One-Stage Incomplete Multi-view Clustering via Late Fusion

Yi Zhang
zhangy@nudt.edu.cn
College of Computer, National University of Defense Technology
Changsha, Hunan, China

Xinwang Liu*
xinwangliu@nudt.edu.cn
College of Computer, National University of Defense Technology
Changsha, Hunan, China

Siwei Wang
wangsiwei13@nudt.edu.cn
College of Computer, National University of Defense Technology
Changsha, Hunan, China

Jiyuan Liu
liujiyuan13@nudt.edu.cn
College of Computer, National University of Defense Technology
Changsha, Hunan, China

Sisi Dai
daisisi@nudt.edu.cn
College of Computer, National University of Defense Technology
Changsha, Hunan, China

En Zhu*
enzhu@nudt.edu.cn
College of Computer, National University of Defense Technology
Changsha, Hunan, China

ABSTRACT

As a representative of multi-view clustering (MVC), late fusion MVC (LF-MVC) algorithm has attracted intensive attention due to its superior clustering accuracy and high computational efficiency. One common assumption adopted by existing LF-MVC algorithms is that all views of each sample are available. However, it is widely observed that there are incomplete views for partial samples in practice. In this paper, we propose One-Stage Late Fusion Incomplete Multi-view Clustering (OS-LF-IMVC) to address this issue. Specifically, we propose to unify the imputation of incomplete views and the clustering task into a single optimization procedure, so that the learning of the consensus partition matrix can directly assist the final clustering task. To optimize the resultant optimization problem, we develop a five-step alternate strategy with theoretically proved convergence. Comprehensive experiments on multiple benchmark datasets are conducted to demonstrate the efficiency and effectiveness of the proposed OS-LF-IMVC algorithm.

CCS CONCEPTS

• Theory of computation → Unsupervised learning and clustering; **• Computing methodologies → Cluster analysis**.

KEYWORDS

incomplete data, multi-view clustering, one stage, late fusion

ACM Reference Format:
Yi Zhang, Xinwang Liu*, Siwei Wang, Jiyuan Liu, Sisi Dai, and En Zhu*. 2021. One-Stage Incomplete Multi-view Clustering via Late Fusion. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475204

*CorrespondingAuthor.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475204

1 INTRODUCTION

Multi-view clustering (MVC) optimally combines a group of complementary features from different views to improve clustering performance [3]. According to the manners in integrating multiple views, existing MVC algorithms can be grouped into the following categories. The first category is based on matrix factorization technique. By seeking a common indicator matrix, the methods can effectively learn the latent cluster structure with low-dimensional representations embedded in multiple views [16, 18, 26, 29, 34, 36, 38]. The second one is based on subspace. This category learns a unified feature representation from all feature subspace of views by assuming all views share this representation [14, 23]. Most multi-view subspace clustering methods learn a sample affinity graph matrix for each view to build a consensus graph [5, 30, 35]. Some other approaches directly learn a common graph matrix [1]. The third one is based on multiple kernel learning. This category seeks to find a fused graph across different views and then uses graph-cut algorithms or other techniques on the obtained graph to produce clustering results. This kind of methods provide an elegant framework to group samples into clusters by extracting common structure from complementary information sources (base kernels) [6, 9, 10, 17, 28]. Besides, deep clustering and ensemble clustering are also concerned in multi-view community [4, 7, 8, 12, 19, 32]. The proposed algorithm belongs to the third category.

To reduce the computational and storage complexity of multiple kernel clustering, Wang et al. propose a late fusion method, known as LF-MVC, which maximally aligns the consensus partition with the weighted base partition, instead of fusing the view information in the early stage [27]. Despite demonstrating promising clustering performance, LF-MVC assumes that all pre-computed kernel matrices are completely observable. However, in some practical applications [15, 31], it is widely observed that certain types of features are not available, leading to the incompleteness of corresponding relevant kernel matrices. As a result, the violation to this assumption causes that existing multi-view clustering cannot effectively handle incomplete MVC.

A general solution to deal with this issue is to first estimate the incomplete kernels by imputation algorithms [2, 24, 25, 33], and then applies the MVC algorithm to the imputed kernels. However, the imputation and clustering procedure are performed independently and lack of necessary interaction, which would prevent these two learning procedures from negotiating with each other to achieve the best clustering performance.

Moreover, we also observe that the common practices of most MVC algorithms employ k-means on the learned consensus clustering partition matrix to generate the discrete cluster label. This means that the consensus partition matrix learning and cluster label generation are performed separately. These two processes lack of the necessary connection and cannot help each other to achieve the optimal goal, which adversely affects the clustering performance.

In this paper, we propose an one-stage late-fusion incomplete multi-view clustering algorithm (OS-LF-IMVC) to solve the above-mentioned issues. OS-LF-IMVC first clusters each independent incomplete similarity matrix to obtain the base clustering partition matrices, and then impute these incomplete base clustering partition matrices. They are taken to learn the consensus clustering partition matrix, which will in return to impute each incomplete basic clustering matrix. These two steps are performed alternately until convergence. Specifically, the obtained clustering results provide guidance for the imputation of incomplete base clustering matrices which is fed to the subsequent clustering. In this way, these two processes are seamlessly connected so that better clustering performance can be achieved. Meanwhile, we also decompose the consensus partition matrix to integrate the consensus clustering partition learning and cluster label generation into unified optimization goal. By this way, the consensus partition matrix learning can directly assist the final clustering task. Since the late fusion learning method is adopted and the learning process directly points to the generation of cluster labels, it is foreseeable that OS-LF-IMVC will have significant advantages compared with existing algorithms in terms of optimization and computational efficiency. We design a simple and efficient five-step alternate strategy to optimize the resultant optimization problem with theoretically proven convergence, and analyze its computational and storage complexity. After that, we conduct comprehensive experiments on eight benchmark datasets to study properties of the proposed OS-LF-IMVC, including clustering performance in various missing ratios, the learned consensus matrix evolution and objective value variation with iterations. Experimental results demonstrate that OS-LF-IMVC significantly outperforms the compared methods in terms of clustering performance, and execution time of the algorithm is greatly reduced.

We summarize the main contributions of this work as follows:

(1) We unify the imputation of incomplete base partition matrices, consensus cluster partition matrix learning and cluster label generation into one unified optimization goal. This allows the imputation of the base partition matrices to help the learning of the consensus partition matrix and the final clustering task. These procedures can negotiate with each other in a single optimization to achieve the best clustering results.
(2) We carefully design a simple and efficient five-step alternate optimization strategy. In addition, its computational and storage complexities are discussed and we theoretically prove its convergence.
(3) We discuss advantages of the proposed algorithm compared with other imputation methods in detail, and carry out comprehensive and extensive experiments to verify its effectiveness and efficiency in terms of computational efficiency and clustering performance.

2 RELATED WORK

2.1 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 [27]. Given n samples in k clusters among m views, its optimization goal can be mathematically expressed as

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

where the objective denotes the alignment between the consensus partition matrix $\mathbf{H} \in \mathbb{R}^{n\times k}$ and a group of pre-calculated base partition matrices $\{\mathbf{H}_p\}_{p=1}^m$, and $\mathbf{W}_p \in \mathbb{R}^{k\times k}$ is the p-th transformation matrix. A three-step optimization procedure with proved convergence is developed to solve the optimization in Eq. (1). According to the analysis in [27], the computational complexity of late fusion MVC is linear in the number of samples, which enables it to handle with large-scale clustering tasks.

2.2 Late Fusion Incomplete Multi-view Clustering

Late fusion Incomplete multi-view clustering [21] has recently been proposed to improve imputation quality by impute the incomplete base clustering segmentation matrices$\mathbf{H}_p^{(o)}$, which obtained by solving kernel k-means in MKKM. Specifically, it firstly finds a consensus partition matrix $\mathbf{H}$ from $\{\mathbf{H}_p\}_{p=1}^m$, and then fills the incomplete parts of them. Its optimization goal can be mathematically expressed as

$$\begin{aligned} \max_{\mathbf{H},\{\mathbf{H}_p,\mathbf{W}_p\}_{p=1}^m} \ & \mathrm{Tr}\left[\mathbf{H}^\top\left(\sum_{p=1}^m \mathbf{H}_p\mathbf{W}_p\right)\right] + \lambda \sum_{p=1}^m \mathrm{Tr}\left(\hat{\mathbf{H}}_p^\top \mathbf{H}_p^{(0)}\right), \\ s.t.\ & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \mathbf{W}_p^\top\mathbf{W}_p = \mathbf{I}_k, \mathbf{H}_p^\top\mathbf{H}_p = \mathbf{I}_k, \end{aligned} \tag{2}$$

where $\mathbf{H}$ and $\mathbf{H}_p$ denote the consensus clustering matrix and the p-th base clustering matrix, respectively. Nevertheless, $\mathbf{W}_p$ is the p-th permutation matrix in order to optimally match $\mathbf{H}_p$ and $\mathbf{H}$. $\mathbf{s}_p$ denotes the indexes of observable samples.

Although the recent LF-IMVC has some nice properties such as less imputation variables and higher computational efficiency compared with other algorithms, such as MKKM-IK [22], it also suffers from the following non-ignorable drawbacks. (i) It is vulnerable to low-quality imputation. As seen from Eq. (2), the observed part of each base clustering matrix $\{\mathbf{H}_p\}_{p=1}^m$ does not require unchanged during the learning course. Consequently, there are $n \times k$ elements to be optimized for each $\mathbf{H}_p$. This unnecessarily increases the complexity of the optimization and the risk of being trapped into a low-quality local minimum. In addition, the imputation on $\{\mathbf{H}_p\}_{p=1}^m$ would affect the clustering of all samples, even if they are complete. This improperly increases the impact of imputation on all samples, especially for those with complete views. (ii) It does not sufficiently consider different importance of different view and the correlation among kernels. This could result in that low-quality views are used to for clustering with the same weight and kernel selection would be mutually redundant and lack of the diversity of information sources. (iii) The learning procedures of learning the consensus partition matrix and generating cluster labels are performed separately. These two processes lack of the necessary connection and cannot help each other to achieve the most ideal final goal, which affects the clustering performance.

In the following part, we develop the one stage late fusion incomplete multi-view clustering algorithm (OS-LF-IMVC) to address the above issue.

2.3 The Proposed One Stage Late Fusion Incomplete Multi-view Clustering

2.4 Formulation

As seen from Eq. (1), the widely applied criterion in late fusion multi-view clustering [27] has shown its simplicity and effectiveness. And its variant [20, 21] tries to execute clustering with incomplete data. Though demonstrating relatively good clustering performance in some applications, we observe that the procedure of obtaining cluster labels is to perform k-means on consensus partition matrix $\mathbf{H}$. Therefore, it is obvious that the action of discretization is lacking of negotiation between consensus partition matrix learning and cluster label generation to achieve the ultimate goal. To address this problem and the incomplete clustering problem, we propose an One Pass Incomplete Late Fusion Multi-view Clustering which takes 'one-stage' method to unify imputation and clustering into a single optimization procedure and directly learns the discrete cluster label. In order to achieve this goal, we firstly decompose the consensus clustering partition matrix $\mathbf{H}$ as

$$\mathbf{H} = \mathbf{Y}\mathbf{C} \;\; s.t. \;\; \mathbf{Y}\mathbf{1} = \mathbf{1}, \tag{3}$$

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 the row sum of $\mathbf{Y}$ is 1 implies that each row of $\mathbf{Y}$ only has one element as 1 and others as 0.

Then, we defined the p-th$(1 \le p \le m)$ base clustering partition matrix as

$$\mathbf{H}_p = [\mathbf{H}_p^{(o)}\;;\mathbf{H}_p^{(u)}] \in \mathbb{R}^{n\times k}, \tag{4}$$

where $\mathbf{H}_p^{(o)} \in \mathbb{R}^{n_p\times k}$ is obtained by solving kernel k-means with m incomplete base kernel matrices $\{\mathbf{K}_p(\mathbf{o}_p,\mathbf{o}_p)\}_{p=1}^m$ corresponding to the observable entries, while $\mathbf{H}_p^{(u)} \in \mathbb{R}^{(n-n_p)\times k}$ denote the unobserved part of $\mathbf{H}_p$ that is required to be imputed.

According to the above discussion, OS-LF-IMVC proposes to simultaneously impute $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ and perform clustering while keeping $\{\mathbf{H}_p^{(o)}\}_{p=1}^m$ unchanged during the optimization procedure. Specially, the consensus cluster label matrix $\mathbf{Y}$ is optimized by imputed $\{\mathbf{H}_p\}_{p=1}^m$, and then impute $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ with the beneficial information of $\mathbf{Y}$. These learning procedures are integrated seamlessly. So that, our proposed algorithm can negotiate with each other to achieve optimal clustering as follows:

$$\begin{aligned}
&\max_{\mathbf{Y},\mathbf{C},\{\mathbf{W}_p,\mathbf{H}_p^{(u)},\beta_p\}_{p=1}^m} \mathrm{Tr}\left(\mathbf{C}^\top\mathbf{Y}^\top\sum_{p=1}^m\beta_p\begin{pmatrix}\mathbf{H}_p^{(o)}\\ \mathbf{H}_p^{(u)}\end{pmatrix}\mathbf{W}_p\right)+\lambda\mathrm{Tr}\left(\mathbf{C}^\top\mathbf{Y}^\top\mathbf{H}_0\right),\\
&s.t.\;\; \mathbf{C}^\top\mathbf{C}=\mathbf{I}_k,\; \mathbf{W}_p^\top\mathbf{W}_p=\mathbf{I}_k,\; \mathbf{H}_p^{(u)\top}\mathbf{H}_p^{(u)}=\mathbf{I}_k,\;\forall p,\\
&\mathbf{Y}\in\{0,1\}^{n\times k},\;\mathbf{Y}\mathbf{1}=\mathbf{1},\;\sum\nolimits_{p=1}^m\beta_p^2=1,\;\beta_p\ge 0,
\end{aligned} \tag{5}$$

where $\mathbf{H}_0$ is an initial estimate of $\mathbf{H}$, and λ is the regularization parameter. Note that an extra orthogonal constraint is imposed on $\mathbf{C}$ to bound the optimization. We also put orthogonal constraints on $\mathbf{H}_p^{(u)}$ and $\mathbf{W}_p$ since they are clustering matrix and permutation matrix, respectively.

2.5 Alternate Optimization

There are five variables in Eq. (5) to be optimized. Simultaneously optimizing them is not easy, we therefore design a five-step optimization procedure to alternately solve it. In each step, one variable is optimized with the others fixed.

2.5.1 Optimization $\mathbf{Y}$. Fixing $\boldsymbol{\beta}$, $\mathbf{C}$ and $\{\mathbf{W}_p\}_{p=1}^m$ and $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$, the optimization in Eq. (5) w.r.t $\mathbf{Y}$ is transformed to

$$\max_{\mathbf{Y}} \;\; \mathrm{Tr}\left(\mathbf{Y}\mathbf{B}^\top\right) \;\; s.t. \;\; \mathbf{Y}\in\{0,1\}^{n\times k}, \;\; \mathbf{Y}\mathbf{1}=\mathbf{1}, \tag{6}$$

where $\mathbf{B} = \sum_{p=1}^m \beta_p\mathbf{H}_p\mathbf{W}_p\mathbf{C}^\top + \lambda\mathbf{H}_0\mathbf{C}^\top$. Therefore, the optimal $\mathbf{Y}$ for Eq. (6) is

$$\mathbf{Y}(i,j) = 1, \tag{7}$$

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

2.5.2 Optimization $\mathbf{C}$. Fixing $\mathbf{Y}$, $\boldsymbol{\beta}$, $\{\mathbf{W}_p\}_{p=1}^m$ and $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$, the optimization in Eq. (5) w.r.t $\mathbf{C}$ is reduced to

$$\max_{\mathbf{C}} \;\; \mathrm{Tr}\left(\mathbf{C}^\top\mathbf{A}\right) \;\; s.t. \;\; \mathbf{C}^\top\mathbf{C}=\mathbf{I}_k, \tag{8}$$

where $\mathbf{A} = \mathbf{Y}^\top\sum_{p=1}^m\beta_p\mathbf{H}_p\mathbf{W}_p + \mathbf{Y}^\top\lambda\mathbf{H}_0$. Eq. (8) can be efficiently solved by SVD with computational complexity $O(nk^2)$.

2.5.3 Optimization $\{\mathbf{W}_p\}_{p=1}^m$. Fixing $\boldsymbol{\beta}, \mathbf{C}\,\mathbf{Y}$ and $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$, the optimization in Eq. (5) w.r.t each $\mathbf{W}_p$ can be rewritten as,

$$\max_{\mathbf{W}_p} \;\; \mathrm{Tr}(\mathbf{W}_p^\top\mathbf{H}_p^\top\mathbf{Y}\mathbf{C}) \;\; s.t. \;\; \mathbf{W}_p^\top\mathbf{W}_p=\mathbf{I}_k. \tag{9}$$

Similar to Eq. (8), it can be efficiently solved via SVD with computational complexity $O(nk^2)$.

2.5.4 Optimization $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$. Fixing $\boldsymbol{\beta}, \mathbf{C}, \mathbf{Y}$ and $\{\mathbf{W}_p\}_{p=1}^m$, the optimization in Eq. (5) w.r.t each $\mathbf{H}_p^{(u)}$ is equivalent to,

$$\max_{\mathbf{H}_p^{(u)}} \;\; \mathrm{Tr}(\mathbf{H}_p^{(u)\top}\mathbf{Y}^{(u)}\mathbf{C}\mathbf{W}_p^\top) \;\; s.t. \;\; \mathbf{H}_p^{(u)\top}\mathbf{H}_p^{(u)}=\mathbf{I}_k. \tag{10}$$

It can be efficiently solved via SVD with computational complexity $O(nk^2)$.

Algorithm 1 One Stage Late Fusion Incomplete Multi-view Clustering (OS-LF-IMVC)

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. (6);
  update $\mathbf{C}$ by optimizing Eq. (8);
  update $\{\mathbf{W}_p\}_{p=1}^m$ by optimizing Eq. (9);
  update $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ by optimizing Eq. (10);
  update $\boldsymbol{\beta}$ by optimizing Eq. (11);
  if $(\mathrm{obj}^{(t)}-\mathrm{obj}^{(t-1)})/\mathrm{obj}^{(t)}\le 10^{-3}$ **then**
    flag=0;
  end if
  $t\leftarrow t+1$;
end while

2.5.5 Optimization $\boldsymbol{\beta}$. Fixing $\mathbf{Y}$, $\mathbf{C}$, $\{\mathbf{W}_p\}_{p=1}^m$ and $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$, the optimization in Eq. (5) w.r.t $\boldsymbol{\beta}$ is equivalently rewritten as

$$\max_{\boldsymbol{\beta}}\ \sum\nolimits_{p=1}^m \beta_p\alpha_p\ \ s.t.\ \ \sum\nolimits_{p=1}^m \beta_p^2=1,\ \beta_p\ge 0, \tag{11}$$

where $\alpha_p=\mathrm{Tr}\left(\mathbf{C}^\top\mathbf{Y}^\top\mathbf{H}_p\mathbf{W}_p\right)$. Therefore, the optimal solution for Eq. (11) is,

$$\beta_p=\alpha_p/\sqrt{\sum\nolimits_{q=1}^m \alpha_q^2}. \tag{12}$$

The whole optimization procedure in solving Eq. (5) is outlined in Algorithm 1, where $\mathrm{obj}^{(t)}$ indicates the objective value at the t-th iteration.

2.6 Discussion

2.6.1 Convergence. Note that the objective value in Eq. (5) monotonically increases when one variable is optimized with the others fixed. Moreover, our objective is upper-bounded. As a result, the optimization procedure in solving Eq. (5) is theoretically guaranteed to be (locally) convergent, as validated by our experimental results in Figure 5.

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

2.6.3 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 other multi-view clustering and incomplete clustering. Moreover, some prior knowledge and local information could incorporated into the formulation of OS-LF-IMVC to further improve the clustering performance. Our work provides a more effective paradigm to fuse incomplete multi-view data for clustering, which could inspire novel research on multi-view clustering.

3 EXPERIMENTAL RESULTS

In this section, the comprehensive experimental study is conducted to evaluate the proposed OS-LF-IMVC in terms of overall clustering performance comparison, convergence analysis, the evolution of the learned cluster label with iterations and running time comparison. Furthermore, we carefully design an ablation experiment to clearly demonstrate the efficiency and effectiveness of directly learning the cluster labels.

3.1 Experimental Settings

The experimental comparison is experimentally evaluated on a number of publicly available multi-view benchmark datasets, including

AR10P[1] 130/6/10, *3Sources*[2] 169/3/6, *Sonar*[3] 207/10/2, *Heart*[4] 270/13/2, *Wdbc*[5] 569/10/2, *Plant*[6] 940/69/4, *Citeseer*[7] 3312/2/6, *SUNRGBD*[8] 10335/2/45. The above three numbers represent the numbers of samples, views and clusters. Taking SUNRGBD dataset as an example, it has 10335 samples, 2 data views and 45 clusters. As observed, the numbers of samples, views and clusters vary within fairly large intervals, which supply a good platform to compare 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. We follow the approach in [20–22, 37] to generate the missing vectors$\{\mathbf{s}_p\}_{p=1}^m$. The parameter ϵ refers to the missing ratio in experiment , and it affect the performance of all the comparison algorithms. In order to show the influence by ϵ in depth, we compare all algorithms with respect to ϵ, which is set as $[0.05:0.05:0.5]$.

We use four widely used criteria to evaluate the clustering performance of all algorithms: clustering accuracy (ACC), normalized mutual information (NMI), purity and rand index (RI). For all compared algorithms, to reduce the effect of randomness by k-means, each experiment is repeated for 20 times and report the average results and the corresponding standard deviations. Best results are marked in bold.

We compare OS-LF-IMVC with several widely used imputation methods, including mean zero filling (ZF), filling (MF), k-nearest-neighbor filling (KNN) and alignment-maximization filling (AF) [25]. The commonly used MKKM [13] is performed on these imputed base kernels. These two-step methods are termed MKKM+ZF, MKKM+MF, MKKM+KNN and MKKM+AF,respectively. Also, some recently proposed MKKM-IK [22], MIC [3] and DAIMC [11] are also incorporated into comparison. In addition, the EE-R-IMVC [20] is compared, which improves and consistently exceeds LF-IMVC [21] and EE-IMVC and almost achieved state-of-the-art performance in handling incomplete multiple kernel clustering.

The implementations of the compared algorithms are publicly available in corresponding papers, and we directly adopt them without modification in our experiments. Note that the issue of

[1]http://featureselection.asu.edu/
[2]http://mlg.ucd.ie/datasets/3sources.html
[3]http://archive.ics.uci.edu/ml/datasets/
[4]http://archive.ics.uci.edu/ml/datasets/
[5]http://archive.ics.uci.edu/ml/datasets/
[6]https://bmi.inf.ethz.ch/supplements/protsubloc/
[7]http://linqs-data.soe.ucsc.edu/public/lbc/
[8]http://rgbd.cs.princeton.edu/

Table 1: Aggregated clustering accuracy (ACC), normalized mutual information (NMI), Purity and rand index (RI) comparison (mean±std) of different clustering algorithms on all benchmark datasets. Best results are marked in bold.

Dataset	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF	MKKM-IK	LI-MKKM	MIC	DAIMC	EE-R-IMVC	OS-LF-IMVC
					ACC					
AR10P	40.7 ± 4.4	40.7 ± 4.3	37.6 ± 4.2	40.5 ± 4.1	41.6 ± 4.6	35.0 ± 3.4	36.1 ± 2.3	38.6 ± 3.2	40.3 ± 4.1	**44.7 ± 3.0**
3Sources	37.5 ± 2.0	37.5 ± 2.1	37.6 ± 2.0	37.6 ± 2.0	39.4 ± 2.1	34.8 ± 1.5	49.4 ± 3.9	51.0 ± 2.4	51.1 ± 2.7	**55.7 ± 3.6**
Sonar	57.6 ± 0.4	57.3 ± 0.5	57.6 ± 0.5	57.5 ± 0.4	57.5 ± 0.3	56.9 ± 0.1	53.8 ± 0.2	55.3 ± 0.0	56.6 ± 0.8	**59.7 ± 0.2**
Heart	56.7 ± 0.2	55.2 ± 0.1	59.8 ± 1.2	65.8 ± 0.8	64.9 ± 0.1	73.0 ± 0.1	55.8 ± 0.4	75.9 ± 0.0	81.4 ± 0.0	**82.3 ± 0.1**
Wdbc	91.0 ± 0.0	90.9 ± 0.0	91.0 ± 0.0	91.0 ± 0.0	91.1 ± 0.0	81.1 ± 0.0	63.0 ± 0.7	70.9 ± 0.0	90.9 ± 0.0	**94.1 ± 0.0**
Plant	56.7 ± 0.4	56.7 ± 0.4	57.7 ± 0.5	56.9 ± 0.5	56.4 ± 0.4	46.2 ± 0.5	39.2 ± 0.1	43.2 ± 0.2	56.0 ± 0.9	**59.6 ± 2.0**
Citeseer	19.9 ± 0.2	19.3 ± 0.2	20.0 ± 0.2	20.0 ± 0.2	20.5 ± 0.2	30.7 ± 0.5	31.5 ± 1.9	30.9 ± 0.4	37.1 ± 0.2	**37.5 ± 1.7**
SUNRGBD	16.1 ± 0.4	16.1 ± 0.4	16.1 ± 0.4	16.8 ± 0.5	16.8 ± 0.5	17.3 ± 0.4	11.4 ± 0.4	16.1 ± 0.5	17.4 ± 0.5	**17.9 ± 0.3**
					NMI					
AR10P	40.1 ± 4.3	40.0 ± 4.3	36.4 ± 4.1	39.7 ± 4.0	41.1 ± 4.5	31.5 ± 3.1	33.3 ± 2.2	38.0 ± 2.9	39.2 ± 3.9	**42.1 ± 2.9**
3Sources	23.8 ± 2.3	23.5 ± 2.3	23.6 ± 2.1	24.0 ± 2.2	27.3 ± 2.3	17.8 ± 1.6	36.3 ± 3.2	46.3 ± 2.0	**48.8 ± 2.3**	41.6 ± 3.2
Sonar	1.7 ± 0.2	1.6 ± 0.2	1.7 ± 0.2	1.7 ± 0.2	1.7 ± 0.1	1.2 ± 0.0	0.6 ± 0.1	1.0 ± 0.0	1.4 ± 0.3	**2.8 ± 0.1**
Heart	2.2 ± 0.1	1.3 ± 0.0	7.1 ± 1.1	9.9 ± 0.7	8.5 ± 0.1	17.2 ± 0.1	0.5 ± 0.2	20.9 ± 0.0	30.5 ± 0.0	**32.5 ± 0.2**
Wdbc	54.8 ± 0.0	54.7 ± 0.0	54.9 ± 0.0	55.0 ± 0.0	55.3 ± 0.0	36.3 ± 0.0	0.7 ± 0.8	12.5 ± 0.0	54.9 ± 0.0	**67.6 ± 0.0**
Plant	20.3 ± 0.4	20.3 ± 0.4	21.4 ± 0.5	20.6 ± 0.5	19.9 ± 0.3	12.8 ± 0.3	0.5 ± 0.0	11.1 ± 0.1	20.1 ± 0.9	**24.7 ± 2.0**
Citeseer	1.3 ± 0.2	0.6 ± 0.1	1.3 ± 0.2	1.4 ± 0.2	1.8 ± 0.3	8.0 ± 0.4	8.6 ± 1.5	9.1 ± 0.3	**15.6 ± 0.1**	14.1 ± 1.2
SUNRGBD	18.6 ± 0.3	18.6 ± 0.3	18.8 ± 0.3	19.2 ± 0.3	19.6 ± 0.3	**20.9 ± 0.2**	11.3 ± 0.3	19.8 ± 0.2	20.3 ± 0.3	17.5 ± 0.2
					Purity					
AR10P	41.3 ± 4.3	41.2 ± 4.2	38.1 ± 4.1	41.0 ± 4.1	42.0 ± 4.5	35.6 ± 3.4	37.3 ± 2.2	39.3 ± 3.1	40.8 ± 4.0	**45.5 ± 2.9**
3Sources	53.0 ± 2.1	52.9 ± 2.2	52.9 ± 2.0	53.0 ± 2.0	55.2 ± 2.0	49.3 ± 1.4	61.4 ± 2.8	69.8 ± 1.5	**71.0 ± 1.7**	66.5 ± 2.4
Sonar	57.6 ± 0.4	57.3 ± 0.5	57.6 ± 0.5	57.5 ± 0.4	57.5 ± 0.3	56.9 ± 0.1	53.9 ± 0.2	55.5 ± 0.0	56.6 ± 0.8	**59.7 ± 0.2**
Heart	58.0 ± 0.1	57.0 ± 0.0	61.7 ± 1.0	66.2 ± 0.7	65.4 ± 0.1	73.0 ± 0.1	55.9 ± 0.1	75.9 ± 0.0	81.4 ± 0.0	**82.3 ± 0.1**
Wdbc	91.0 ± 0.0	90.9 ± 0.0	91.0 ± 0.0	91.0 ± 0.0	91.1 ± 0.0	81.1 ± 0.0	63.2 ± 0.4	70.9 ± 0.0	90.9 ± 0.0	**94.1 ± 0.0**
Plant	56.7 ± 0.4	56.7 ± 0.4	57.7 ± 0.5	56.9 ± 0.5	56.4 ± 0.4	54.1 ± 0.3	39.5 ± 0.0	46.8 ± 0.1	56.9 ± 0.8	**59.8 ± 1.7**
Citeseer	23.4 ± 0.2	22.6 ± 0.2	23.3 ± 0.2	23.4 ± 0.3	24.1 ± 0.3	33.7 ± 0.6	33.3 ± 1.9	34.9 ± 0.4	40.4 ± 0.2	**40.5 ± 1.6**
SUNRGBD	33.3 ± 0.5	33.3 ± 0.5	33.5 ± 0.6	34.1 ± 0.5	34.8 ± 0.5	35.6 ± 0.4	24.6 ± 0.5	35.0 ± 0.4	**35.8 ± 0.5**	32.5 ± 0.4
					RI					
AR10P	17.7 ± 4.1	17.7 ± 4.1	14.5 ± 3.6	17.4 ± 3.8	18.7 ± 4.3	11.1 ± 2.5	12.8 ± 1.9	15.6 ± 2.6	17.2 ± 3.7	**20.7 ± 2.9**
3Sources	14.5 ± 2.2	14.4 ± 2.3	14.5 ± 2.1	14.7 ± 2.2	17.6 ± 2.3	10.3 ± 1.3	28.3 ± 4.5	35.8 ± 2.6	35.7 ± 3.3	**35.8 ± 4.0**
Sonar	1.9 ± 0.2	1.7 ± 0.2	1.9 ± 0.3	1.9 ± 0.3	1.8 ± 0.2	1.5 ± 0.0	0.2 ± 0.1	0.9 ± 0.0	1.4 ± 0.5	**3.4 ± 0.2**
Heart	2.2 ± 0.1	1.3 ± 0.1	7.6 ± 1.5	12.1 ± 0.9	10.5 ± 0.2	21.3 ± 0.1	0.2 ± 0.1	26.9 ± 0.0	39.2 ± 0.1	**41.7 ± 0.3**
Wdbc	67.0 ± 0.0	66.9 ± 0.0	67.0 ± 0.0	67.1 ± 0.0	67.5 ± 0.0	38.6 ± 0.0	0.8 ± 0.9	17.6 ± 0.0	66.9 ± 0.0	**77.6 ± 0.0**
Plant	18.2 ± 0.5	18.2 ± 0.4	19.4 ± 0.6	18.5 ± 0.6	17.8 ± 0.4	9.9 ± 0.3	0.2 ± 0.0	8.4 ± 0.1	18.8 ± 1.0	**22.7 ± 2.3**
Citeseer	0.2 ± 0.0	0.1 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	0.4 ± 0.1	6.1 ± 0.3	6.3 ± 1.1	6.7 ± 0.2	**9.9 ± 0.1**	9.3 ± 1.1
SUNRGBD	6.8 ± 0.2	6.8 ± 0.2	6.8 ± 0.2	7.2 ± 0.2	7.6 ± 0.3	7.8 ± 0.2	3.2 ± 0.2	7.3 ± 0.2	**7.9 ± 0.2**	7.4 ± 0.2

hyper-parameter tuning in clustering tasks is still an open problem, and LI-MKKM [37] and EE-R-IMVC [20] 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.

3.2 Experimental Results

3.2.1 Overall Clustering Performance Comparison. Table 1 presents the ACC, NMI, Purity and RI comparison of the above algorithms. From this table, we have the following observations:

- EE-R-IMVC [20] is recently proposed and demonstrates overall better performance when compared with other algorithms in handling incomplete multiple kernel clustering, indicating the advantage of late fusion over kernel based fusion. For example, EE-R-IMVC exceeds the second best result by at least 10 percents in terms of any criteria on 3Sources dataset and always maintain a leading position, which verifies the effectiveness of late fusion paradigm in solving multi-view clustering.
- Proposed OS-LF-IMVC achieves significantly better performance than MKKM+ZF, MKKM+MF, MKKM+KNN and MKKM +AF. For instance, it exceeds the best of them by 4.0%, 18.1%, 2.3%, 16.5%, 3.1%, 1.9%, 17.5% and 1.1% in terms of ACC on all benchmark datasets. The gaps in terms of other criteria are similar. These results well demonstrate the effectiveness of unifying the imputation of incomplete views and the clustering task into a single optimization procedure.
- The proposed OS-LF-IMVC further improves EE-R-IMVC and achieves the state-of-the-art clustering performance. For example, it exceeds EE-R-IMVC by 4.4%, 4.6%, 3.2%, 0.9%, 3.2%, 3.6%, 0.4% and 0.5% in terms of ACC on all benchmark

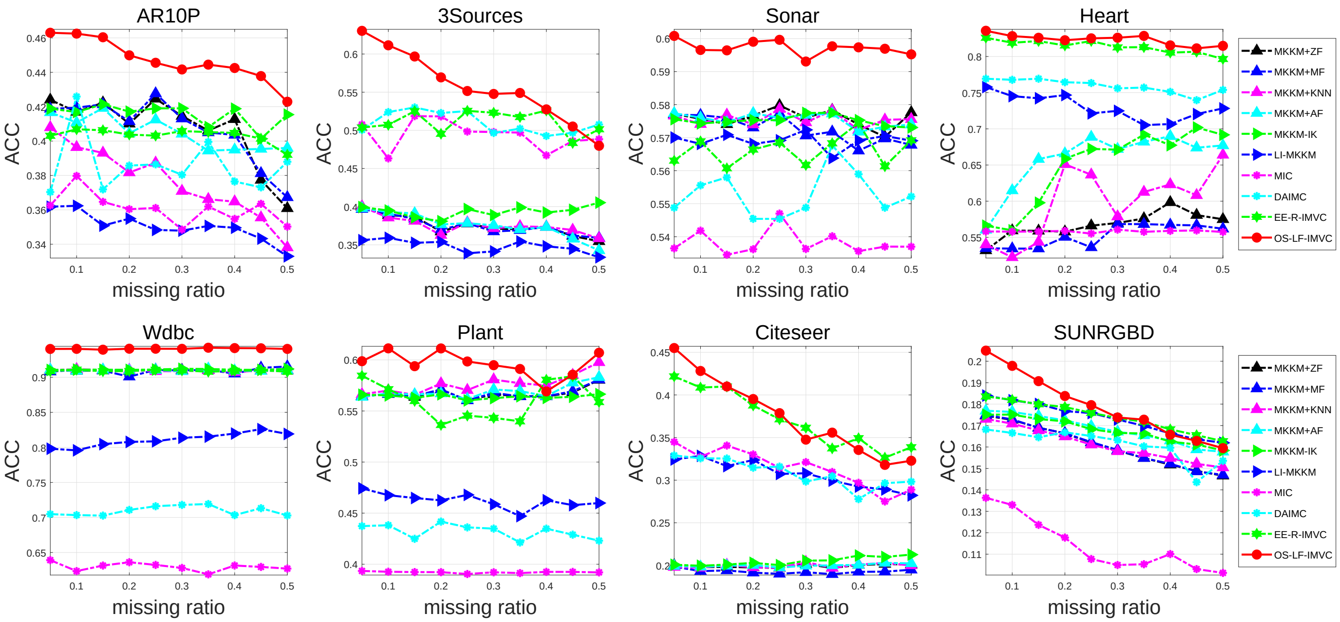

Figure 1: ACC comparison with the variation of missing ratios on eight benchmark datasets. For each given missing ratio, the 'incomplete patterns' are randomly generated for 10 times and their averaged results are reported.

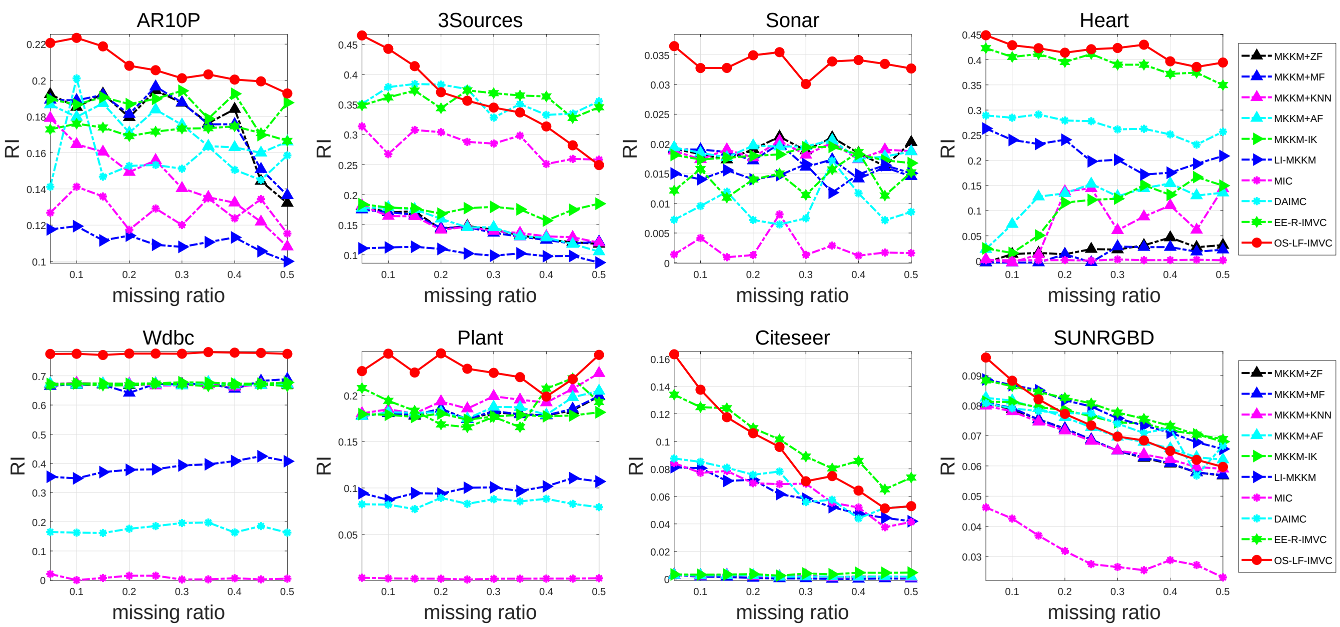

Figure 2: RI comparison with the variation of missing ratios on eight benchmark datasets. For each given missing ratio, the 'incomplete patterns' are randomly generated for 10 times and their averaged results are reported.

datasets. The improvements in terms of other criteria are similar. These results well demonstrate he superiority of jointly learning cluster labels.

In summary, OS-LF-IMVC demonstrates superior clustering performance over the alternatives on all datasets. We believe that it will be a better choice when dealing with incomplete clustering due to its high efficiency and superior performance.

3.2.2 Ablation comparison. In this paragraph, we design an ablation study to clearly demonstrate the superiority of the proposed OS-LF-IMVC. To do so, we specially compared proposed algorithm with LF-IMVC and EE-R-IMVC, which first generate a consensus partition clustering matrix $\mathbf{H}$ and $\mathbf{H}$ is then taken as the input of k-means to produce the cluster labels. We term this algorithm as two stage algorithms.

We experimentally compare these algorithms on all benchmark datasets and report the clustering performance and running time in Table 2 and Table 3. The result of MKKM+ZF is also provided

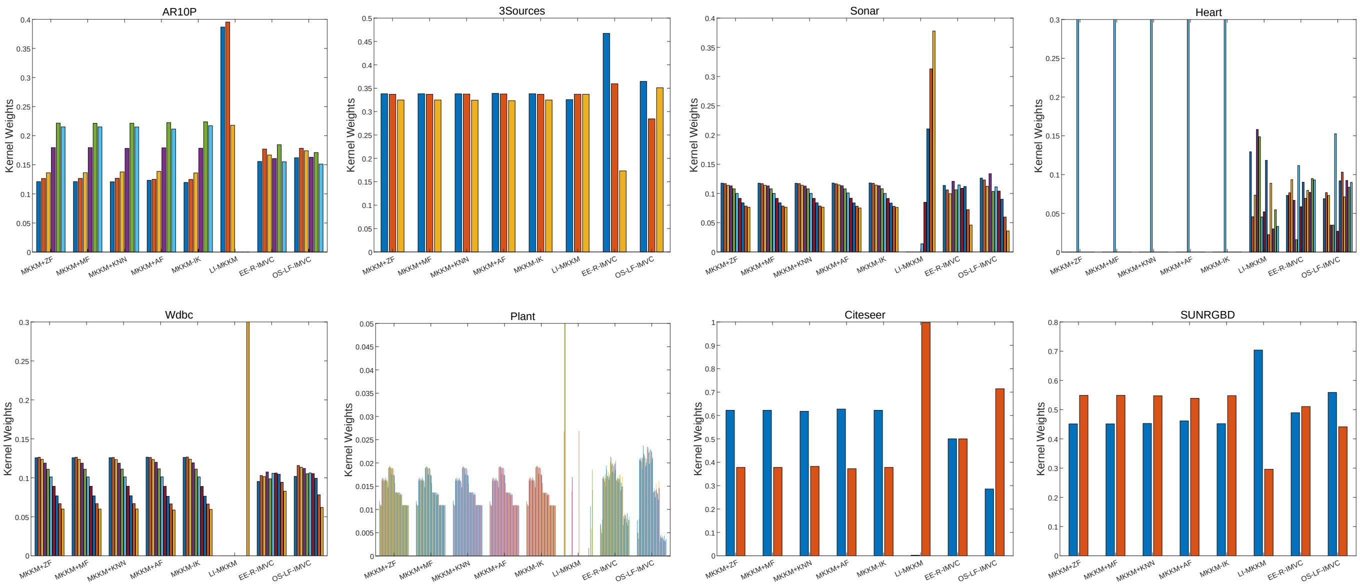

Figure 3: The kernel weights learned by different algorithms. OS-LF-IMVC maintains reduced sparsity compared to several competitors.

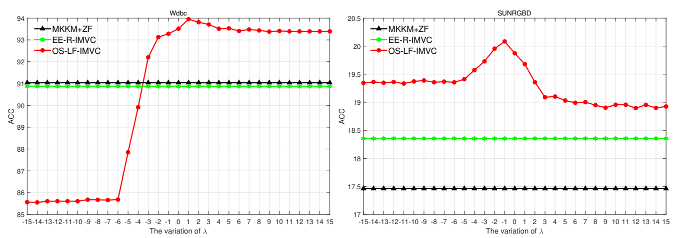

Figure 4: The sensitivity of OS-LF-IMVC with the variation of λ in term of ACC with missing ratio 0.05 on Wdbc and SUNRGBD. The results of MKKM+ZF and EE-R-IMVC are also provided as a baseline reference. The results in terms of NMI, Purity and RI with other missing ratios and ones on other datasets are similar and omitted due to space limit.

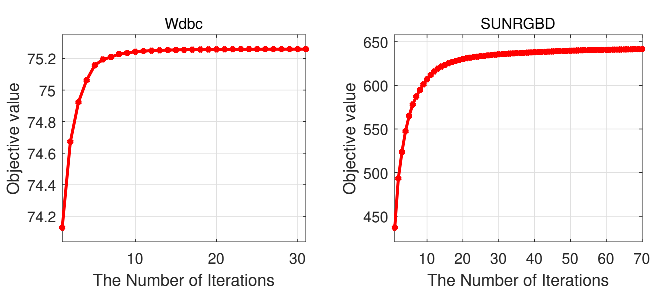

Figure 5: The objective values of OS-LF-IMVC with missing ratio 0.05 on Wdbc and SUNRGBD. The curves with other missing ratios and on other datasets are similar and we omit them due to space limit.

as a baseline. We can see that OS-LF-IMVC significantly improves LF-IMVC and EE-R-IMVC on all benchmark datasets. For example, OS-LF-IMVC gains 4.1%, 2.6%, 2.5%, 3.3%, 3.1%, 4.0%, 0.2%

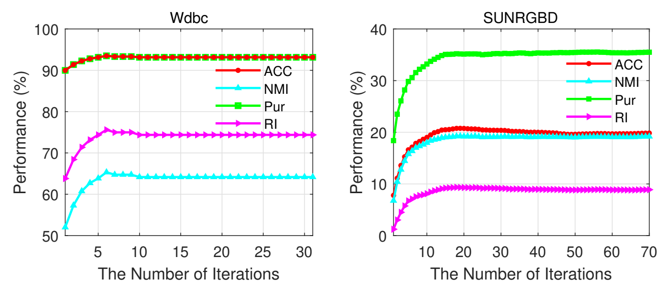

Figure 6: The evolution of the learned cluster labels Y by OS-LF-IMVC with missing ratio 0.05 on Wdbc and SUNRGBD. The curves with other missing ratios and on other datasets are similar and we omit them due to space limit.

and 2.2% improvement in term of ACC compared to LF-IMVC, verifying the effectiveness of sufficient negotiation between the learning procedures of consensus partition matrix and generating cluster labels. We can get similar observations when compared with baseline algorithm. In addition, proposed OS-LF-IMVC take the least time to complete the clustering task, demonstrating its computational efficiency. This ablation study clearly reveals the important improvement of OS-LF-IMVC over two stage algorithms: OS-LF-IMVC is a goal-directed and enables the learning procedure of consensus matrix to best serve for the final clustering task.

3.2.3 Convergence and Evolution of the Learned $\mathbf{Y}$. As discussed in Section 2.6, OS-LF-IMVC is theoretically guaranteed to converge. To show this point in depth, we plot the objective curves of OS-LF-IMVC with iterations on Wdbc and SUNRGBD datasets,with missing ratio 0.05, as shown in Figure 5. The objective curves with other missing ratios and ones on other datasets are similar and we omit them due to space limit. From these figures, we observe that its objective monotonically increases and the algorithm usually converges in a few iterations on all datasets. Also, to show the

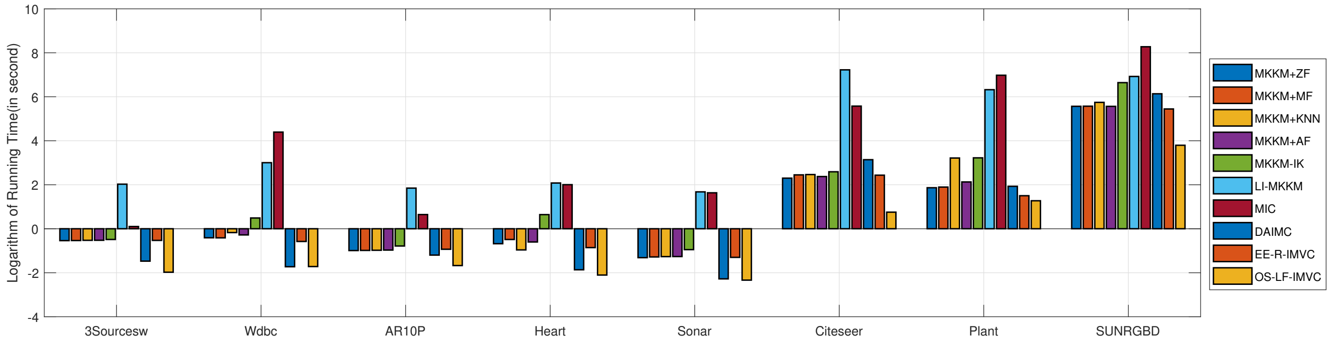

Figure 7: Running time comparison of different algorithms on eight benchmark datasets (in logarithm of second). The experiments are conducted on a PC with Intel (R) Core (TM)-i9-10900X 3.7GHz CPU and 64G RAM in MATLAB R2020b environment.

clustering performance of OS-LF-IMVC with iterations, we take **Y** at each iteration to calculate ACC, NMI, purity and RI, and report them in Figure 6. As observed, the clustering performance of OS-LF-IMVC is firstly increased with iterations, and then kept stable, which sufficiently verifies the effectiveness of our algorithm.

3.2.4 Parameter Sensitivity Analysis. As can be seen in Eq. (5), OS-LF-IMVC introduces the regularization parameter λ to trade off the clustering and imputation. We conduct experiments to show the effect of this parameter on the clustering performance on all datasets. Figure 4 presents the ACC of OS-LF-IMVC on Wdbc and SUNRGBD datasets with missing ratio 0.05 by varying λ from 2^{-15} to 2^{15}, where the MKKM+ZF and EE-R-IMVC are also provided as reference. From these figures, we observe that the ACC first increases to a high value and generally maintains it up to slight variation for a while and then drops with the increasing value of λ. LF-IMVC demonstrates stable performance across a wide range of λ. The curves with other missing ratios and on other datasets are similar, therefore, are omitted due to space limit. Comparing to MKKM+ZF and EE-R-IMVC, OS-LF-IMVC demonstrates stable performance across a wide range of λ.

Table 2: Clustering performance comparison among OS-LF-IMVC, LF-IMVC and EE-R-IMVC on eight datasets in terms of ACC. The results in terms of NMI, Purity and RI are similar and omitted due to space limit.

Dataset	MKKM+ZF	LF-IMVC	EE-R-IMVC	OS-LF-IMVC
AR10P	40.7 ± 4.4	40.2 ± 3.9	40.3 ± 4.1	**44.7 ± 3.0**
3Sources	37.5 ± 2.0	53.1 ± 3.1	51.1 ± 2.7	**55.7 ± 3.6**
Sonar	57.6 ± 0.4	57.2 ± 0.5	56.6 ± 0.8	**59.7 ± 0.2**
Heart	56.7 ± 0.2	79.0 ± 0.2	81.4 ± 0.0	**82.3 ± 0.1**
Wdbc	91.0 ± 0.0	91.0 ± 0.0	90.9 ± 0.0	**94.1 ± 0.0**
Plant	56.7 ± 0.4	55.6 ± 0.8	56.0 ± 0.9	**59.6 ± 2.0**
Citeseer	19.9 ± 0.2	37.3 ± 0.2	37.1 ± 0.2	**37.5 ± 1.7**
SUNRGBD	16.1 ± 0.4	15.7 ± 0.4	17.4 ± 0.5	**17.9 ± 0.3**

3.2.5 Running Time Comparison. To evaluate the computational efficiency of the proposed OS-LF-IMVC, Fig. 7 reports the running time of the all compared algorithms on all benchmark datasets. Note that in order to better illustration, we take logarithm of the running time of all algorithms. As seen, OS-LF-IMVC always has shortest running time on all datasets, verifying its computational efficiency.

Table 3: Running time comparison among OS-LF-IMVC, LF-IMVC and EE-R-IMVC on eight datasets (in second).

Dataset	MKKM+ZF	LF-IMVC	EE-R-IMVC	OS-LF-IMVC
AR10P	0.37	1.87	0.39	**0.19**
3Sources	0.58	2.84	0.59	**0.14**
Sonar	0.27	1.44	0.27	**0.10**
Heart	0.51	2.08	0.42	**0.12**
Wdbc	0.66	2.67	0.56	**0.18**
Plant	6.46	16.47	4.49	**3.57**
Citeseer	9.98	58.83	11.43	**2.13**
SUNRGBD	261.93	814.80	231.85	**44.55**

In summary, both the theoretical and the experimental results well demonstrate the advantage of the proposed OS-LF-IMVC in term of computational efficiency, which makes OS-LF-IMVC more efficient to deal with large-scale practical clustering tasks with incomplete data.

4 CONCLUSION

In this paper, we propose the OS-LF-IMVC algorithm which simultaneously performs clustering and impute the incomplete base clustering matrices, and directly optimizes the cluster labels, instead of a consensus partition matrix. By this way, OS-LF-IMVC not only improves mutual guidance between the clustering task and imputation procedure, but also enhances the negotiation between the generation of cluster labels and the optimization of clustering. We develop a five-step alternate algorithm with proved convergence to solve the resultant optimization problem. In addition, extensive experiments on benchmark datasets well demonstrate the superiority of our algorithms. In the future, we plan to explore a broader application range of the one-stage frame and extend it to different clustering methods.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China 2020AAA0107100, the Natural Science Foundation of China (project no. 61922088, 61773392, 61872377 and 61976196).

REFERENCES

[1] Mahdi Abavisani and Vishal M Patel. 2018. Multimodal sparse and low-rank subspace clustering. *Information Fusion* 39 (2018), 168–177.

[2] Sahely Bhadra, Samuel Kaski, and Juho Rousu. 2017. Multi-view kernel completion. *Machine Learning* 106, 5 (2017), 713–739.

[3] Steffen Bickel and Tobias Scheffer. 2004. Multi-view clustering.. In *ICDM*, Vol. 4. Citeseer, 19–26.

[4] Lucas Franek and Xiaoyi Jiang. 2014. Ensemble clustering by means of clustering embedding in vector spaces. *Pattern Recognition* 47, 2 (2014), 833–842.

[5] Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. 2015. Multi-view subspace clustering. In *Proceedings of the IEEE international conference on computer vision*. 4238–4246.

[6] Mehmet Gönen and Ethem Alpaydın. 2011. Multiple kernel learning algorithms. *The Journal of Machine Learning Research* 12 (2011), 2211–2268.

[7] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. 2017. Improved Deep Embedded Clustering with Local Structure Preservation.. In *IJCAI*. 1753–1759.

[8] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. 2017. Deep clustering with convolutional autoencoders. In *International conference on neural information processing*. Springer, 373–382.

[9] Yina Han, Kunde Yang, Yuanliang Ma, and Guizhong Liu. 2013. Localized multiple kernel learning via sample-wise alternating optimization. *IEEE transactions on cybernetics* 44, 1 (2013), 137–148.

[10] Yina Han, Kunde Yang, Yixin Yang, and Yuanliang Ma. 2016. Localized multiple kernel learning with dynamical clustering and matrix regularization. *IEEE transactions on neural networks and learning systems* 29, 2 (2016), 486–499.

[11] M. Hu and S. Chen. 2019. Doubly Aligned Incomplete Multi-view Clustering. (2019).

[12] Dong Huang, Jian-Huang Lai, and Chang-Dong Wang. 2015. Robust ensemble clustering using probability trajectories. *IEEE transactions on knowledge and data engineering* 28, 5 (2015), 1312–1326.

[13] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. 2011. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems* 20, 1 (2011), 120–134.

[14] Zhao Kang, Honghui Xu, Boyu Wang, Hongyuan Zhu, and Zenglin Xu. 2019. Clustering with similarity preserving. *Neurocomputing* 365 (2019), 211–218.

[15] Ritwik Kumar, Ting Chen, Moritz Hardt, David Beymer, Karen Brannon, and Tanveer Syeda-Mahmood. 2013. Multiple kernel completion and its application to cardiac disease discrimination. In *2013 IEEE 10th International Symposium on Biomedical Imaging*. IEEE, 764–767.

[16] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

[17] Teng Li, Yong Dou, Xinwang Liu, Yang Zhao, and Qi Lv. 2017. Multiple kernel clustering with corrupted kernels. *Neurocomputing* 267 (2017), 447–454.

[18] Zechao Li, Jinhui Tang, and Xiaofei He. 2017. Robust structured nonnegative matrix factorization for image representation. *IEEE transactions on neural networks and learning systems* 29, 5 (2017), 1947–1960.

[19] Hongfu Liu, Junjie Wu, Tongliang Liu, Dacheng Tao, and Yun Fu. 2017. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. *IEEE transactions on knowledge and data engineering* 29, 5 (2017), 1129–1143.

[20] Xinwang Liu, Miaomiao Li, Chang Tang, Jingyuan Xia, Jian Xiong, Li Liu, Marius Kloft, and En Zhu. 2020. Efficient and effective regularized incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence* (2020).

[21] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, Chang Tang, Jianping Yin, Dinggang Shen, Huaimin Wang, and Wen Gao. 2018. Late fusion incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence* 41, 10 (2018), 2410–2423.

[22] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, En Zhu, Tongliang Liu, Marius Kloft, Dinggang Shen, Jianping Yin, and Wen Gao. 2019. Multiple kernel k k-means with incomplete kernels. *IEEE transactions on pattern analysis and machine intelligence* 42, 5 (2019), 1191–1204.

[23] Xi Peng, Jiashi Feng, Shijie Xiao, Wei-Yun Yau, Joey Tianyi Zhou, and Songfan Yang. 2018. Structured autoencoders for subspace clustering. *IEEE Transactions on Image Processing* 27, 10 (2018), 5076–5086.

[24] Weixiang Shao, Lifang He, and S Yu Philip. 2015. Multiple Incomplete Views Clustering via Weighted Nonnegative Matrix Factorization with $L_{2,1}$ Regularization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 318–334.

[25] Anusua Trivedi, Piyush Rai, Hal Daumé III, and Scott L DuVall. 2010. Multiview clustering with incomplete views. In *NIPS workshop*, Vol. 224. 1–8.

[26] Jing Wang, Feng Tian, Hongchuan Yu, Chang Hong Liu, Kun Zhan, and Xiao Wang. 2017. Diverse non-negative matrix factorization for multiview data representation. *IEEE transactions on cybernetics* 48, 9 (2017), 2620–2632.

[27] Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. 2019. Multi-view Clustering via Late Fusion Alignment Maximization.. In *IJCAI*. 3778–3784.

[28] Yueqing Wang, Xinwang Liu, Yong Dou, and Rongchun Li. 2017. Multiple kernel clustering framework with improved kernels. *Discover* 1, 2 (2017), 3–4.

[29] Yang Wang, Lin Wu, Xuemin Lin, and Junbin Gao. 2018. Multiview spectral clustering via structured low-rank matrix factorization. *IEEE transactions on neural networks and learning systems* 29, 10 (2018), 4833–4843.

[30] Yang Wang, Wenjie Zhang, Lin Wu, Xuemin Lin, Meng Fang, and Shirui Pan. 2016. Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering. *arXiv preprint arXiv:1608.05560* (2016).

[31] Shuo Xiang, Lei Yuan, Wei Fan, Yalin Wang, Paul M Thompson, and Jieping Ye. 2013. Multi-source learning with block-wise missing data for alzheimer's disease prediction. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 185–193.

[32] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. PMLR, 478–487.

[33] Chang Xu, Dacheng Tao, and Chao Xu. 2015. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing* 24, 12 (2015), 5812–5825.

[34] Zuyuan Yang, Yu Zhang, Yong Xiang, Wei Yan, and Shengli Xie. 2018. Non-negative matrix factorization with dual constraints for image clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50, 7 (2018), 2524–2533.

[35] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. 2017. Latent multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4279–4287.

[36] Zhong Zhang, Zhili Qin, Peiyan Li, Qinli Yang, and Junming Shao. 2018. Multi-view discriminative learning via joint non-negative matrix factorization. In *International Conference on Database Systems for Advanced Applications*. Springer, 542–557.

[37] Xinzhong Zhu, Xinwang Liu, Miaomiao Li, En Zhu, Li Liu, Zhiping Cai, Jianping Yin, and Wen Gao. 2018. Localized Incomplete Multiple Kernel k-means.. In *IJCAI*. 3271–3277.

[38] Fuzhen Zhuang, Xuebing Li, Xin Jin, Dapeng Zhang, Lirong Qiu, and Qing He. 2017. Semantic feature learning for heterogeneous multitask classification via non-negative matrix factorization. *IEEE transactions on cybernetics* 48, 8 (2017), 2284–2293.