MVC. The methods in the first category concatenate features from different views into a high-dimensional representation, which is then taken as the input of existing single view clustering algorithms to generate cluster labels. Though simple and computationally efficient, these methods usually demonstrate unsatisfying clustering performance since that the complementary information among different views cannot be sufficiently exploited.

By following the multiple kernel learning framework, the second category, i.e., multiple kernel clustering, firstly calculates a similarity (kernel) matrix based on each view, and then optimally combines these kernel matrices to learn an optimal kernel matrix for clustering (Zhao et al., 2009). Along this line, many variants have been developed (Yu et al., 2012; Gönen & Margolin, 2014; Liu et al., 2016b; Li et al., 2016). The work in (Yu et al., 2012) proposes a three-step alternate algorithm to jointly perform kernel clustering, coefficients optimization and dimension reduction. The work in (Gönen & Margolin, 2014) develops a localized multiple kernel k-means(MKKM) where the kernel weight for each sample is adaptive. In (Liu et al., 2016b), a matrix-induced regularization term is incorporated into existing MKKM to enhance the diversity and reduce the redundancy of the selected kernel matrices. Furthermore, a local kernel alignment criterion (Li et al., 2016) has been applied to multiple kernel learning to enhance the clustering performance in (Liu et al., 2016b). The methods in the second category have been intensively studied and shown superior clustering performance in various applications (Liu et al., 2017b). However, their computational complexity is usually cubic of sample number, which prohibits them from median or large-scale clustering tasks.

To alleviate the computational cost of multiple kernel clustering algorithms, the third category proposes a different paradigm for MVC, which is termed as late fusion MVC. Specifically, these methods firstly calculate a clustering partition matrix $\mathbf{H}_p$ by performing kernel k-means with $\mathbf{K}_p$, where $\mathbf{K}_p$ represents pairwise sample similarity of the p-th view. After that, a consensus matrix is learned from $\mathbf{H}_p$'s with linear computational complexity (Wang et al., 2019).

[1]School of Computer, National University of Defense Technology. [2]Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. [3]School of Computer Science, China University of Geosciences. Correspondence to: Xinwang Liu <xinwangliu@nudt.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

Besides significantly reducing the computational complexity of MKC, the methods in the last category usually demonstrate promising clustering performance in various applications (Wang et al., 2019). These advantages make the late fusion paradigm a representative in solving MVC.

Though late fusion based MVC algorithms achieve a significant improvement in terms of both clustering accuracy and computational complexity, we observe that the generation of cluster labels and the learning of consensus partition matrix are separately performed. Specifically, the learned consensus partition matrix is usually taken as the input of k-means to generate cluster labels. As seen, the learned consensus matrix by existing late fusion MVC methods may not best serve for the generation of the cluster labels, leading to unsatisfying clustering performance. This motivates us to design a novel MVC algorithm which unifies the learning of consensus matrix and the generation of cluster labels. To fulfill this goal, we propose to integrate the aforementioned two learning procedures into an unified optimization, in which the consensus partition matrix can better serve for the generation of cluster labels, and the latter is more conducive to guide the learning of the former. By this way, the two learning procedures can be seamlessly connected to attain a superior solution, leading to improved clustering performance. To optimize the resultant optimization problem, we develop a four-step alternate algorithm with proved convergence. Furthermore, we theoretically analyze the clustering generalization error of the proposed algorithm on unseen samples. Comprehensive experiments on multiple benchmark datasets demonstrate the superiority of our algorithm in terms of both clustering accuracy and computational efficiency.

2. Related Work

In this section, we briefly introduce the most related work to our study in this paper, including multiple kernel k-means and late fusion multi-view clustering.

2.1. Multiple kernel k-means (MKKM)

As an important learning paradigm in solving multi-view clustering, MKKM and its variants have been intensively studied. It is extended from the the widely used (kernel) k-means (Liu et al., 2017a). By assuming that the optimal kernel can be expressed as a linear/nonlinear combination of a group of pre-calculated kernel matrices, MKKM and its variants jointly learn the optimal combination coefficient of kernels and the clustering partition matrix.

Let $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ be a collection of n samples, and $\phi_p(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}_p$ be the p-th feature mapping that maps $\mathbf{x}$ onto a reproducing kernel Hilbert space $\mathcal{H}_p$ $(1 \leq p \leq m)$. In the multiple kernel setting, each sample is represented as $\phi_{\boldsymbol{\beta}}(\mathbf{x}) = [\beta_1\phi_1(\mathbf{x})^\top, \cdots, \beta_m\phi_m(\mathbf{x})^\top]^\top$, where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_m]^\top$ consists of the coefficients of the m base kernels $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$. Based on the definition of $\phi_{\boldsymbol{\beta}}(\mathbf{x})$, a kernel function can be expressed as

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

A kernel matrix $\mathbf{K}_{\boldsymbol{\beta}}$ is then calculated by applying the kernel function $\kappa_{\boldsymbol{\beta}}(\cdot,\cdot)$ into $\{\mathbf{x}_i\}_{i=1}^n$. Based on the kernel matrix $\mathbf{K}_{\boldsymbol{\beta}}$, the objective of MKKM can be written as

$$\begin{aligned} \min\nolimits_{\mathbf{H},\,\boldsymbol{\beta}} \ & \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k,\ \boldsymbol{\beta}^\top\mathbf{1}_m = 1,\ \beta_p \geq 0,\ \forall p, \end{aligned} \tag{2}$$

where $\mathbf{I}_k$ is an identity matrix with size $k \times k$.

In literature, the optimization problem in Eq. (2) can be solved by alternately updating $\mathbf{H}$ and $\boldsymbol{\beta}$:

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

$$\max\nolimits_{\mathbf{H}} \ \mathrm{Tr}(\mathbf{H}^\top\mathbf{K}_{\boldsymbol{\beta}}\mathbf{H}),\ s.t.\ \mathbf{H} \in \mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k. \tag{3}$$

The optimal $\mathbf{H}$ for Eq. (3) can be obtained by taking the k eigenvectors with respect to the largest k eigenvalues of $\mathbf{K}_{\boldsymbol{\beta}}$.

ii) **Optimizing $\boldsymbol{\beta}$ given $\mathbf{H}$.** With $\mathbf{H}$ fixed, $\boldsymbol{\beta}$ can be optimized via solving the following quadratic programming with linear constraints,

$$\begin{aligned} \min\nolimits_{\boldsymbol{\beta}} \ & \sum\nolimits_{p=1}^m \beta_p^2 \mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right), \\ s.t. \ & \boldsymbol{\beta}^\top\mathbf{1}_m = 1,\ \beta_p \geq 0. \end{aligned} \tag{4}$$

As seen from the aforementioned optimization in Eq. (3), existing MKKM and its variants need to solve a eigndecomposition at each iteration, suffering from intensive computational burden.

2.2. Late Fusion Multi-view Clustering

Late fusion multi-view clustering has recently been proposed to reduce the computational complexity. Based on the assumption that multiple views are expected to share a consensus partition matrix at partition level, it seeks an optimal partition by combing linearly-transformed base partitions obtained from single views (Wang et al., 2019). Given n samples in k clusters among m views, its optimization goal can be mathematically expressed as,

$$\begin{aligned} \max\nolimits_{\mathbf{H},\{\mathbf{W}_p\}_{p=1}^m,\boldsymbol{\beta}} \ & \mathrm{Tr}(\mathbf{H}^\top \sum\nolimits_{p=1}^m \beta_p\mathbf{H}_p\mathbf{W}_p) \\ s.t. \ & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \mathbf{W}_p^\top\mathbf{W}_p = \mathbf{I}_k, \\ & \sum\nolimits_{p=1}^m \beta_p^2 = 1, \beta_p \geq 0,\ \forall i, \end{aligned} \tag{5}$$

where the objective denotes the alignment between the consensus partition matrix $\mathbf{H}$ and a group of pre-calculated base partition matrices $\{\mathbf{H}_p\}_{p=1}^m$, and $\mathbf{W}_p$ is the p-th transformation matrix. A three-step optimization procedure with proved convergence is developed to solve the optimization in Eq. (5). According to the analysis in (Wang et al., 2019), the computational complexity of late fusion MVC is linear in the number of samples, which enables it to handle with large-scale cluster tasks.

In existing late fusion MVC (Wang et al., 2019), the learned consensus partition matrix is usually taken into k-means to generate cluster labels. As seen, both procedures are separately done without negotiation, which makes the learned consensus matrix may not best serve for the generation of cluster labels. In the following part, we develop the one pass late fusion multi-view clustering algorithm (OP-LFMVC) to address the above issue.

3. One Pass Late Fusion Multi-view Clustering (OP-LFMVC)

In this section, we first give the objective of the proposed OP-LFMVC, and then develop a four-step algorithm to alternately solve the resultant optimization problem. After that, we discuss the convergence, computational complexity and extension of the proposed algorithm.

3.1. The Proposed Formulation of OP-LFMVC

Eq. (5) is a widely used criterion in late fusion MVC due to its simplicity and effectiveness (Wang et al., 2019). Though demonstrating promising clustering performance in some applications, we observe that it has to discretize the learned consensus partition matrix $\mathbf{H}$ to generate clustering labels. This implies that these two procedures lack of negotiation to achieve optimality. To address the above issue, we propose an one pass late fusion multi-view clustering algorithm which directly learns the discrete clustering labels. To do so, we firstly decompose the consensus clustering partition matrix $\mathbf{H}$ as,

$$\mathbf{H} = \mathbf{Y}\mathbf{C}, \tag{6}$$

where $\mathbf{Y} \in \{0,1\}^{n\times k}$ is the cluster label matrix, and $\mathbf{C} \in \mathbb{R}^{k\times k}$ is the k centroids. Note that each row of $\mathbf{Y}$ has one element as 1 and others 0.

By incorporating Eq. (6) into Eq. (5), we obtain the formulation of our proposed OP-LFMVC as follows,

$$\begin{aligned}\max_{\mathbf{Y},\mathbf{C},\{\mathbf{W}_p\}_{p=1}^m,\boldsymbol{\beta}} \ & \mathrm{Tr}\left(\mathbf{C}^\top\mathbf{Y}^\top\sum\nolimits_{p=1}^m \beta_p\mathbf{H}_p\mathbf{W}_p\right) \\ s.t.\ & \mathbf{C}^\top\mathbf{C}=\mathbf{I}_k,\ \mathbf{W}_p^\top\mathbf{W}_p=\mathbf{I}_k,\ \forall p, \\ & \mathbf{Y}\in\{0,1\}^{n\times k},\ \sum\nolimits_{p=1}^m\beta_p^2=1,\ \beta_p\geq 0,\end{aligned} \tag{7}$$

where an extra orthogonal constraint is imposed on $\mathbf{C}$ to make the optimization bounded. As seen from Eq. (7), instead of learning a consensus matrix $\mathbf{H}$ as in Eq. (6), our algorithm optimizes the cluster labels directly. By this way, the learning of cluster labels and clustering can be negotiated with each other to achieve optimality, leading to improved clustering performance.

3.2. Alternate Optimization

There are four variables in Eq. (7) to be optimized. Simultaneously optimizing them is difficult. In the following, we design a four-step optimization procedure to alternately solve it. In each step, one variable is optimized with others fixed.

Optimization $\mathbf{Y}$ Fixing $\boldsymbol{\beta}$, $\{\mathbf{W}_p\}_{p=1}^m$ and $\mathbf{C}$, the optimization in Eq. (7) w.r.t $\mathbf{Y}$ is transformed to,

$$\max_{\mathbf{Y}} \ \mathrm{Tr}\left(\mathbf{Y}\mathbf{B}^\top\right)\ \ s.t.\ \ \mathbf{Y}\in\{0,1\}^{n\times k}, \tag{8}$$

where

$$\mathbf{B}=\sum\nolimits_{p=1}^m\beta_p\mathbf{H}_p\mathbf{W}_p\mathbf{C}^\top. \tag{9}$$

Therefore, the optimal $\mathbf{Y}$ for Eq. (8) is

$$\mathbf{Y}(i,j)=1, \tag{10}$$

where $j=\arg\max\mathbf{B}(i,:)$. As a result, the computational complexity of optimizing $\mathbf{Y}$ is $\mathcal{O}(n)$.

Optimization $\mathbf{C}$ Fixing $\boldsymbol{\beta}$, $\{\mathbf{W}_p\}_{p=1}^m$ and $\mathbf{Y}$, the optimization in Eq. (7) w.r.t $\mathbf{C}$ is reduced to

$$\max_{\mathbf{C}}\ \mathrm{Tr}\left(\mathbf{C}^\top\mathbf{A}\right)\ \ s.t.\ \ \mathbf{C}^\top\mathbf{C}=\mathbf{I}_k, \tag{11}$$

where

$$\mathbf{A}=\mathbf{Y}^\top\sum\nolimits_{p=1}^m\beta_p\mathbf{H}_p\mathbf{W}_p. \tag{12}$$

Eq. (11) can be efficiently solved by SVD with computational complexity $\mathcal{O}(nk^2)$.

Optimization $\mathbf{W}_p$ Fixing $\boldsymbol{\beta}$, $\mathbf{Y}$ and $\mathbf{C}$, the optimization in Eq. (7) w.r.t each $\mathbf{W}_p$ can be rewritten as,

$$\max_{\mathbf{W}_p}\ \mathrm{Tr}(\mathbf{W}_p^\top\mathbf{H}_p^\top\mathbf{Y}\mathbf{C})\ \ s.t.\ \ \mathbf{W}_p^\top\mathbf{W}_p=\mathbf{I}_k. \tag{13}$$

Similar to Eq. (11), it can be efficiently solved via SVD with computational complexity $\mathcal{O}(nk^2)$.

Optimization $\boldsymbol{\beta}$ Fixing $\mathbf{Y}$, $\mathbf{C}$ and $\{\mathbf{W}_p\}_{p=1}^m$, the optimization in Eq. (7) w.r.t $\boldsymbol{\beta}$ is equivalently rewritten as

$$\max_{\boldsymbol{\beta}}\ \sum\nolimits_{p=1}^m\beta_p\alpha_p\ \ s.t.\ \ \sum\nolimits_{p=1}^m\beta_p^2=1,\ \beta_p\geq 0, \tag{14}$$

Algorithm 1 One Pass Late Fusion Multi-view Clustering

Input: $\{\mathbf{H}_p\}_{p=1}^m,\ k,\ t=1.$
Initialize $\boldsymbol{\beta} = \mathbf{1}/\sqrt{m}$, $\{\mathbf{W}_p\}_{p=1}^m$, $\mathbf{C}$, flag $=1$.
while flag **do**
  update $\mathbf{Y}$ by optimizing Eq. (8);
  update $\mathbf{C}$ by optimizing Eq. (11);
  update $\{\mathbf{W}_p\}_{p=1}^m$ by optimizing Eq. (13);
  update $\boldsymbol{\beta}$ by optimizing Eq. (14);
  if $(\mathrm{obj}^{(t)} - \mathrm{obj}^{(t-1)})/\mathrm{obj}^{(t)} \leq 1e-3$ **then**
    flag=0.
  end if
  $t \leftarrow t+1$.
end while

where

$$\alpha_p = \mathrm{Tr}\left(\mathbf{C}^\top \mathbf{Y}^\top \mathbf{H}_p \mathbf{W}_p\right). \tag{15}$$

The optimal solution for Eq. (14) is

$$\beta_p = \alpha_p / \sqrt{\sum\nolimits_{q=1}^m \alpha_q^2}. \tag{16}$$

The whole optimization procedure in solving Eq. (7) is outlined in Algorithm 1, where $\mathrm{obj}^{(t)}$ indicates the objective value at the t-th iteration.

3.3. Discussion

Convergence Note that the objective value in Eq. (7) is monotonically increased when one variable is optimized with the others fixed. Moreover, our objective is upper-bounded. As a result, the optimization procedure in solving Eq. (7) is theoretically guaranteed to be (locally) convergent, as validated by our experimental results in Figure 1.

Computational Complexity According to the optimization procedure in Algorithm 1, the computational complexity of our algorithm at each iteration is $\mathcal{O}(n+nk^2+mnk^2)$, which is linear to the number of samples. Further, optimizing $\{\mathbf{W}_p\}_{p=1}^m$ can be trivially implemented in a parallel way since each optimization w.r.t $\mathbf{W}_p$ is independent. This could further reduce the computational complexity. The computational efficiency enables our algorithm to handle with large-scale clustering tasks.

Extension The idea of learning the cluster labels, instead of the consensus partition matrix, is not limited to late fusion MVC. In fact, it can be readily extended to multiple kernel clustering. Moreover, some prior knowledge could incorporated into the formulation of OP-LFMVC to further improve the clustering performance. Our work provides a more effective paradigm to fuse multi-view data for clustering, which could trigger novel research on MVC.

4. The Theoretical Results

Generalization error for k-means clustering has been studied by fixing the centroids obtained in the training process and computing their generalization to unseen data (Maurer & Pontil, 2010; Liu et al., 2016a). In this section, we study how the centroids obtained by the proposed OP-LFMVC generalizes onto test data by deriving its generalization bound.

We now define the error of OP-LFMVC. Let $\hat{\mathbf{C}} = [\hat{\mathbf{C}}_1, \cdots, \hat{\mathbf{C}}_k]$ be the learned matrix composed of the k centroids, $\hat{\boldsymbol{\beta}}$ the learned kernel weights and $\{\hat{\mathbf{W}}_p\}_{p=1}^m$ the transformation matrices learned by the proposed OP-LFMVC. By defining $\Theta = \{\mathbf{e}_1, \cdots, \mathbf{e}_k\}$, effective OP-LFMVC should make the following error small,

$$1 - \mathbb{E}_{\mathbf{x}}\left[\max\nolimits_{\mathbf{y}\in\Theta}\left\langle \sum\nolimits_{p=1}^m \hat{\beta}_p \hat{\mathbf{W}}_p^\top \mathbf{h}_p(\mathbf{x}), \hat{\mathbf{C}}\mathbf{y}\right\rangle\right], \tag{17}$$

where $\mathbf{h}_p(\mathbf{x})$ denotes the p-th partition vector corresponding to the p-th view of $\mathbf{x}$ with $\|\mathbf{h}_p(\mathbf{x})\| = 1$, and $\mathbf{e}_1, \cdots, \mathbf{e}_k$ form the orthogonal bases of $\mathbb{R}^k$. Intuitively, it says that the expected alignment between test points and their closest centroid should be high. We show how the proposed algorithm achieves this goal.

Let us define a function class first:

$$\begin{aligned}\mathcal{F} = \Big\{ f:\ \mathbf{x} \mapsto 1 - \max\nolimits_{\mathbf{y}\in\Theta}\left\langle \sum\nolimits_{p=1}^m \beta_p \mathbf{W}_p^\top \mathbf{h}_p(\mathbf{x}), \mathbf{C}\mathbf{y}\right\rangle \Big| \\ \sum\nolimits_{p=1}^m \beta_p^2 = 1,\ \beta_p \geq 0, \mathbf{C}\in\mathbb{R}^{k\times k},\ \mathbf{C}^\top\mathbf{C} = \mathbf{I}_k, \\ \mathbf{W}_p \in \mathbb{R}^{k\times k},\ \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_k,\ \forall p \Big\}.\end{aligned} \tag{18}$$

We have the following claim on the generalization error bound for the proposed OP-LFMVC.

Theorem 1. *For any $\delta > 0$, with probability at least $1-\delta$, the following inequality holds for all $f \in \mathcal{F}$,*

$$\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] \leq \frac{1}{n}\sum_{i=1}^n f(\mathbf{x}_i) + \frac{\sqrt{\pi/2}k}{\sqrt{n}} + 2\sqrt{\frac{\log 1/\delta}{2n}}. \tag{19}$$

The detailed proof is provided in the appendix due to the space limit.

According to Theorem 1, for any learned $\hat{\gamma}$, $\hat{\mathbf{C}} = [\hat{\mathbf{C}}_1, \cdots, \hat{\mathbf{C}}_k]$ and $\{\hat{\mathbf{W}}_p\}_{p=1}^m$, to achieve a small,

$$\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] = 1 - \mathbb{E}_{\mathbf{x}}\left[\max\nolimits_{\mathbf{y}\in\Theta}\left\langle \sum\nolimits_{p=1}^m \hat{\beta}_p \hat{\mathbf{W}}_p^\top \mathbf{h}_p(\mathbf{x}), \hat{\mathbf{C}}\mathbf{y}\right\rangle\right], \tag{20}$$

the corresponding $\frac{1}{n}\sum_{i=1}^n f(\mathbf{x}_i)$ needs to be as small as possible. Assume that $\boldsymbol{\beta}$, $\{\mathbf{W}_p\}_{p=1}^m$ and $\mathbf{C}$ are obtained by

Table 1. Datasets used in our experiments.

Dataset	Number of Samples	Number of Kernels	Number of Clusters
3Sources	169	3	6
Football	248	9	20
Olympics	464	9	29
BBCSport	544	2	5
Cal-20	2386	6	20
Cora	2708	2	7
Citeseer	3312	2	6
SUNRGBD	10335	2	45

minimizing $\frac{1}{n}\sum_i^n f(\mathbf{x}_i)$, we have,

$$\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i) = 1 - \frac{1}{n}\sum_{i=1}^{n} \max_{\mathbf{y}\in\Theta} \left\langle \sum_{p=1}^{m} \beta_p \mathbf{W}_p^\top \mathbf{h}_p(\mathbf{x}_i), \mathbf{C}\mathbf{y} \right\rangle$$
$$= 1 - \max_{\mathbf{Y}} \frac{1}{n}\mathrm{Tr}\left(\mathbf{Y}^\top \sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p \mathbf{C}^\top\right), \tag{21}$$

where the last equality holds since the optimal $\mathbf{y} \in \boldsymbol{\Theta}$ for each sample $\mathbf{x}_i$ is independent. This means that $1 - \max_{\mathbf{Y}} \frac{1}{n}\mathrm{Tr}\left(\mathbf{Y}^\top \sum_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p \mathbf{C}^\top\right)$ is an upper bound of $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$. To minimize the upper bound, we have to maximize over $\boldsymbol{\beta}$, $\{W_p\}_{p=1}^m$ and $\mathbf{C}$, leading to $\max_{\mathbf{Y},\mathbf{C},\{\mathbf{W}_p\},\boldsymbol{\beta}} \mathrm{Tr}(\mathbf{Y}^\top \sum_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p \mathbf{C}^\top)$, which is exactly the objective of the proposed algorithm in Eq. (7). This well justifies the effectiveness of our objective.

5. Experimental Results

In this section, we conduct a comprehensive experimental study to evaluate the proposed OP-LFMVC in terms of overall clustering performance comparison, convergence analysis, the evolution of the learned $\mathbf{Y}$ with iterations and running time comparison. In addition, we design an ablation experiment to clearly demonstrate the effectiveness of jointly learning the cluster labels.

5.1. Experimental Settings

We conduct experimental comparison on a number of publicly available multi-view benchmark datasets, including *3Sources*[1], *Football*[2], *Olympics*[3], *BBCSport*[4], *Cal-20*[5], *Cora*[6], *Citeseer*[7], *SUNRGBD*[8]. These dataset information is summarized in Table 1. As observed, the number of samples, kernels and categories of these datasets show considerable variation, providing a good platform to compare the performance of different clustering algorithms.

For all datasets, it is assumed that the true number of clusters k is known and set as the true number of classes. The clustering performance of all algorithms is evaluated by four widely used metrics: clustering accuracy (ACC), normalized mutual information (NMI), purity and rand index. For all compared algorithms, to alleviate the adverse influence of randomness by k-means, we repeat each experiment for 50 times and report the average values and the corresponding standard deviations. The highest and those with no statistical difference with it are marked in bold.

In our experiments, we compare OP-LFMVC with several state-of-the-art multi-view clustering methods, including:

- **Average kernel k-means (Avg-KKM)**. All kernels are averagely weighted to construct the optimal kernel, which is used as the input of kernel k-means algorithm.
- **Multiple kernel k-means (MKKM)** (Huang et al., 2012). The algorithm alternatively performs kernel k-means and updates the kernel coefficients.
- **Localized multiple kernel k-means(LMKKM)** (Gönen & Margolin, 2014). LMMKM combines the base kernels by sample-adaptive weights.
- **Optimal neighborhood kernel clustering (ONKC)** (Liu et al., 2017b). The consensus kernel is chosen from the neighbor of linearly combined base kernels.
- **Multiple kernel k-means with matrix-induced regularization (MKKM-MiR)** (Liu et al., 2016b). The optimal combination weights are learned by introducing a matrix-induced regularization term to reduce the redundancy among the base kernels.
- **Mulitple kernel clustering with local alignment maximization (LKAM)** (Li et al., 2016). The similarity of a sample to its k-nearest neighbors, instead of all samples, is aligned with the ideal similarity matrix.
- **Multi-view clustering via late fusion alignment maximization (LF-MVC)** (Wang et al., 2019). Base partitions are first computed within corresponding data views and then integrated into a consensus partition.
- **MKKM-MM** (Bang et al., 2018). It proposes a min-max formulation that combines views in a way to reveal high within-cluster variance in the combined kernel space and then updates clusters by minimizing such variance.
- **SMKKM** (Liu et al., 2020). It extends the widely used supervised kernel alignment criterion to multiple kernel clustering, and introduces a novel clustering objective by minimizing alignment for the kernel coefficient and maximizing it for the clustering partition matrix.

[1] http://mlg.ucd.ie/datasets/3sources.html
[2] http://mlg.ucd.ie/aggregation/
[3] http://mlg.ucd.ie/aggregation/
[4] http://mlg.ucd.ie/datasets/segment.html
[5] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[6] https://linqs-data.soe.ucsc.edu/public/lbc/
[7] https://linqs-data.soe.ucsc.edu/public/lbc/
[8] http://rgbd.cs.princeton.edu/

Table 2. Empirical evaluation and comparison of OP-LFMVC with nine baseline methods on eight datasets in terms of clustering accuracy (ACC), normalized mutual information (NMI), Purity and rand index (RI). Boldface means no statistical difference from the best one.

Dataset	Avg-KKM	MKKM	LMKKM	ONKC	MKKM-MiR	LKAM	LF-MVC	MKKM-MM	SMKKM	OP-LFMVC
					ACC					
3Sources	40.5 ± 2.2	40.4 ± 2.3	32.4 ± 1.6	40.8 ± 2.1	39.8 ± 2.2	28.5 ± 0.6	50.5 ± 1.1	40.5 ± 2.2	34.9 ± 2.7	**60.8 ± 5.3**
Football	73.6 ± 1.9	73.0 ± 2.8	51.7 ± 3.4	74.0 ± 3.5	76.1 ± 3.4	57.3 ± 2.3	**80.8 ± 3.3**	73.6 ± 1.9	70.4 ± 2.7	**82.2 ± 4.5**
Olympics	63.1 ± 3.2	62.1 ± 3.1	61.1 ± 2.4	67.7 ± 2.8	64.8 ± 2.9	35.2 ± 1.7	68.1 ± 3.5	63.1 ± 3.2	66.2 ± 2.7	**74.1 ± 2.5**
BBCSport	39.5 ± 0.7	39.4 ± 0.7	39.1 ± 0.4	39.7 ± 0.6	39.4 ± 0.7	28.4 ± 0.5	50.2 ± 4.7	39.5 ± 0.7	39.4 ± 0.7	**58.6 ± 4.5**
Caltech20	36.2 ± 2.2	29.5 ± 1.2	28.7 ± 1.3	39.4 ± 2.7	39.5 ± 1.9	37.5 ± 1.5	39.2 ± 2.0	36.2 ± 2.2	38.9 ± 2.4	**45.4 ± 2.6**
Cora	30.7 ± 0.8	25.3 ± 0.4	22.5 ± 0.2	35.2 ± 0.1	35.7 ± 0.1	26.4 ± 0.3	40.9 ± 0.1	30.7 ± 0.8	35.7 ± 0.1	**44.1 ± 3.0**
Citeseer	20.8 ± 0.0	20.1 ± 0.0	20.6 ± 0.0	40.3 ± 2.3	41.3 ± 0.1	23.1 ± 0.0	40.6 ± 0.1	20.8 ± 0.0	40.5 ± 2.4	**47.1 ± 1.1**
SUNRGBD	18.2 ± 0.5	17.4 ± 0.5	-	16.2 ± 0.5	14.5 ± 0.4	12.6 ± 0.6	17.8 ± 0.4	18.2 ± 0.5	19.2 ± 0.5	**20.5 ± 0.7**
					NMI					
3Sources	30.5 ± 1.7	30.9 ± 2.4	15.1 ± 1.1	30.6 ± 1.6	29.9 ± 1.4	14.0 ± 1.2	48.9 ± 2.5	30.5 ± 1.7	18.1 ± 4.7	**52.9 ± 4.3**
Football	78.6 ± 1.3	78.9 ± 1.2	59.4 ± 2.1	79.2 ± 1.9	79.6 ± 1.4	64.1 ± 2.1	85.4 ± 2.0	78.6 ± 1.3	75.9 ± 1.2	**87.3 ± 2.6**
Olympics	73.0 ± 1.4	72.5 ± 1.4	71.1 ± 1.5	76.0 ± 1.4	73.7 ± 1.2	49.9 ± 1.1	77.3 ± 1.6	73.0 ± 1.4	74.5 ± 1.5	**80.3 ± 1.3**
BBCSport	15.7 ± 0.5	15.7 ± 0.5	15.4 ± 0.3	16.1 ± 0.4	15.7 ± 0.5	3.1 ± 0.2	31.6 ± 4.2	15.7 ± 0.5	15.7 ± 0.5	**43.3 ± 3.0**
Caltech20	49.5 ± 1.1	37.9 ± 0.6	38.8 ± 0.5	**54.6 ± 0.8**	**54.2 ± 0.6**	52.1 ± 0.8	52.2 ± 0.8	49.5 ± 1.1	52.5 ± 1.2	**54.1 ± 1.0**
Cora	15.7 ± 1.4	9.5 ± 0.2	6.7 ± 0.3	16.9 ± 0.1	18.9 ± 0.2	9.2 ± 0.1	**26.6 ± 0.1**	15.7 ± 1.4	18.8 ± 0.2	**25.3 ± 2.1**
Citeseer	2.3 ± 0.0	1.9 ± 0.0	1.6 ± 0.0	18.6 ± 1.2	18.9 ± 0.1	4.0 ± 0.0	18.9 ± 0.1	2.3 ± 0.0	18.1 ± 1.7	**22.6 ± 0.9**
SUNRGBD	**22.6 ± 0.3**	21.3 ± 0.3	-	19.5 ± 0.3	17.8 ± 0.2	16.3 ± 0.3	**22.6 ± 0.2**	**22.6 ± 0.3**	**23.3 ± 0.3**	21.2 ± 0.4
					Purity					
3Sources	55.9 ± 2.1	56.4 ± 2.8	48.4 ± 1.4	55.8 ± 2.0	55.3 ± 1.5	48.3 ± 1.3	**72.0 ± 2.2**	55.9 ± 2.1	50.0 ± 3.9	**72.4 ± 3.2**
Football	75.7 ± 1.7	75.7 ± 1.9	55.1 ± 3.3	75.8 ± 3.0	77.9 ± 2.3	60.3 ± 2.0	**83.6 ± 2.6**	75.7 ± 1.7	72.3 ± 1.9	**84.5 ± 4.0**
Olympics	71.1 ± 2.0	70.8 ± 2.1	68.9 ± 1.9	75.9 ± 2.0	73.3 ± 1.7	45.8 ± 1.4	76.8 ± 2.4	71.1 ± 2.0	74.6 ± 2.1	**79.8 ± 2.2**
BBCSport	48.9 ± 0.3	48.9 ± 0.2	48.7 ± 0.3	49.1 ± 0.1	48.8 ± 0.2	35.6 ± 0.2	62.0 ± 2.9	48.9 ± 0.3	48.9 ± 0.3	**69.7 ± 2.8**
Caltech20	72.0 ± 1.5	62.9 ± 0.7	63.4 ± 0.8	**75.7 ± 0.8**	**75.4 ± 0.5**	74.0 ± 1.0	74.0 ± 1.2	72.0 ± 1.5	74.3 ± 1.1	**76.0 ± 1.4**
Cora	41.5 ± 1.3	36.1 ± 1.0	35.0 ± 0.2	43.4 ± 0.1	47.0 ± 0.1	35.1 ± 0.1	**51.9 ± 0.0**	41.5 ± 1.3	47.0 ± 0.1	**51.3 ± 2.5**
Citeseer	24.9 ± 0.0	24.2 ± 0.0	23.7 ± 0.0	42.9 ± 2.5	43.8 ± 0.1	26.0 ± 0.0	44.1 ± 0.1	24.9 ± 0.0	42.8 ± 2.3	**49.0 ± 1.2**
SUNRGBD	**38.0 ± 0.7**	36.2 ± 0.5	-	34.2 ± 0.4	32.3 ± 0.4	29.0 ± 0.6	**38.0 ± 0.5**	**38.0 ± 0.7**	**38.7 ± 0.4**	36.6 ± 0.6
					Rand Index					
3Sources	19.3 ± 2.7	20.0 ± 3.4	8.5 ± 1.2	19.6 ± 2.5	18.5 ± 1.9	7.2 ± 0.8	34.2 ± 1.5	19.3 ± 2.7	10.6 ± 3.9	**44.2 ± 6.0**
Football	61.1 ± 2.2	60.3 ± 2.2	31.6 ± 3.5	62.0 ± 3.1	63.5 ± 3.0	38.6 ± 3.0	71.0 ± 3.5	61.1 ± 2.2	56.8 ± 2.5	**74.6 ± 4.8**
Olympics	47.9 ± 3.0	47.0 ± 3.1	44.6 ± 2.5	55.4 ± 2.9	52.1 ± 2.5	20.5 ± 1.3	57.0 ± 3.3	47.9 ± 3.0	53.5 ± 2.7	**64.6 ± 2.8**
BBCSport	9.3 ± 0.3	9.2 ± 0.3	9.1 ± 0.2	9.6 ± 0.2	9.2 ± 0.3	1.5 ± 0.1	21.5 ± 3.2	9.3 ± 0.3	9.3 ± 0.3	**32.6 ± 4.4**
Caltech20	26.0 ± 1.4	18.6 ± 0.8	19.5 ± 1.0	28.9 ± 1.6	28.6 ± 1.1	25.6 ± 1.1	27.3 ± 1.2	26.0 ± 1.4	28.0 ± 1.8	**33.1 ± 2.0**
Cora	6.5 ± 0.6	3.6 ± 0.3	1.7 ± 0.1	11.7 ± 0.1	11.4 ± 0.1	5.3 ± 0.1	17.3 ± 0.1	6.5 ± 0.6	11.4 ± 0.1	**18.6 ± 2.2**
Citeseer	0.6 ± 0.0	0.3 ± 0.0	0.1 ± 0.0	12.5 ± 0.9	13.0 ± 0.1	1.7 ± 0.0	11.7 ± 0.1	0.6 ± 0.0	12.5 ± 1.0	**18.3 ± 0.9**
SUNRGBD	8.9 ± 0.3	8.2 ± 0.3	-	7.3 ± 0.2	6.2 ± 0.2	5.3 ± 0.3	8.7 ± 0.2	8.9 ± 0.3	9.3 ± 0.2	**9.9 ± 0.4**

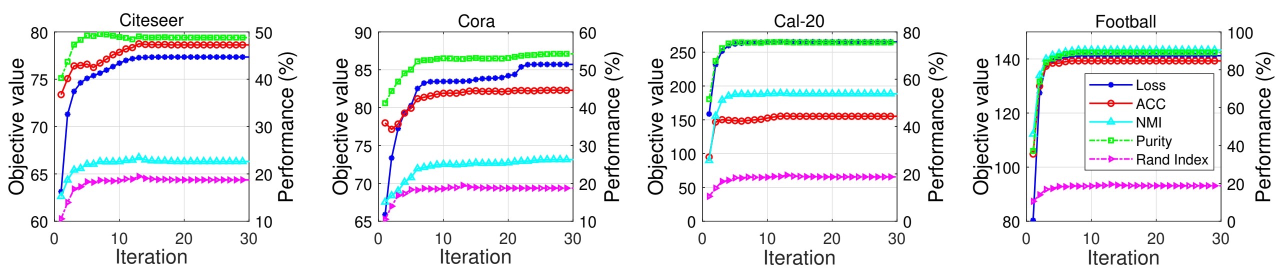

Figure 1. The curves of convergence and clustering performance of the proposed OP-LFMVC with the increase of iterations on benchmark datasets. The results on other datasets are similar and omitted due to space limit.

The implementations of the above algorithms are publicly available in corresponding papers, and we directly adopt them without modification in our experiments. Note that the issue of hyper-parameter tuning in clustering tasks is still an open problem. The proposed algorithm is free of hyper-parameter. However, among all compared algorithms, ONKC (Liu et al., 2017b), MKKM-MiR (Liu et al., 2016b), LKAM (Li et al., 2016) and LF-MVC (Wang et al., 2019) have hyper-parameters to be tuned. By following the same way in literature, we reuse their released codes and tune the hyper-parameters by grid search to produce the best possible results on each dataset. By this way, the reported results of these algorithms with hyper-parameters would be over-estimated. As a result, the hyper-parameter tuning would prohibit these multiple kernel (view) clustering algorithm from practical applications. It is therefore desired that a clustering algorithm is parameter-free, as the proposed OP-LFMVC does.

5.2. Experimental Results

Overall Clustering Performance Comparison Table 2 presents the ACC, NMI, Purity and RI comparison of the above algorithms. From this table, we have the following

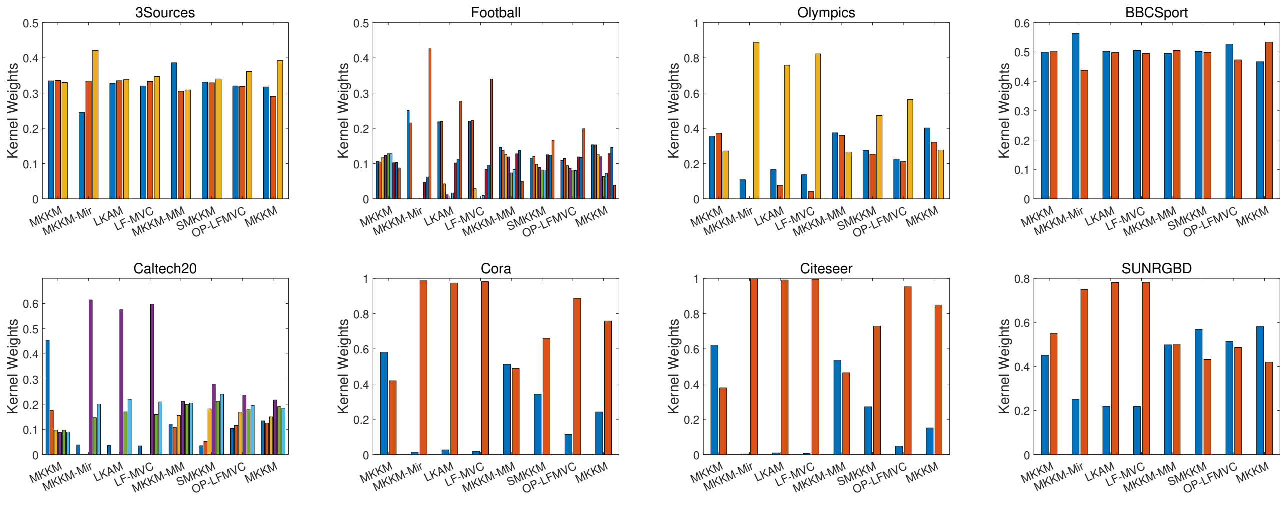

Figure 2. The kernel weights learned by different algorithms. OP-LFMVC maintains reduced sparsity compared to several competitors.

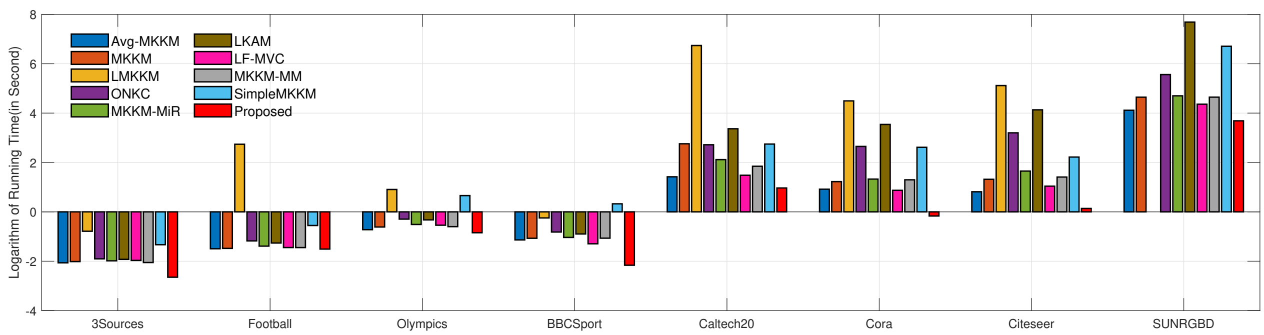

Figure 3. Run time comparison of different algorithms on eight benchmark datasets (in seconds). The experiments are conducted on a PC with Intel (R) Core (TM)-i9-10900X 3.7GHz CPU and 64G RAM in MATLAB environment.

observations:

- LF-MVC (Wang et al., 2019) demonstrates overall better clustering performance when compard with multiple kernel clustering algorithms on all benchmark datasets, indicating the advantage of late fusion over kernel based fusion. For example, LF-MVC exceeds SMKKM (Liu et al., 2020) by nearly 10 percents in terms of ACC on Football dataset. Note that SMKKM has been considered as the state-of-the-art among multiple kernel clustering algorithms. These results verify the effectiveness of late fusion paradigm in solving multi-view clustering.
- The proposed OP-LFMVC further improves LF-MVC and achieves the best clustering performance. For example, it exceeds the second best one by 5.3%, 8.4%, 9.3%, 3.5%, 22.9%, 9.9%, 7.8% and 2.8% in terms of ACC on all benchmark datasets. The improvements in terms of other criteria are similar. These results well demonstrate the superiority of jointly learning cluster labels.
- The variance of the proposed OP-LFMVC is zero. This is because the output of our algorithm is discrete, which avoids the randomness of k-means in generating clustering labels. This property demonstrates the robustness of OP-LFMVC.
- Our OP-LFMVC achieves better performance than MKKM-MiR (Liu et al., 2016b), ONKC (Liu et al., 2017b), and LF-MVC (Wang et al., 2019), all of which have several hyper-parameters to tune due to the incorporation of regularization on the kernel weights. These algorithms need to take a lot of effort to determine the best hyper-parameters in practical applications. And parameter tuning may be impossible in real applications where there is no ground truth clustering to optimize. In contrast, our OPLF-MVC is parameter-free.

In summary, OP-LFMVC demonstrates superior clustering performance over the alternatives on all datasets and has no hyper-parameter to tune. We expect that the simplicity and effectiveness of OP-LFMVC makes it a good option to be considered for practical clustering applications. Note that the results of LMKKM (Gönen & Margolin, 2014) on

Table 3. Clustering performance comparison between OP-LFMVC and TP-LFMVC on eight datasets in terms of ACC, NMI, Purity and Rand Index. The results of MKKM are also provided as a baseline.

Dataset	MKKM	TP-LFMVC	OP-LFMVC
	ACC		
3Sources	40.4 ± 2.3	50.5 ± 1.1	**60.8 ± 5.3**
Football	73.0 ± 2.8	80.8 ± 3.3	**82.2 ± 4.5**
Olympics	62.1 ± 3.1	68.1 ± 3.5	**74.1 ± 2.5**
BBCSport	39.4 ± 0.7	50.2 ± 4.7	**58.6 ± 4.5**
Caltech20	29.5 ± 1.2	39.2 ± 2.0	**45.4 ± 2.6**
Cora	25.3 ± 0.4	40.9 ± 0.1	**44.1 ± 3.0**
Citeseer	20.1 ± 0.0	40.6 ± 0.1	**47.1 ± 1.1**
SUNRGBD	17.4 ± 0.5	17.8 ± 0.4	**20.5 ± 0.7**
	NMI		
3Sources	30.9 ± 2.4	48.9 ± 2.5	**52.9 ± 4.3**
Football	78.9 ± 1.2	85.4 ± 2.0	**87.3 ± 2.6**
Olympics	72.5 ± 1.4	77.3 ± 1.6	**80.3 ± 1.3**
BBCSport	15.7 ± 0.5	31.6 ± 4.2	**43.3 ± 3.0**
Caltech20	37.9 ± 0.6	52.2 ± 0.8	**54.1 ± 1.0**
Cora	9.5 ± 0.2	**26.6 ± 0.1**	25.3 ± 2.1
Citeseer	1.9 ± 0.0	18.9 ± 0.1	**22.6 ± 0.9**
SUNRGBD	21.3 ± 0.3	**22.6 ± 0.2**	21.2 ± 0.4
	Purity		
3Sources	56.4 ± 2.8	72.0 ± 2.2	**72.4 ± 3.2**
Football	75.7 ± 1.9	83.6 ± 2.6	**84.5 ± 4.0**
Olympics	70.8 ± 2.1	76.8 ± 2.4	**79.8 ± 2.2**
BBCSport	48.9 ± 0.2	62.0 ± 2.9	**69.7 ± 2.8**
Caltech20	62.9 ± 0.7	74.0 ± 1.2	**76.0 ± 1.4**
Cora	36.1 ± 1.0	**51.9 ± 0.0**	51.3 ± 2.5
Citeseer	24.2 ± 0.0	44.1 ± 0.1	**49.0 ± 1.2**
SUNRGBD	36.2 ± 0.5	**38.0 ± 0.5**	36.6 ± 0.6
	Rand Index		
3Sources	20.0 ± 3.4	34.2 ± 1.5	**44.2 ± 6.0**
Football	60.3 ± 2.2	71.0 ± 3.5	**74.6 ± 4.8**
Olympics	47.0 ± 3.1	57.0 ± 3.3	**64.6 ± 2.8**
BBCSport	9.2 ± 0.3	21.5 ± 3.2	**32.6 ± 4.4**
Caltech20	18.6 ± 0.8	27.3 ± 1.2	**33.1 ± 2.0**
Cora	3.6 ± 0.3	17.3 ± 0.1	**18.6 ± 2.2**
Citeseer	0.3 ± 0.0	11.7 ± 0.1	**18.3 ± 0.9**
SUNRGBD	8.2 ± 0.3	8.7 ± 0.2	**9.9 ± 0.4**

Table 4. Running time comparison between OP-LFMVC and TP-LFMVC on eight datasets (in seconds).

Dataset	MKKM	TP-LFMVC	OP-LFMVC
3Sources	0.13	0.14	**0.07**
Football	0.23	0.24	**0.22**
Olympics	0.54	0.58	**0.43**
BBCSport	0.34	0.27	**0.12**
Caltech20	15.77	4.40	**2.63**
Cora	3.40	2.39	**0.85**
Citeseer	3.74	2.83	**1.15**
SUNRGBD	104.09	78.34	**39.96**

some datasets are not reported due to out-of-memory errors, which are caused by its cubic computational and memory complexity.

Ablation comparison In this section, we design an ablation study to clearly demonstrate the superiority of the proposed OP-LFMVC. To do so, we develop an additional algorithm, which optimizes (7) by alternate optimization to generate a consensus partition clustering matrix $\mathbf{H}$. $\mathbf{H}$ is then taken as the input of k-means to produce the cluster labels. We term this algorithm as two pass late fusion MVC (TP-LFMVC). As seen, the only difference between these algorithms is the manner of generating cluster labels.

We experimentally compare both algorithms on all benchmark datasets and report the results in Table 3. We can see that OP-LFMVC significantly improves TP-LFMVC in terms of all clustering criteria. Taking the results on Citeseer for example, OPLF-MVC gains 22.9%, 15.1%, 5.9 and 5.6% improvement in terms of ACC, NMI, purity and rand index compared to LF-MVC, verifying the effectiveness of sufficient negotiation between consensus partition matrix learning and cluster labels generation. We can get similar observations from the results on other datasets.

This ablation study clearly reveals the important difference between OP-LFMVC and TP-LFMVC: OP-LFMVC is a goal-directed, making the learned consensus matrix best serve for the generation of cluster labels. Meanwhile, we also record the running time of both algorithms in Table 4. As seen, the proposed OP-LFMVC demonstrates slightly better computational efficiency on all datasets.

Convergence and Evolution of the Learned $\mathbf{Y}$ As discussed in Section 3.3, OP-LFMVC is theoretically guaranteed to converge. To show this point in depth, we plot the objective of OP-LFMVC with iterations on all datasets, as shown in Figure 1. From these figures, we observe that its objective is monotonically increased and the algorithm usually converges in less than ten iterations on all datasets. Also, to show the clustering performance of OP-LFMVC with iterations, we take $\mathbf{Y}$ at each iteration to calculate ACC, NMI, purity and rand index, and report them in Figure 1. As observed, the clustering performance of OP-LFMVC is firstly increased with iterations, and then kept stable, which sufficiently demonstrates the effectiveness of our algorithm. These results considerably show the effectiveness and necessity of the learning procedure.

Running Time Comparison To evaluate the computational efficiency of the proposed algorithms, Fig. 3 reports the running time of the aforementioned algorithms on all benchmark datasets. Note that we take logarithm of the running time of all algorithms for better illustration. As can be seen, OP-LFMVC has much shorter running time on all datasets when compared to other multi-view algorithms, verifying its computational efficiency. In sum, both the theoretical and the experimental results have well demonstrated the computational advantage of the proposed algorithms, making them efficient to handle practical multi-view clustering tasks.

6. Conclusion

In this paper, we propose the OP-LFMVC algorithm which directly optimizes the cluster labels, instead of a consensus partition matrix. By this way, OP-LFMVC enhances the negotiation between the generation of clustering labels and the optimization of clustering. We show that the resultant objective can be amenable to a solution by the widely used alternate optimization with proved convergence. We derive a generalization bound for our approach using global Rademacher complexity analysis. Comprehensive experiments demonstrate the effectiveness and efficiency of OP-LFMVC. We expect that the simplicity, free of hyper-parameters, and effectiveness of OP-LFMVC makes it a go-to solution for practical multi-view clustering applications in the future. Future work may aim to extend OP-LFMVC to handle incomplete views, study further applications, and derive convergence rates using local Rademacher complexity analysis (Kloft & Blanchard, 2012; Cortes et al., 2013).

Acknowledgements

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61922088, 61906020 and 61773392).

References

Bang, S., Yu, Y., and Wu, W. Robust multiple kernel k-means clustering using min-max optimization, 2018.

Cortes, C., Kloft, M., and Mohri, M. Learning kernels using local rademacher complexity. In *Advances in Neural Information Processing Systems*, pp. 2760–2768, 2013.

Gönen, M. and Margolin, A. A. Localized data fusion for kernel k-means clustering with application to cancer biology. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 1305–1313, 2014.

Huang, H., Chuang, Y., and Chen, C. Multiple kernel fuzzy clustering. *IEEE Trans. Fuzzy Systems*, 20(1):120–134, 2012.

Huang, Z., Hu, P., Zhou, J. T., Lv, J., and Peng, X. Partially view-aligned clustering. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*.

Kloft, M. and Blanchard, G. On the convergence rate of lp-norm multiple kernel learning. *J. Mach. Learn. Res.*, 13:2465–2502, 2012.

Li, M., Liu, X., Wang, L., Dou, Y., Yin, J., and Zhu, E. Multiple kernel clustering with local kernel alignment maximization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pp. 1704–1710, 2016.

Liu, T., Tao, D., and Xu, D. Dimensionality-dependent generalization bounds for k-dimensional coding schemes. *Neural computation*, 28(10):2213–2249, 2016a.

Liu, W., Shen, X., and Tsang, I. W. Sparse embedded k-means clustering. In *Advances in Neural Information Processing Systems 30*, pp. 3319–3327, 2017a.

Liu, X., Dou, Y., Yin, J., Wang, L., and Zhu, E. Multiple kernel k-means clustering with matrix-induced regularization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pp. 1888–1894, 2016b.

Liu, X., Zhou, S., Wang, Y., Li, M., Dou, Y., Zhu, E., and Yin, J. Optimal neighborhood kernel clustering with multiple kernels. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 2266–2272, 2017b.

Liu, X., Zhu, E., and Liu, J. Simplemkkm: Simple multiple kernel k-means. *arXiv preprint arXiv:2005.04975*, 2020.

Maurer, A. and Pontil, M. k-dimensional coding schemes in Hilbert spaces. *IEEE Transactions on Information Theory*, 56(11):5839–5846, 2010.

Peng, X., Huang, Z., Lv, J., Zhu, H., and Zhou, J. T. COMIC: multi-view clustering without parameter selection. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, pp. 5092–5101.

Wang, H., Nie, F., and Huang, H. Multi-view clustering and feature learning via structured sparsity. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, volume 28, pp. 352–360.

Wang, S., Liu, X., Zhu, E., Tang, C., Liu, J., Hu, J., Xia, J., and Yin, J. Multi-view clustering via late fusion alignment maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 3778–3784, 2019.

Yu, S., Tranchevent, L.-C., Liu, X., Glänzel, W., Suykens, J. A. K., Moor, B. D., and Moreau, Y. Optimized data fusion for kernel k-means clustering. *IEEE TPAMI*, 34 (5):1031–1039, 2012.

Zhao, B., Kwok, J. T., and Zhang, C. Multiple kernel clustering. In *SDM*, pp. 638–649, 2009.