Efficient and Effective Regularized Incomplete Multi-view Clustering

Xinwang Liu, Miaomiao Li, Chang Tang, Jingyuan Xia, Jian Xiong, Li Liu, Marius Kloft, and En Zhu

Abstract—Incomplete multi-view clustering (IMVC) optimally combines multiple pre-specified incomplete views to improve clustering performance. Among various excellent solutions, the recently proposed multiple kernel k-means with incomplete kernels (MKKM-IK) forms a benchmark, which redefines IMVC as a joint optimization problem where the clustering and kernel matrix imputation tasks are alternately performed until convergence. Though demonstrating promising performance in various applications, we observe that the manner of kernel matrix imputation in MKKM-IK would incur intensive computational and storage complexities, over-complicated optimization and limitedly improved clustering performance. In this paper, we firstly propose an Efficient and Effective Incomplete Multi-view Clustering (EE-IMVC) algorithm to address these issues. Instead of completing the incomplete kernel matrices, EE-IMVC proposes to impute each incomplete base matrix generated by incomplete views with a learned consensus clustering matrix. Moreover, we further improve this algorithm by incorporating prior knowledge to regularize the learned consensus clustering matrix. Two three-step iterative algorithms are carefully developed to solve the resultant optimization problems with linear computational complexity, and their convergence is theoretically proven. After that, we theoretically study the generalization bound of the proposed algorithms. Furthermore, we conduct comprehensive experiments to study the proposed algorithms in terms of clustering accuracy, evolution of the learned consensus clustering matrix and the convergence. As indicated, our algorithms deliver their effectiveness by significantly and consistently outperforming some state-of-the-art ones.

Index Terms—multiple kernel clustering, multiple view learning, incomplete kernel learning

1 Introduction

Multi-view clustering (MVC) optimally integrates features from different views to improve clustering performance [1]. It has been intensively studied and widely applied into various applications during the last few decade [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. All these MVC algorithms assume that the views of samples are observable. However, in some practical applications [14], [15], this assumption may not hold anymore due to the absence of partial views among samples. The violation on this assumption makes the aforementioned MVC algorithms not applicable to handle incomplete multi-view clustering (IMVC) tasks.

Many efforts have been devoted to addressing IMVC, which can roughly be grouped into two categories. In the first category, the incomplete views are firstly filled with an imputation algorithm such as zero-filling, mean value filling, k-nearest-neighbor filling, expectation-maximization (EM) filling [16] and other advanced ones [17], [18], [19], [20], [21]. A standard MVC algorithm is subsequently applied into these imputed views to perform clustering tasks. This kind of algorithms are termed "two-stage" ones, where the imputation and clustering processes are separately carried out. By observing that the above-mentioned "two-stage" algorithms disconnect the processes of imputation and clustering, the other category, termed as "one-stage", puts forward to unify imputation and clustering into a single optimization procedure and instantiate a clustering-oriented algorithm termed as multiple kernel k-means with incomplete kernels (MKKM-IK) algorithm [22]. Specifically, the clustering result at the last iteration guides the imputation of absent kernel elements, and the latter is used in turn to conduct the subsequent clustering. By this way, these two procedures are seamlessly connected, with the aim to achieve better clustering performance.

Of the above-mentioned IMVC algorithms, the "one-stage" methods form a benchmark, where the incomplete views are optimized to best serve clustering. The main contribution of these methods is the unification of imputation and clustering, so that the imputation would be meaningful and beneficial for clustering. It has been demonstrated that the "one-stage" methods can achieve promissing clustering performance in various applications [22], [23], but they also suffer from the following non-ignorable drawbacks. i) *High computational and storage complexities*. Its computational and storage complexities are $\mathcal{O}(n^3)$ and $\mathcal{O}(mn^2)$ per iteration, respectively, where n and m are the number of samples and views. It prevents them from being applied to large-

- *X. Liu and E. Zhu are with College of Computer, National University of Defense Technology, Changsha, 410073, China. E-mail: {xinwangliu, enzhu}@nudt.edu.cn.*
- *M. Li is with Department of Computer, Changsha College, Changsha, China, 410073 (E-mail: miaomiaolinudt@gmail.com.)*
- *C. Tang is with School of Computer Science, China University of Geosciences, 430074 (E-mail: tangchang@cug.edu.cn)*
- *J. Xia is with Department of Electric and Electronic Engineering, Imperial College London, London, SW72AZ, UK (E-mail: j.xia16@imperial.ac.uk).*
- *J. Xiong is with School of Business Administration, Southwestern University of Finance and Economics, Chengdu, Sichuan, 611130, China (e-mail: xiongjian2017@swufe.edu.cn).*
- *L. Liu is with the College of System Engineering, National University of Defense Technology, Changsha, China, and also with the Center for Machine Vision and Signal Analysis, University of Oulu, 90014 Oulu, Finland (E-mail: li.liu@oulu.fi).*
- *M. Kloft is with Department of Computer Science, Technische Universität Kaiserslautern, Kaiserslautern, Germany, 67653. (E-mail: kloft@cs.uni-kl.de).*

Manuscript received January 28, 2020.

scale clustering tasks. ii) *Over-complicated imputation*. Existing "one-stage" methods directly impute multiple incomplete similarity matrices, in which the number of variables increases quadratically with the number of samples for each view. This could make the whole optimization over-complicated and also considerably increase the risk of falling into a low-quality local minimum. iii) *Limitedly improved clustering performance*. Note that a clustering result is determined by a whole similarity matrix in [22]. As a result, the imputation to an incomplete similarity matrix has impact to the clustering of all samples, no matter whether a sample is complete or not. When an imputation is not of high quality, it could adversely affect the clustering performance of all samples, especially for those with complete views.

All of the above issues signal that directly imputing the incomplete similarity matrices seems to be problematic and that a more efficient and effective approach shall be taken. In this paper, we propose efficient and effective incomplete multi-view clustering (EE-IMVC) to address these issues. EE-IMVC imputes each incomplete base clustering matrix generated by performing clustering on each separated incomplete similarity matrix, instead of itself. These imputed base clustering matrices are then used to learn a consensus clustering matrix, which is then employed to impute each incomplete base clustering matrix. These two steps are alternately performed until convergence. This idea is fulfilled by maximizing the alignment between the consensus clustering matrix and an adaptively weighted base clustering matrices with an optimal permutation. Though being theoretically elegant, we also observe that this algorithm does not sufficiently consider that learning the consensus clustering matrix could benefit from some other prior knowledge, besides the original orthogonal constraint. As a result, we further improve EE-IMVC by developing another variant, termed as efficient and effective regularized incomplete multi-view clustering (EE-R-IMVC). It explicitly designs a regularization term where the consensus clustering matrix is required to lie in the neighborhood of a pre-specified one. This prior knowledge is beneficial for the learning of the consensus clustering matrix, leading to improved clustering performance. We design two simple and computationally efficient algorithms to solve the resultant optimization problems by three singular value decomposition (SVD) per iteration, and analyze their computational and storage complexities and theoretically prove the convergence. After that, we conduct comprehensive experiments on six benchmark datasets to study the properties of the proposed algorithms, including the clustering accuracy with the various missing ratios, the evolution of the learned consensus matrix with iterations and the objective value with iterations. As demonstrated, EE-IMVC significantly and consistently outperforms the state-of-the-art methods in terms of clustering accuracy with much less running time. Meanwhile, we observe that the other proposed variant, i.e., EE-R-IMVC, further improves the clustering performance of EE-IMVC. It is expected that the simplicity and effectiveness of these clustering algorithms will make them a good option to be considered for practical applications where incomplete views are encountered.

This work is a substantially extended version of our original conference paper [24]. Its significant improvement over the previous one can be summarized as follows: 1) We design a new algorithm, termed EE-R-IMVC, by incorporating some prior knowledge on the consensus matrix into existing EE-IMVC, and develop an iterative algorithm to efficiently solve the resultant optimization problem. The prior knowledge can be treated as an initial clustering partition of data, which can be obtained by performing traditional clustering algorithms on imputed kernel matrices. It regularizes the learning of the consensus matrix, and this is beneficial for the newly proposed EE-R-IMVC to significantly outperform EE-IMVC proposed in the previous paper [24]. 2) We theoretically study the generalization bound of the proposed EE-IMVC and EE-R-IMVC on test data. 3) Besides more detailed discussion and extension, we also conduct more comprehensive experiments to validate the effectiveness of the proposed algorithms.

2 Related Work

2.1 Multiple Kernel k-means (MKKM)

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 $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$. These coefficients will be optimized during learning. Based on the definition of $\phi_{\boldsymbol{\beta}}(\mathbf{x})$, a kernel function can be expressed as $\kappa_{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\beta}}(\mathbf{x}_i)^\top\phi_{\boldsymbol{\beta}}(\mathbf{x}_j) = \sum_{p=1}^m \beta_p^2\kappa_p(\mathbf{x}_i, \mathbf{x}_j)$. 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 $\mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^m \beta_p^2\mathbf{K}_p$, the objective of MKKM can be written as

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

where $\mathbf{I}_k$ is an identity matrix with size of number of clusters k, and $\mathbf{H} = [\mathbf{h}_1^\top; \mathbf{h}_2^\top; \cdots; \mathbf{h}_n^\top] \in \mathbb{R}^{n\times k}$ is a clustering partition matrix. For each $\mathbf{h}_i = [h_{i1}, h_{i2}, \cdots, h_{ik}]^\top$ $(1 \leq i \leq n)$, $h_{ic} = 1/\sqrt{n_c}$ if $\mathbf{x}_i$ belongs to the c-th cluster $(1 \leq c \leq k)$, and 0 otherwise, where n_c is the number of samples belonging to the c-th cluster. It is not difficult to verify that $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k$. Note that the variables of $\mathbf{H}$ are discrete, which makes the optimization problem difficult to solve. However, one can approximate this problem through relaxing $\mathbf{H}$ to take arbitrary real values.

The optimization problem in Eq. (1) can be solved by alternately updating $\mathbf{H}$ and $\boldsymbol{\beta}$. Specifically, $\mathbf{H}$ is updated by given $\boldsymbol{\beta}$, and $\boldsymbol{\beta}$ is then optimized with updated $\mathbf{H}$. These two steps are alternately performed until convergence.

2.2 Multiple Kernel k-means with Incomplete Kernels (MKKM-IK)

The recently proposed MKKM-IK [22] has extended the existing MKKM in Eq. (1) to enable it to handle multiple kernel clustering with incomplete kernels. It unifies the imputation and clustering procedure into a single optimization objective and alternately optimizes each of them. That

is, i) imputing the absent kernels under the guidance of clustering; and ii) updating the clustering with the imputed kernels. The above idea is mathematically fulfilled as,

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

where $\mathbf{s}_p\,(1 \leq p \leq m)$ denote the sample indices for which the p-th view is present and $\mathbf{K}_p^{(cc)}$ be used to denote the kernel sub-matrix computed with these samples. The constraint $\mathbf{K}_p(\mathbf{s}_p, \mathbf{s}_p) = \mathbf{K}_p^{(cc)}$ is imposed to ensure that $\mathbf{K}_p$ maintains the known entries during the course. Different from the optimization in MKKM, [22] incorporates an extra step to impute the missing entries of base kernels, leading to a three-step alternate optimization algorithm. Interested readers are referred to [22].

Although MKKM-IK demonstrates excellent clustering performance in handling incomplete multi-view clustering tasks [22], it also suffers from the following non-ignorable drawbacks. Firstly, from the above optimization procedure, we observe that its computational complexity is $\mathcal{O}(n^3 + \sum_{p=1}^m n_p^3 + m^3)$ per iteration, where n, $n_p\,(n_p \leq n)$ and m are the number of all samples, observed samples of p-th view and views. During the learning procedure, it requires to store m base kernel matrices with size n. Therefore, its storage complexity is $\mathcal{O}(mn^2)$. The relatively high computational and storage complexities preclude it from being applied to large-scale clustering tasks. Furthermore, as seen from Eq. (2), there are $\frac{1}{2}(n-n_p)(n+n_p+1)$ elements to be imputed for the p-th incomplete base kernel matrix $\mathbf{K}_p(1 \leq p \leq m)$. It unnecessarily increases the complexity of the optimization and the risk of be trapped into a local minimum, adversely affecting the clustering performance. In addition, note that a clustering result is determined by a whole similarity matrix in [22]. As a result, the imputation to an incomplete similarity matrix has impact to the clustering of all samples, no matter whether a sample is complete or not. This improperly increases the influence of imputation on all samples, especially for those with complete views.

2.3 Late Fusion Incomplete Multi-view Clustering (LF-IMVC)

Instead of imputing incomplete similarity matrices $\{\mathbf{K}_p\}_{p=1}^m$, the work in [25] develops a late fusion incomplete multi-view clustering (LF-IMVC) algorithm, which proposes to impute the incomplete base clustering matrices to overcome the aforementioned disadvantages of MKKM-IK. It simultaneously performs clustering and the imputation of missing elements among base clustering matrices $\mathbf{H}_p \in \mathbb{R}^{n\times k}\,(1 \leq p \leq m)$, where the observed part of $\mathbf{H}_p$, denoted as $\mathbf{H}_p^{(0)} \in \mathbb{R}^{n_p\times k}\,(1 \leq p \leq m)$, can be obtained by solving kernel k-means in Eq. (2) with m incomplete base kernel matrices $\{\mathbf{K}_p(\mathbf{s}_p, \mathbf{s}_p)\}_{p=1}^m$.

Specifically, LF-IMVC firstly finds a consensus clustering matrix $\mathbf{H}$ from $\{\mathbf{H}_p\}_{p=1}^m$, and then imputes the incomplete parts of them with the learned consensus matrix. By this way, the above two learning processes can be seamlessly coupled and they are allowed to negotiate with each other to achieve better clustering. The above idea can be fulfilled as follows,

$$\begin{aligned} \max_{\mathbf{H},\{\mathbf{W}_p,\mathbf{H}_p\}_{p=1}^m} \ & \mathrm{Tr}\left[\mathbf{H}^\top\left(\sum_{p=1}^m \mathbf{H}_p\mathbf{W}_p\right)\right] + \lambda\sum_{p=1}^m \mathrm{Tr}\Big(\mathbf{H}_p^\top\hat{\mathbf{H}}_p^{(0)}\Big) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ & \mathbf{W}_p \in \mathbb{R}^{k\times k}, \ \mathbf{W}_p^\top\mathbf{W}_p = \mathbf{I}_k, \\ & \mathbf{H}_p \in \mathbb{R}^{n\times k}, \ \mathbf{H}_p^\top\mathbf{H}_p = \mathbf{I}_k, \end{aligned} \tag{3}$$

where $\mathbf{H}$ and $\mathbf{H}_p$ are the consensus clustering matrix and the p-th base clustering matrix, respectively, $\mathbf{W}_p$ is the p-th permutation matrix in order to optimally match $\mathbf{H}_p$ and $\mathbf{H}$, $\hat{\mathbf{H}}_p^{(0)}(\mathbf{s}_p, :) = \mathbf{H}_p^{(0)}$ with other elements being zeros and λ is a regularization parameter to trade of clustering and imputation. The orthogonal constraints are imposed on $\mathbf{H}$, $\mathbf{H}_p$ and $\mathbf{W}_p$ since they are clustering matrices and permutation matrix, respectively.

Although the recently proposed LF-IMVC [25] has some nice properties such as less imputation variables and higher computational efficiency compared with MKKM-IK [22], it also suffers from the following non-ignorable drawbacks. i) *More vulnerable to low-quality imputation*. As seen from Eq. (3), the observed part of each base clustering matrix $\mathbf{H}_p\,(1 \leq p \leq m)$ doesnot require to be kept 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, no matter whether they are complete. This improperly increases the impact of imputation on all samples, especially for those with complete views. ii) *Lack of Theoretical Guarantee*. Although LF-IMVC [25] experimentally demonstrates promising clustering performance in practical applications, it lacks of necessary theoretical analysis on the generalization error bound, which is important to theoretically justify its effectiveness. In addition, this theoretical analysis also provides a guidance to further improve the performance. In this work, we design two new IMVC algorithms to address the aforementioned issues, where the observed part of each base clustering matrix is strictly kept unchanged during the learning course. This, on one hand, is helpful to improve the computational efficiency by significantly reducing the number of variables to be filled. On the other hand, it also enhances the robustness to low-quality imputation. More importantly, we derive a generalization error bound for the proposed EE-IMVC and EE-R-IMVC, which provides the theoretical guarantee for the effectiveness of the proposed algorithms.

3 Efficient and Effective Incomplete Multi-view Clustering (EE-IMVC)

3.1 Formulation of EE-IMVC

In this section, we propose Efficient and Effective Incomplete Multi-view Clustering (EE-IMVC) which performs clustering and imputes the incomplete base clustering ma-

trices simultaneously. We firstly define the p-th $(1 \leq p \leq m)$ base clustering matrix as

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

where $\mathbf{H}_p^{(o)} \in \mathbb{R}^{n_p \times k}$ can be obtained by solving kernel k-means in Eq. (2) with m incomplete base kernel matrices $\{\mathbf{K}_p(\mathbf{s}_p, \mathbf{s}_p)\}_{p=1}^m$, while $\mathbf{H}_p^{(u)} \in \mathbb{R}^{(n-n_p) \times k}$ denote the incomplete part of $\mathbf{H}_p$ that is required to be filled. Note that other similarity based clustering algorithms such as spectral clustering can also be used to generate $\{\mathbf{H}_p^{(o)}\}_{p=1}^m$.

According to the above discussion, EE-IMVC proposes to simultaneously perform clustering and the imputation of $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ while keeping $\{\mathbf{H}_p^{(o)}\}_{p=1}^m$ unchanged during the learning course. Specifically, it firstly optimizes a consensus clustering matrix $\mathbf{H}$ from imputed $\{\mathbf{H}_p\}_{p=1}^m$, and then fill the $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ with $\mathbf{H}$. These two learning processes are seamlessly integrated. By doing so, they are allowed to coordinate with each other to achieve optimal clustering. The above idea can be fulfilled as follows,

$$\begin{aligned} \max_{\mathbf{H}, \{\mathbf{W}_p, \mathbf{H}_p^{(u)}, \beta_p\}_{p=1}^m} & \operatorname{Tr}\left[\mathbf{H}^\top \sum\nolimits_{p=1}^m \beta_p \begin{pmatrix} \mathbf{H}_p^{(o)} \\ \mathbf{H}_p^{(u)} \end{pmatrix} \mathbf{W}_p\right] \\ s.t.\ & \mathbf{H} \in \mathbb{R}^{n \times k},\ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \\ & \mathbf{W}_p \in \mathbb{R}^{k \times k},\ \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_k, \\ & \mathbf{H}_p^{(u)} \in \mathbb{R}^{(n-n_p) \times k},\ {\mathbf{H}_p^{(u)}}^\top \mathbf{H}_p^{(u)} = \mathbf{I}_k, \\ & \boldsymbol{\beta} \in \mathbb{R}^m,\ \sum\nolimits_{p=1}^m \beta_p^2 = 1,\ \beta_p \geq 0, \end{aligned} \tag{5}$$

where $\mathbf{H}$ and $\mathbf{H}_p^{(u)}$ are the consensus clustering matrix and the missing part of the p-th base clustering matrix, respectively, $\mathbf{W}_p$ is the p-th permutation matrix to optimally match $\mathbf{H}_p$ and $\mathbf{H}$, and $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_m]^\top$ is the adaptive weights of m base clustering matrices. Note that the orthogonal constraints are respectively imposed on $\mathbf{H}$ and $\mathbf{H}_p^{(u)}$ since they are clustering matrices. We also put an orthogonal constraint on $\mathbf{W}_p$ because it is a permutation matrix.

Compared with MKKM-IK [22], the objective function of EE-IMVC in Eq. (5) has the following nice properties. (1) *Less imputation variables*: The number of elements needs to be filled for the p-th view is $(n - n_p) \times k$, which is much less than $\frac{1}{2}(n - n_p) \times (n + n_p + 1)$ required by MKKM-IK. This could dramatically simplify the model and enhance its robustness to optimization. (2) *Less vulnerable to low-quality imputation*: In EE-IMVC, clustering on samples with complete views will not be affected by the imputation they are kept unchanged during the learning course. However, it is not the case for MKKM-IK because it needs to fill all incomplete elements and conduct eign-decomposition on the whole imputed similarity for clustering. This is helpful to make the proposed model be more robust in the whole course of optimization. (3) *More reasonable imputation*: EE-IMVC utilizes $\mathbf{H}$ to complete $\mathbf{H}_p^{(u)}$ rather than the incomplete base kernels matrices as in [22], which is more reasonable since both $\mathbf{H}$ and $\mathbf{H}_p^{(u)}$ reside in clustering partition space. Besides, our algorithm is parameter-free once the number of clusters to form is specified. These advantages significantly boosts the clustering performance, as demonstrated in the experimental part. In [24], a three-step iterative algorithm with proved convergence is designed to solve the optimization problem in Eq. (5). Interested readers can refer to [24] for the detail.

3.2 Efficient and Effective Regularized Incomplete Multi-view Clustering (EE-R-IMVC)

3.2.1 Prior Knowledge Encoded by $\mathbf{H}_0$

The proposed EE-IMVC in subsection 3.1 which jointly performs base clustering matrices completion and clustering is elegant, and achieves promising clustering performance as shown in the experimental part. As seen from Eq. (5), EE-IMVC imputes each base clustering matrix by only utilizing the consensus clustering matrix $\mathbf{H}$ and the imputed base clustering matrices are optimally combined to learn $\mathbf{H}$. As a result, it is crucial for EE-IMVC to learn an effective $\mathbf{H}$ in order to improve the clustering performance. However, apart from the orthogonal constraint, EE-IMVC does not utilize any auxiliary information to boost the optimization of $\mathbf{H}$. This could make the optimization with respect to $\mathbf{H}$ being trapped into a local minimum, which could further adversely affect the imputation of base clustering matrices, leading to unsatisfying clustering performance.

To address this issue, we aim to further improve the proposed EE-IMVC by incorporating useful prior knowledge, encoded by $\mathbf{H}_0$, to regularize the learning of $\mathbf{H}$. A question naturally raised is what kind of $\mathbf{H}_0$ is expected. We assume that $\mathbf{H}_0$ could be an initial clustering partition of data. For example, $\mathbf{H}_0$ can be the output of existing MKKM where the incomplete elements of each base kernel matrix can be filled with zeros, mean-value, EM algorithm, to name just a few. $\mathbf{H}_0$ can also be the output of existing kernel k-means (KKM) where the kernel is the average of all base kernel matrices with all missing elements filled with zeros. Further, there are other choices to generate $\mathbf{H}_0$. For example, $\mathbf{H}_0$ could be the output of MKKM-IK [22]. By regularizing the learning of the consensus clustering matrix with $\mathbf{H}_0$, the resultant algorithms can effectively avoid local optimum and demonstrate superior clustering performance. Finally, it is worth pointing out that only prior knowledge about the clusters is far from enough to well partition the data, as will be shown by the results in Table 3. As a result, we still need clustering the data even though we have prior knowledge about the clusters.

3.2.2 Formulation of EE-R-IMVC

Besides the orthogonal constraint, it is assumed that the consensus clustering matrix $\mathbf{H}$ resides in the neighborhood of a pre-specified $\mathbf{H}_0$, and minimizes $\|\mathbf{H} - \mathbf{H}_0\|_{\mathrm{F}}^2$ to guide the learning of $\mathbf{H}$, where $\mathbf{H}_0$ could be prior knowledge about the clusters. Note that minimizing $\|\mathbf{H} - \mathbf{H}_0\|_{\mathrm{F}}^2$ is equivalent to maximizing $\operatorname{Tr}(\mathbf{H}^\top \mathbf{H}_0)$. By integrating the above regularization term into the objective of EE-IMVC in Eq. (5), we obtain the objective function of the proposed efficient and effective regularized incomplete multi-view

clustering (EE-R-IMVC) as follows:

$$\begin{aligned}\max_{\mathbf{H},\{\mathbf{W}_p,\mathbf{H}_p^{(u)},\beta_p\}_{p=1}^m} & \operatorname{Tr}\left[\mathbf{H}^\top\sum_{p=1}^m\beta_p\begin{pmatrix}\mathbf{H}_p^{(o)}\\ \mathbf{H}_p^{(u)}\end{pmatrix}\mathbf{W}_p\right]+\lambda\operatorname{Tr}\Big(\mathbf{H}^\top\mathbf{H}_0\Big),\\ s.t.\ & \mathbf{H}\in\mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k,\\ & \mathbf{W}_p\in\mathbb{R}^{k\times k},\ \mathbf{W}_p^\top\mathbf{W}_p=\mathbf{I}_k,\\ & \mathbf{H}_p^{(u)}\in\mathbb{R}^{(n-n_p)\times k},\ {\mathbf{H}_p^{(u)}}^\top\mathbf{H}_p^{(u)}=\mathbf{I}_k,\\ & \boldsymbol{\beta}\in\mathbb{R}^m,\ \sum_{p=1}^m\beta_p^2=1,\ \beta_p\geq 0,\end{aligned}\tag{6}$$

where $\mathbf{H}$ and $\mathbf{H}_p^{(u)}$ are the consensus clustering matrix and the missing part of the p-th base clustering matrix, respectively, $\mathbf{W}_p$ is the p-th permutation matrix to optimally match $\mathbf{H}_p$ and $\mathbf{H}$, $\boldsymbol{\beta}=[\beta_1,\cdots,\beta_m]^\top$ is the adaptive weights of m base clustering matrices, $\mathbf{H}_0$ is an initial estimate of $\mathbf{H}$, and λ is the regularization parameter. Note that the orthogonal constraints are respectively imposed on $\mathbf{H}$ and $\mathbf{H}_p^{(u)}$ since they are clustering matrices. We also put an orthogonal constraint on $\mathbf{W}_p$ because it is a permutation matrix.

3.2.3 Alternate Optimization

Jointly optimizing $\mathbf{H}$, $\{\mathbf{H}_p^{(u)},\mathbf{W}_p\}_{p=1}^m$ and $\boldsymbol{\beta}$ in Eq. (6) is difficult. In the following, we design a simple and computationally efficient three-step algorithm to solve it alternately.

Solving $\mathbf{H}$ with fixed $\{\mathbf{W}_p,\mathbf{H}_p^{(u)}\}_{p=1}^m$ and $\boldsymbol{\beta}$. Given $\{\mathbf{W}_p,\mathbf{H}_p^{(u)}\}_{p=1}^m$ and $\boldsymbol{\beta}$, the optimization w.r.t $\mathbf{H}$ in Eq. (6) is equivalent to

$$\max_{\mathbf{H}}\ \operatorname{Tr}\Big(\mathbf{H}^\top\mathbf{T}\Big)\ s.t.\ \mathbf{H}\in\mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k,\tag{7}$$

where $\mathbf{T}=\sum_{p=1}^m\beta_p\mathbf{H}_p\mathbf{W}_p+\lambda\mathbf{H}_0$. As seen, in the proposed EE-R-IMVC, the optimization of $\mathbf{H}$ depends on both the base clustering matrix and the pre-specified $\mathbf{H}_0$, which is different from EE-IMVC. The optimization in Eq. (7) is a singular value decomposition (SVD) problem and can be efficiently solved with computational complexity $\mathcal{O}(nk^2)$.

Solving $\{\mathbf{W}_p\}_{p=1}^m$ with fixed $\mathbf{H}$, $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ and $\boldsymbol{\beta}$. Given $\mathbf{H}$, $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ and $\boldsymbol{\beta}$, the optimization w.r.t permutation matrix $\mathbf{W}_p$ in Eq. (6) equivalently reduces to,

$$\max_{\mathbf{W}_p}\ \operatorname{Tr}\Big(\mathbf{W}_p^\top\mathbf{Q}_p\Big)\ s.t.\ \mathbf{W}_p\in\mathbb{R}^{k\times k},\ \mathbf{W}_p^\top\mathbf{W}_p=\mathbf{I}_k,\tag{8}$$

where $\mathbf{Q}_p=\mathbf{H}_p^\top\mathbf{H}$. Again, it is a SVD optimization problem with computational complexity $\mathcal{O}(k^3)$.

Solving $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ with fixed $\{\mathbf{W}_p\}_{p=1}^m$, $\mathbf{H}$ and $\boldsymbol{\beta}$. Given $\mathbf{H}$, $\{\mathbf{W}_p\}_{p=1}^m$ and $\boldsymbol{\beta}$, the optimization w.r.t $\mathbf{H}_p^{(u)}$ in Eq. (5) is equivalent to

$$\begin{aligned}\max_{\mathbf{H}_p^{(u)}}\ & \operatorname{Tr}\Big({\mathbf{H}_p^{(u)}}^\top\mathbf{U}_p\Big)\\ s.t.\ & \mathbf{H}_p^{(u)}\in\mathbb{R}^{(n-n_p)\times k},\ {\mathbf{H}_p^{(u)}}^\top\mathbf{H}_p^{(u)}=\mathbf{I}_k,\end{aligned}\tag{9}$$

where $\mathbf{U}_p=\mathbf{H}(\hat{\mathbf{s}}_p,:)\mathbf{W}_p^\top$ and $\hat{\mathbf{s}}_p$ denotes the sample indices for which the p-th view is missing. Once again, it is a SVD problem and can be efficiently solved with computational complexity $\mathcal{O}((n-n_p)k^2)$.

Solving $\boldsymbol{\beta}$ with fixed $\mathbf{H}$ and $\{\mathbf{W}_p,\mathbf{H}_p^{(u)}\}_{p=1}^m$. Given $\mathbf{H}$ and $\{\mathbf{W}_p,\mathbf{H}_p^{(u)}\}_{p=1}^m$, the optimization w.r.t $\boldsymbol{\beta}$ in Eq. (6) is equivalent to

$$\max_{\boldsymbol{\beta}}\ \boldsymbol{\nu}^\top\boldsymbol{\beta}\ s.t.\ \boldsymbol{\beta}\in\mathbb{R}^m,\ \sum_{p=1}^m\beta_p^2=1,\ \beta_p\geq 0,\tag{10}$$

where $\boldsymbol{\nu}=[\nu_1,\nu_2,\cdots,\nu_m]$ with $\nu_p=\operatorname{Tr}(\mathbf{H}^\top\mathbf{H}_p\mathbf{W}_p)$.

As seen, the optimization in Eq. (10) has an analytical solution if $\nu_p\geq 0\,(1\leq p\leq m)$. The following Theorem 1 tells that the optimal weights of each base clustering matrix can be obtained analytically.

Theorem 1. *The optimal solution for Eq. (10) is $\boldsymbol{\beta}^\star=\boldsymbol{\nu}/\|\boldsymbol{\nu}\|$.*

Proof. Let $(\mathbf{H}^{(t)},\{\mathbf{H}_p^{(t)},\mathbf{W}_p^{(t)}\}_{p=1}^m)$ be the solution at the t-th iteration. We have $\nu_p^{(t)}=\operatorname{Tr}((\mathbf{H}^{(t)})^\top\mathbf{H}_p^{(t)}\mathbf{W}_p^{(t)})=\max_{\mathbf{H}_p^{(u)}}\operatorname{Tr}\big((\mathbf{H}^{(t)})^\top[{\mathbf{H}_p^{(o)}}^\top,{\mathbf{H}_p^{(u)}}^\top]^\top\mathbf{W}_p^{(t)}\big)\geq\max_{\mathbf{W}_p}\operatorname{Tr}\big((\mathbf{H}^{(t)})^\top[{\mathbf{H}_p^{(o)}}^\top,\big({\mathbf{H}_p^{(u)}}^{(t-1)}\big)^\top]^\top\mathbf{W}_p\big)>0$, $\forall p$. The proof is completed by taking the derivative of the Lagrangian function of Eq. (10) on β_p and letting it vanish. □

Algorithm 1 The Proposed EE-R-IMVC

1: **Input**: $\{\mathbf{H}_p^{(o)},\mathbf{s}_p\}_{p=1}^m$, k, $\mathbf{H}_0$, λ and ϵ_0.
2: **Output**: $\mathbf{H}$.
3: Initialize $\mathbf{W}_p^{(0)}=\mathbf{I}_k$, ${\mathbf{H}_p^{(u)}}^{(0)}=\mathbf{0}$, $\boldsymbol{\beta}^{(0)}=\mathbf{1}/\sqrt{m}$ and $t=1$.
4: **repeat**
5: Update $\mathbf{H}^{(t)}$ by solving Eq. (7) with $\{\mathbf{W}_p^{(t-1)},{\mathbf{H}_p^{(u)}}^{(t-1)}\}_{p=1}^m$ and $\boldsymbol{\beta}^{(t-1)}$.
6: Update $\{\mathbf{W}_p^{(t)}\}_{p=1}^m$ with $\mathbf{H}^{(t)}$, $\{{\mathbf{H}_p^{(u)}}^{(t-1)}\}_{p=1}^m$ and $\boldsymbol{\beta}^{(t-1)}$ by Eq. (8).
7: Update $\{{\mathbf{H}_p^{(u)}}^{(t)}\}_{p=1}^m$ with $\mathbf{H}^{(t)}$, $\{\mathbf{W}_p^{(t)}\}_{p=1}^m$ and $\boldsymbol{\beta}^{(t-1)}$ by Eq. (9).
8: Update $\boldsymbol{\beta}^{(t)}$ with $\mathbf{H}^{(t)}$, $\{\mathbf{W}_p^{(t)}\}_{p=1}^m$ and $\{{\mathbf{H}_p^{(u)}}^{(t)}\}_{p=1}^m$ by Eq. (10).
9: $t=t+1$.
10: **until** $\Big(\text{obj}^{(t)}-\text{obj}^{(t-1)}\Big)/\text{obj}^{(t-1)}\leq\epsilon_0$

In sum, our algorithm for solving Eq. (6) is outlined in Algorithm 1, where $\text{obj}^{(t)}$ denotes the objective value at the t-th iteration. The following Theorem 2 shows that Algorithm 1 is guaranteed to converge to a local maximum.

Theorem 2. *Algorithm 1 is guaranteed to converge to a local optimum.*

Proof. Note that for $\forall p$, $\operatorname{Tr}\big(\mathbf{H}^\top[{\mathbf{H}_p^{(o)}}^\top,{\mathbf{H}_p^{(u)}}^\top]^\top\mathbf{W}_p\big)\leq\frac{1}{2}[\operatorname{Tr}(\mathbf{H}^\top\mathbf{H})+\operatorname{Tr}(\mathbf{W}_p^\top[{\mathbf{H}_p^{(o)}}^\top,{\mathbf{H}_p^{(u)}}^\top][{\mathbf{H}_p^{(o)}}^\top,{\mathbf{H}_p^{(u)}}^\top]^\top\mathbf{W}_p)]=\frac{1}{2}[2k+\operatorname{Tr}(\mathbf{W}_p^\top{\mathbf{H}_p^{(o)}}^\top\mathbf{H}_p^{(o)}\mathbf{W}_p)]$. Note that the maximum of $\operatorname{Tr}(\mathbf{W}_p^\top{\mathbf{H}_p^{(o)}}^\top\mathbf{H}_p^{(o)}\mathbf{W}_p)$ with constraint $\mathbf{W}_p^\top\mathbf{W}_p=\mathbf{I}_k$ is $\sum_{j=1}^k\lambda_p^j$, where $\{\lambda_p^j\}_{j=1}^k$ are the k eigenvalue of ${\mathbf{H}_p^{(o)}}^\top\mathbf{H}_p^{(o)}$. We have $\operatorname{Tr}\big(\mathbf{H}^\top[{\mathbf{H}_p^{(o)}}^\top,{\mathbf{H}_p^{(u)}}^\top]^\top\mathbf{W}_p\big)\leq\frac{1}{2}[2k+\sum_{j=1}^k\lambda_p^j]\triangleq a_p$. Correspondingly, $\sum_{p=1}^m\beta_p\operatorname{Tr}\big(\mathbf{H}^\top[{\mathbf{H}_p^{(o)}}^\top,{\mathbf{H}_p^{(u)}}^\top]^\top\mathbf{W}_p\big)\leq\sum_{p=1}^m\beta_p a_p$, which is upper-bounded by $\sum_{p=1}^m\|a_p\|$ due to the ℓ_2-norm constraint on $\boldsymbol{\beta}$. Meanwhile, the objective of Algorithm 1 is guaranteed to be monotonically increased when optimizing one variable with others fixed at each iteration. As a result, our algorithm is guaranteed to converge to a local minimum. □

3.3 Discussion and Extension

We end up this section by analyzing the computational and storage complexities, the initialization of $\{\mathbf{H}_p^{(u)}, \mathbf{W}_p\}_{p=1}^m$ and potential extensions.

Computational complexity: As seen from Algorithm 1, the computational complexity of EE-IMVC and EE-R-IMVC is $\mathcal{O}(nk^2 + m(k^3 + (n - n_p)k^2))$ per iteration, where n, m and k are the number of samples, views and clusters, respectively. Therefore, EE-IMVC and EE-R-IMVC have a linear computational complexity with number of samples, which enables it more efficiently to handle large scale clustering tasks when compared with MKKM-IK [22].

Storage complexity: During the learning procedure, EE-IMVC and EE-R-IMVC need to store $\mathbf{H}$ and $\{\mathbf{H}_p, \mathbf{W}_p\}_{p=1}^m$. Its storage complexity is $\mathcal{O}(nk+mnk+mk^2)$, which is much less than that of MKKM-IK with $\mathcal{O}(mn^2)$ since $n \gg k$ in practice.

Initialization of $\{\mathbf{H}_p^{(u)}, \mathbf{W}_p\}_{p=1}^m$: In our current implementation, we simply initialize $\{\mathbf{H}_p^{(u)}\}_{p=1}^m$ as zeros, and $\{\mathbf{W}_p\}_{p=1}^m$ as identity matrix. This initialization has well demonstrated superior clustering performance of EE-IMVC and EE-R-IMVC in our experiments. Further exploring other initializations and studying their influence on the clustering performance will be an interesting future work.

Regularization on $\mathbf{H}$: The regularization on $\mathbf{H}$ is important to improve the subsequent clustering performance. In this work, we regularize $\mathbf{H}$ by assuming that it lies in the neighborhood of a pre-specified $\mathbf{H}_0$. In our current implementation, $\mathbf{H}_0$ is obtained by performing kernel k-means on unified multiple incomplete kernel matrices with zero-filling. Other approaches to generate $\mathbf{H}_0$ can also be designed to further improve the clustering performance. In addition, many task related prior knowledge such as low-rank can be incorporated to regularize $\mathbf{H}$, which is left as a piece of future work.

Extensions: EE-IMVC and EE-R-IMVC can be extended from the following aspects. Firstly, EE-IMVC and EE-R-IMVC could be further improved by sufficiently considering the correlation among $\{\mathbf{H}_p\}_{p=1}^m$. For example, we may build this correlation by criteria such as Kullback-Leibler (KL) divergence [26] and Hilbert-Schmidt independence criteria (HSIC), to name just a few. This prior knowledge could provide a good regularization on mutual base clustering matrix completion, and would be helpful to improve the clustering performance. Secondly, the way in generating $\{\mathbf{H}_p^{(o)}\}_{p=1}^m$ could be readily extendable to other similarity based clustering algorithms, such us spectral clustering [27]. This could further improve the clustering performance. Last but not least, the idea of joint imputation and clustering is so natural that can be generalized to other learning task such as feature missing.

4 Generalization Analysis of the Proposed Algorithms

The generalization error of k-means clustering has been studied by fixing the centroids obtained in the training process and generalizing them for testing [28], [29]. In this section, we derive generalization bounds of the proposed algorithms via exploiting the reconstruction error to study how the centroids obtained by the proposed EE-IMVC and EE-R-IMVC generalize onto unseen data.

Before defining the reconstruction error of k-means, we model the absence of views firstly. Specifically, let the indicator function $t(\mathbf{x}^{(p)})$ denote the absence of the p-th view of the observation $\mathbf{x}$, i.e., if the p-th view is observed, then $t(\mathbf{x}^{(p)}) = 1$; otherwise its value needs to be optimized. Note that $t(\mathbf{x}^{(p)})$ is a random variable depending on $\mathbf{x}$, whose distribution is unknown. Let $\hat{\mathbf{\Sigma}} = [\hat{\boldsymbol{\mu}}_1, \cdots, \hat{\boldsymbol{\mu}}_k]$ be the learned matrix composed of the k centroids, and $\hat{\boldsymbol{\beta}}$, $\{\mathbf{W}_p\}_{p=1}^m$ the learned kernel weights and permutation matrices by the proposed EE-IMVC and EE-R-IMVC. Effective k-means clustering algorithms should have the following reconstruction error small

$$\mathbb{E}\left[\min_{\mathbf{y}\in\{\mathbf{e}_1,\cdots,\mathbf{e}_k\}} \left\|\mathbf{h}_{\boldsymbol{\beta},\mathbf{t},\{\mathbf{W}_p\}_{p=1}^m}(\mathbf{x}_i) - \hat{\mathbf{\Sigma}}\mathbf{y}\right\|_{\mathrm{F}}^2\right], \tag{11}$$

where $\mathbf{h}_{\boldsymbol{\beta},\mathbf{t},\mathbf{W}}(\mathbf{x}_i) = \sum_{p=1}^m \beta_p t_p(\mathbf{x}_i^{(p)})\mathbf{W}_p^\top \mathbf{h}_p(\mathbf{x}_i^{(p)})$ and $\mathbf{e}_1, \ldots, \mathbf{e}_k$ form the orthogonal bases of $\mathbb{R}^k$. We show how the proposed algorithms achieve this goal.

Let us define a function class first:

$$\begin{aligned}\mathcal{F} = \Big\{ f : \mathbf{x} \mapsto & \min_{\mathbf{y}\in\{\mathbf{e}_1,\cdots,\mathbf{e}_k\}} \left\|\mathbf{h}_{\boldsymbol{\beta},\mathbf{t},\{\mathbf{W}_p\}_{p=1}^m}(\mathbf{x}_i) - \mathbf{\Sigma}\mathbf{y}\right\|_{\mathrm{F}}^2 \Big| \\ & \sum\nolimits_{p=1}^m \beta_p^2 = 1,\ \beta_p \geq 0,\ \mathbf{\Sigma} \in \mathbb{R}^{k\times k},\ \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_k, \\ & \mathbf{H}_p^{(u)\top} \mathbf{H}_p^{(u)} = \mathbf{I}_k,\ \forall p, \forall \mathbf{x}_i \in \mathcal{X} \Big\}.\end{aligned} \tag{12}$$

Theorem 3. *For any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f \in \mathcal{F}$:*

$$\begin{aligned}\mathbb{E}\left[f(\mathbf{x})\right] \leq \frac{1}{n}\sum_{i=1}^n f(\mathbf{x}_i) + \frac{\sqrt{2\pi}m\mathcal{G}_{1n}(\boldsymbol{\beta}, \mathbf{t}, \{\mathbf{W}_p\}_{p=1}^m, \{\mathbf{H}_p\}_{p=1}^m)}{n} \\ + \frac{\sqrt{2\pi}k(k+\sqrt{2})}{\sqrt{n}} + 4\sqrt{\frac{\log 1/\delta}{2n}},\end{aligned} \tag{13}$$

where

$$\begin{aligned}\mathcal{G}_{1n}(\boldsymbol{\beta}, \mathbf{t}, \{\mathbf{W}_p\}_{p=1}^m, \{\mathbf{H}_p\}_{p=1}^m) = \mathbb{E}_\gamma\Big[\sup_{\boldsymbol{\beta},\mathbf{t},\{\mathbf{W}_p,\mathbf{H}_p\}_{p=1}^m} \sum_{i=1}^n \\ \sum_{p,q=1}^m \gamma_{ipq}\beta_p\beta_q t_p(\mathbf{x}_i^{(p)}) t_q(\mathbf{x}_i^{(q)}) \mathbf{h}_p^\top(\mathbf{x}_i^{(p)}) \mathbf{W}_p \mathbf{W}_q^\top \mathbf{h}_q(\mathbf{x}_i^{(q)})\Big],\end{aligned} \tag{14}$$

and $\gamma_{ipq}, i \in \{1, \ldots, n\}, p, q \in \{1, \ldots, m\}$ are i.i.d. Gaussian random variables with zero mean and unit standard deviation.

Note that if all the views are accessible, we have $\mathcal{G}_{1n}(\boldsymbol{\beta}, t) \leq m^2\sqrt{n}$. This implies that with an ideal access to all views, the proposed algorithms will have generalization bounds of order $\mathcal{O}(\sqrt{1/n})$. However, when the number of absent views are increasing, the values of $\mathcal{G}_{1n}(\boldsymbol{\beta}, t)$ will become lager, making it more difficult to learn and more training examples are required to secure a given clustering accuracy.

According to Theorem 3, for any learned $\boldsymbol{\beta}$, $\{\mathbf{H}_p, \mathbf{W}_p\}_{p=1}^m$ and $\mathbf{\Sigma}$, to achieve a small

$$\mathbb{E}[f(\mathbf{x})] = \mathbb{E}\Big[\min_{\mathbf{y}\in\{\mathbf{e}_1,\cdots,\mathbf{e}_k\}} \left\|\mathbf{h}_{\boldsymbol{\beta},\mathbf{t},\{\mathbf{H}_p,\mathbf{W}_p\}_{p=1}^m}(\mathbf{x}_i) - \mathbf{\Sigma}\mathbf{y}\right\|_{\mathrm{F}}^2\Big], \tag{15}$$

TABLE 1: Datasets used in our experiments.

Dataset	#Samples	#Kernels	#Classes
Flower17	1360	7	17
Flower102	8189	4	102
CCV	6773	6	20
Caltech102-30	3060	48	102
UCI-Digital	2000	3	10
ProteinFold	694	12	27

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

$$\begin{aligned}
\sum_{i=1}^{n} f(\mathbf{x}_i) \;&=\; \sum_{i=1}^{n} \min_{\mathbf{y}\in\{\mathbf{e}_1,\cdots,\mathbf{e}_k\}} \|\mathbf{h}_{\boldsymbol{\beta},\mathbf{t},\mathbf{W}}(\mathbf{x}_i) - \boldsymbol{\Sigma}\mathbf{y}\|_{\mathrm{F}}^2 \\
&=\; \mathrm{Tr}\Big(\big(\sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\big)\big(\sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\big)^\top (\mathbf{I} - \mathbf{H}\mathbf{H}^\top)\Big) \\
&\leq\; \sum\nolimits_{p=1}^{m} \mathrm{Tr}\Big(\big(\beta_p \mathbf{H}_p \mathbf{W}_p\big)\big(\beta_p \mathbf{H}_p \mathbf{W}_p\big)^\top\Big) \\
&\quad - \mathrm{Tr}\Big(\big(\sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\big)\big(\sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\big)^\top \mathbf{H}\mathbf{H}^\top\Big) \\
&=\; 2k - \mathrm{Tr}\Big(\big(\sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\big)\big(\sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\big)^\top \mathbf{H}\mathbf{H}^\top\Big) \\
&\leq\; 2k - \frac{1}{k}\Big(\mathrm{Tr}\Big(\mathbf{H}^\top \sum\nolimits_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\Big)\Big)^2,
\end{aligned} \tag{16}$$

Eq. (16) implies that $2k - \frac{1}{k}\Big(\mathrm{Tr}\Big(\mathbf{H}^\top \sum_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\Big)\Big)^2$ shall be minimized to ensure a small $\sum_{i=1}^{n} f(\mathbf{x}_i)$ for good generalization. It is equivalent to maximize $\mathrm{Tr}\Big(\mathbf{H}^\top \sum_{p=1}^{m} \beta_p \mathbf{H}_p \mathbf{W}_p\Big)$, which is the objective of the proposed algorithms in Eq. (5) and Eq. (6). This also verifies the good generalization ability of the proposed algorithms. The detailed proof are provided in the supplemental material due to space limit.

5 Experiments

5.1 Experimental settings

The proposed EE-IMVC and EE-R-IMVC are experimentally evaluated on six widely used multiple kernel benchmark-data sets shown in Table 1. They are Oxford Flower17 and Flower102[1], Caltech102[2], Columbia Consumer Video (CCV)[3], UCI Digital[4] and Protein Fold Prediction[5]. For these datasets, all kernel matrices are pre-computed and can be publicly downloaded from the above websites. Their number of samples varies from one thousand to over eight thousands, clusters from ten to 102, and views from four to 48.

We compare EE-IMVC and EE-R-IMVC with several commonly used imputation methods, including zero filling (ZF), mean filling (MF), k-nearest-neighbor filling (KNN) and the alignment-maximization filling (AF) proposed in [17]. The widely used MKKM [30] is applied with these imputed base kernels. These two-stage methods are termed MKKM+ZF, MKKM+MF, MKKM+KNN and MKKM+AF, respectively. We also compare with the recently proposed MKKM-IK [22], which jointly optimizes the imputation and clustering. In addition, we compare EE-IMVC and EE-R-IMVC with late fusion IMVC (LF-IMVC) [25], which is regarded as the state-of-the-art in handling incomplete multi-view clustering tasks. Among all the compared algorithms, only LF-IMVC has one hyper-parameter to be tuned. In our experiments, we have reused the released Matlab codes and carefully tuned this hyper-parameter according to the setting up in [25] to produce their best possible results on each dataset for fair comparison.

For all data sets, it is assumed that the true number of clusters k is known and it is set as the true number of classes. We follow the approach in [22], [23], [25] to generate the missing vectors $\{\mathbf{s}_p\}_{p=1}^m$. The parameter ε, termed missing ratio in this experiment, controls the percentage of samples that have absent views, and it affects the performance of the algorithms in comparison. To show this point in depth, we compare these algorithms with respect to ε. Specifically, ε on all the datasets is set as $[0.0 : 0.1 : 0.9]$, where $\varepsilon = 0$ indicates that all views of data are available.

The widely used clustering accuracy (ACC), normalized mutual information (NMI), purity and rand index are applied to evaluate the clustering performance. For given $\mathbf{x}_i\,(1 \leq i \leq n)$, let c_i and y_i be its predicted cluster label and the provided ground-truth label, respectively. Let $\mathbf{c} = [c_1, \cdots, c_n]^\top$ and $\mathbf{y} = [y_1, \cdots, y_n]^\top$ denote the predicted cluster labels of a clustering algorithm and the provided ground-truth labels of $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, respectively. The clustering accuracy (ACC) is defined as follows,

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n}, \tag{17}$$

where $\delta(u, v)$ is the delta function that equals one if $u = v$ and equals zero otherwise, and $map(c_i)$ is the permutation mapping function that maps each cluster label c_i to the equivalent label from data. The best mapping can be found by using the Kuhn-Munkres algorithm [31]. The mutual information between $\mathbf{y}$ and $\mathbf{c}$, denoted as $\mathrm{MI}(\mathbf{y}, \mathbf{c})$, is defined as follows:

$$\mathrm{MI}(\mathbf{y}, \mathbf{c}) = \sum\nolimits_{y_i \in \mathbf{y},\, c'_j \in \mathbf{c}} p(y_i, c'_j) \log_2 \frac{p(y_i, c'_j)}{p(y_i)p(c'_j)}, \tag{18}$$

where $p(y_i)$ and $p(c'_j)$ are the probabilities that a sample arbitrarily selected from data belongs to the clusters y_i and c'_j, respectively, and $p(y_i, c'_j)$ is the joint probability that the arbitrarily selected samples belongs to the clusters y_i and c'_j at the same time. The normalized mutual information (NMI) is then defined as follows:

$$\mathrm{NMI}(\mathbf{y}, \mathbf{c}) = \frac{\mathrm{MI}(\mathbf{y}, \mathbf{c})}{\max\big(\mathrm{H}(\mathbf{y}), \mathrm{H}(\mathbf{c})\big)}, \tag{19}$$

where $\mathrm{H}(\mathbf{y})$ and $\mathrm{H}(\mathbf{c})$ are the entropies of $\mathbf{y}$ and $\mathbf{c}$, respectively.

For all algorithms, we repeat each experiment for 50 times with random initialization to reduce the effect of randomness caused by k-means, and report the best result. Meanwhile, we randomly generate the "incomplete" patterns for 30 times in the above-mentioned way and report the statistical results. The aggregated ACC, NMI, purity and rand index are used to evaluate the goodness of the

1. http://www.robots.ox.ac.uk/~vgg/data/flowers/
2. http://files.is.tue.mpg.de/pgehler/projects/iccv09/
3. http://www.ee.columbia.edu/ln/dvmm/CCV/
4. http://ss.sysu.edu.cn/~py/
5. http://mkl.ucsd.edu/dataset/protein-fold-prediction/

algorithms in comparison. Taking the aggregated ACC for example, it is obtained by averaging the averaged ACC achieved by an algorithm over different ε.

In the following parts, we conduct comprehensive experiments to study the properties of EE-IMVC and EE-R-IMVC from the following four aspects: clustering performance, the evolution of the learned consensus clustering matrix, regularization on clustering matrix $\mathbf{H}$, algorithm convergence and the sensitivity of EE-R-IMVC with the regularization parameter λ.

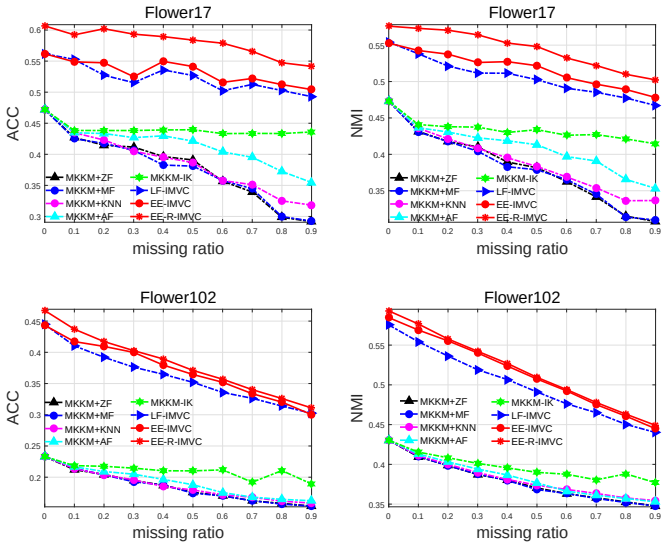


Fig. 1: ACC and NMI comparison with the variation of missing ratios on Flower17 and Flower102 datasets. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported. The Purity and Rand Index comparison are provided in the appendix due to space limit.

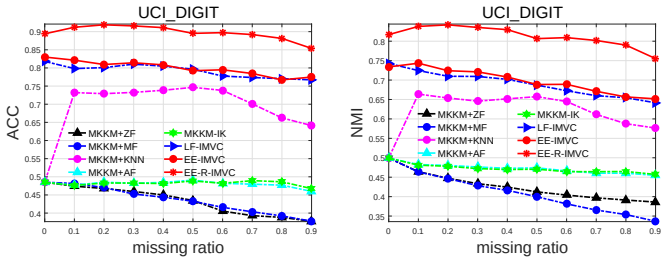


Fig. 2: ACC and NMI comparison with the variation of missing ratios on UCI Digital dataset. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported. The Purity and Rand Index comparison are provided in the appendix due to space limit.

5.2 Clustering Performance

We compare the proposed EE-IMVC and EE-R-IMVC with the aforementioned two-stage methods such as MKKM+ZF, MKKM+MF, MKKM+KNN and MKKM+AF, and one-stage methods such as MKKM-IK [22] and LF-IMVC [25] on Oxford Flower17 and Flower102, which have been widely used as MKL benchmark data sets [32]. There are seven views available for these two datasets. For each view, we apply a Gaussian kernel with the averaged pairwise distance as the width parameter to generate a kernel matrix. In this way, we

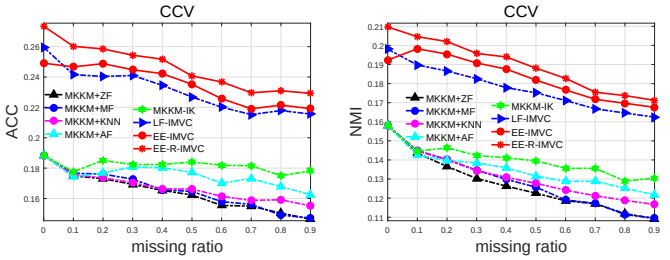


Fig. 3: ACC and NMI comparison with the variation of missing ratios on CCV dataset. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported. The Purity and Rand Index comparison are provided in the appendix due to space limit.

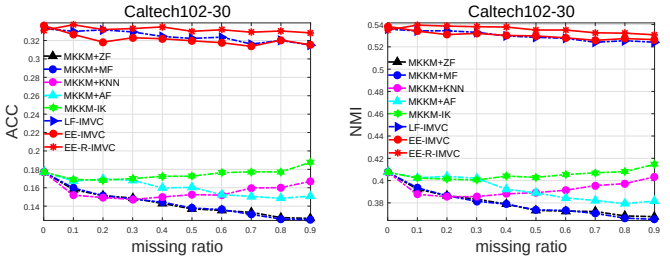


Fig. 4: ACC and NMI comparison with the variation of missing ratios on Caltech102-30 dataset. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported. The Purity and Rand Index comparison are provided in the appendix due to space limit.

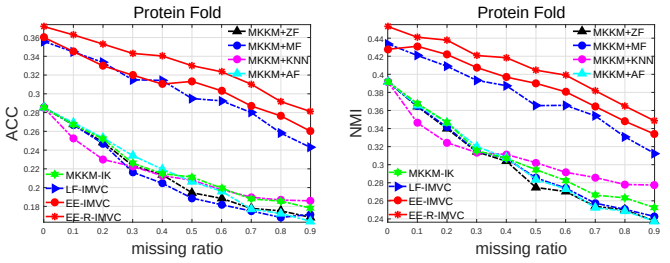


Fig. 5: ACC and NMI comparison with the variation of missing ratios on Protein Fold dataset. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported. The Purity and Rand Index comparison are provided in the appendix due to space limit.

TABLE 2: Aggregated ACC, NMI, purity and rand index comparison (mean±std) of different clustering algorithms on all benchmark datasets.

Datasets	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF [17]	MKKM-IK [22]	LF-IMVC [19]	EE-IMVC Proposed	EE-R-IMVC Proposed
ACC								
Flower17	36.96 ± 0.42	36.75 ± 0.66	37.75 ± 0.61	40.80 ± 0.40	43.67 ± 0.42	51.87 ± 0.69	52.96 ± 0.69	$\mathbf{57.72 \pm 0.71}$
Flower102	17.98 ± 0.16	17.95 ± 0.18	18.20 ± 0.16	18.73 ± 0.15	20.90 ± 0.16	35.26 ± 0.32	36.44 ± 0.31	$\mathbf{37.25 \pm 0.28}$
UCI-Digital	42.78 ± 0.44	43.00 ± 0.32	71.35 ± 0.94	47.98 ± 0.44	48.19 ± 0.47	78.89 ± 0.73	79.64 ± 0.51	$\mathbf{89.75 \pm 0.43}$
CCV	16.13 ± 0.11	16.29 ± 0.23	16.52 ± 0.18	17.37 ± 0.19	18.09 ± 0.23	22.81 ± 0.32	23.37 ± 0.39	$\mathbf{24.35 \pm 0.19}$
Caltech102-30	14.01 ± 0.13	14.00 ± 0.14	15.44 ± 0.18	15.86 ± 0.15	17.45 ± 0.18	32.36 ± 0.30	31.96 ± 0.28	$\mathbf{33.18 \pm 0.16}$
ProteinFold	20.64 ± 0.26	20.22 ± 0.24	20.95 ± 0.36	21.02 ± 0.39	21.36 ± 0.51	29.73 ± 0.39	30.50 ± 0.74	$\mathbf{32.62 \pm 0.37}$
NMI								
Flower17	37.30 ± 0.35	37.21 ± 0.34	38.22 ± 0.37	40.31 ± 0.30	42.99 ± 0.34	50.06 ± 0.49	51.38 ± 0.57	$\mathbf{54.17 \pm 0.41}$
Flower102	37.42 ± 0.14	37.38 ± 0.11	37.77 ± 0.11	37.90 ± 0.15	39.40 ± 0.10	49.31 ± 0.16	50.77 ± 0.09	$\mathbf{51.08 \pm 0.13}$
UCI-Digital	41.77 ± 0.15	39.90 ± 0.22	63.25 ± 0.49	46.98 ± 0.24	46.91 ± 0.26	68.45 ± 0.50	69.48 ± 0.42	$\mathbf{81.20 \pm 0.53}$
CCV	12.40 ± 0.10	12.58 ± 0.14	12.87 ± 0.09	13.25 ± 0.11	13.83 ± 0.17	17.52 ± 0.24	18.22 ± 0.22	$\mathbf{18.75 \pm 0.12}$
Caltech102-30	37.72 ± 0.11	37.66 ± 0.12	39.15 ± 0.08	39.08 ± 0.09	40.51 ± 0.14	52.90 ± 0.19	52.94 ± 0.13	$\mathbf{53.55 \pm 0.05}$
ProteinFold	28.99 ± 0.30	29.31 ± 0.27	30.34 ± 0.25	29.28 ± 0.31	29.96 ± 0.48	37.10 ± 0.37	38.60 ± 0.51	$\mathbf{40.19 \pm 0.30}$
Purity								
Flower17	38.46 ± 0.42	38.31 ± 0.61	39.19 ± 0.48	42.28 ± 0.31	45.11 ± 0.41	53.65 ± 0.72	54.66 ± 0.69	$\mathbf{59.01 \pm 0.63}$
Flower102	22.49 ± 0.17	22.44 ± 0.17	22.76 ± 0.17	23.17 ± 0.21	25.62 ± 0.18	40.37 ± 0.17	41.89 ± 0.18	$\mathbf{42.44 \pm 0.23}$
UCI-Digital	44.71 ± 0.44	43.30 ± 0.31	71.47 ± 0.66	50.42 ± 0.35	50.84 ± 0.41	78.94 ± 0.63	79.69 ± 0.51	$\mathbf{89.75 \pm 0.43}$
CCV	20.36 ± 0.11	20.63 ± 0.15	20.73 ± 0.10	21.21 ± 0.13	21.90 ± 0.20	25.86 ± 0.34	26.46 ± 0.42	$\mathbf{27.36 \pm 0.18}$
Caltech102-30	15.35 ± 0.18	15.29 ± 0.17	16.97 ± 0.12	17.06 ± 0.17	18.84 ± 0.15	34.56 ± 0.34	34.25 ± 0.23	$\mathbf{35.32 \pm 0.10}$
ProteinFold	26.95 ± 0.36	27.00 ± 0.42	27.76 ± 0.34	27.25 ± 0.47	27.70 ± 0.54	35.76 ± 0.38	36.99 ± 0.72	$\mathbf{38.99 \pm 0.41}$
Rand Index								
Flower17	20.05 ± 0.37	19.92 ± 0.41	20.83 ± 0.35	22.81 ± 0.31	25.57 ± 0.30	33.99 ± 0.64	35.29 ± 0.62	$\mathbf{39.07 \pm 0.67}$
Flower102	8.11 ± 0.12	8.06 ± 0.14	8.32 ± 0.11	8.66 ± 0.13	10.27 ± 0.14	22.22 ± 0.27	23.57 ± 0.29	$\mathbf{24.23 \pm 0.20}$
UCI-Digital	25.46 ± 0.19	22.14 ± 0.22	51.55 ± 0.69	30.86 ± 0.31	31.05 ± 0.27	62.89 ± 0.61	64.20 ± 0.48	$\mathbf{79.31 \pm 0.70}$
CCV	4.57 ± 0.06	4.64 ± 0.09	4.74 ± 0.05	5.02 ± 0.08	5.45 ± 0.09	7.80 ± 0.14	8.17 ± 0.13	$\mathbf{8.65 \pm 0.07}$
Caltech102-30	4.06 ± 0.09	4.03 ± 0.09	5.28 ± 0.12	5.58 ± 0.11	6.75 ± 0.14	18.20 ± 0.18	18.06 ± 0.21	$\mathbf{18.95 \pm 0.13}$
ProteinFold	6.68 ± 0.23	6.53 ± 0.18	7.06 ± 0.24	6.93 ± 0.27	7.21 ± 0.32	12.94 ± 0.32	14.36 ± 0.54	$\mathbf{15.83 \pm 0.37}$

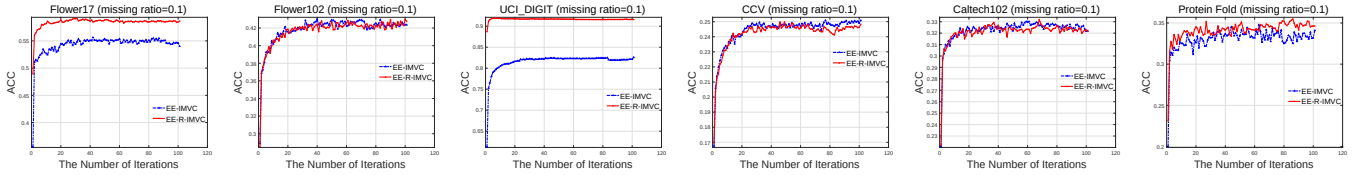


Fig. 6: The evolution of the learned consensus clustering matrix $\mathbf{H}$ by EE-IMVC and EE-R-IMVC with missing ratio 0.1 on all datasets. The curves with other missing ratios are similar and we omit them due to space limit.

obtain seven base kernels, and use them for all the multi-view clustering algorithms compared in our experiment.

Figure 1 presents the ACC, NMI, purity and rand index comparison of the above algorithms with different missing ratios on these two datasets. From this figure, we have the following observations:

- The proposed MKKM-IK [22] (in green) outperforms existing two-stage imputation methods. For example, it exceeds the best two-stage imputation method (MKKM+AF) by 0.4%, 0.5%, 1.2%, 0.9%, 1.8%, 2.9%, 3.9%, 6.1% and 8.1% in terms of ACC, with the variation of missing ratios in $[0.1, \cdots, 0.9]$ on Flower17. The improvement is more significant with the increase of missing ratios. These results well demonstrates the effectiveness of its joint optimization on imputation and clustering.
- The recently proposed LF-IMVC [25] (in blue) further improve MKKM-IK [22]. For example, it improves the latter by 11.4%, 8.9%, 7.7%, 9.7%, 8.7%, 6.9%, 7.9%, 6.9% and 5.7% in terms of ACC with the variation of missing ratios in $[0.1, \cdots, 0.9]$ on Flower17. These results verify the effectiveness of imputing base clustering matrices rather than kernel matrices.
- Our EE-IMVC achieves comparable or slightly better performance than LF-IMVC [25]. Moreover, EE-R-IMVC significantly and consistently outperforms EE-IMVC. Taking the results on Flower17 for example. It improves the EE-IMVC by 4.4%, 5.5%, 6.9%, 4.0%, 4.3%, 6.3%, 4.4%, 3.5% and 3.7% in terms of ACC with the variation of missing ratios in $[0.1, \cdots, 0.9]$, indicating the effectiveness of incorporating regularization on the consensus clustering matrix.
- The proposed EE-IMVC and EE-R-IMVC also

demonstrate superior clustering performance when all the views of data are available.

- The curves in terms of ACC and NMI on Flower102 are plotted in sub-figures 1c-1d, which is similar to the results on Flower17.

UCI-Digital dataset has been widely used as a benchmark in multi-view clustering [22], [25]. We also compare the clustering performance of the aforementioned algorithms on this dataset. The clustering accuracy, NMI, purity and rand index of these algorithms with the variation of missing ratio are plotted in Figure 2. From Figure 2a, we observe that the proposed MKKM-IK gives poor performance on this dataset, which is clearly inferior to the MKKM+KNN. The proposed LF-IMVC [25] significantly improves this situation, demonstrating superior clustering performance. Our proposed EE-IMVC achieve comparable or slightly better performance than LF-IMVC, and EE-R-IMVC further significantly and consistently outperforms the latter. For example, EE-R-IMVC exceeds LF-IMVC by 11.4%, 11.8%, 10.5%, 10.6%, 9.9%, 12.0%, 11.8%, 11.1% and 8.7% in terms of ACC with the variation of missing ratios. In addition, the result in terms of NMI is similar, as seen from sub-figure 2b.

We evaluate the performance of the proposed algorithms on CCV dataset, and report the results in Figure 3. We once again observe that the proposed EE-IMVC and EE-R-IMVC significantly outperform the compared ones in terms of ACC, NMI, purity and rand index. For example, EE-IMVC slightly improves the performance of the second best one (LF-IMCV), and EE-R-IMVC further significantly increases the improvement by 1.9%, 1.8%, 1.3%, 1.7%, 1.4%, 1.7%, 1.4%, 1.3% and 1.4% in terms of ACC. The result in terms of NMI is similar, as shown in sub-figure 3b.

We conduct another experiment on the Caltech102 dataset to evaluate the performance of the proposed algorithms. This dataset consists of a group of kernels derived from various visual features computed on the Caltech-102 object recognition task with 102 categories. It has 48 base kernels which are publicly available. The ACC and NMI of the aforementioned algorithms with the variation of missing ratios are plotted in sub-figures 4a-4b, respectively. As seen, the proposed EE-IMVC and EE-R-IMVC demonstrate comparable or better clustering clustering performance than the state-of-the-art one in the literature.

Besides the above five visual datasets, we finally compare the aforementioned algorithms on the protein fold dataset, which is a multi-source and multi-class dataset based on a subset of the PDB-40D SCOP collection. It contains 12 different feature spaces, including composition, secondary, hydrophobicity, volume, polarity, polarizability, L1, L4, L14, L30, SWblosum62 and SWpam50. This dataset has been widely adopted in the MKL community [33], [34]. For the protein fold dataset, the input features are available and the kernel matrices are generated as in [33], where the second order polynomial kernels are employed for feature sets one to ten and the linear kernel for the rest two feature sets.

The clustering performance of these algorithms are plotted in sub-figures 5a-5b. From these sub-figures, we observe that the proposed EE-IMVC demonstrates slightly better clustering performance than the second best one (LF-IMVC), and EE-R-IMVC further consistently and significantly improves EE-IMVC. For example, EE-R-IMVC exceeds LF-IMVC by 1.8%, 1.9%, 2.8%, 2.6%, 3.5%, 3.1%, 3.0%, 3.4% and 3.8% in terms of ACC with the missing ratios. Meanwhile, we observe that the results in terms of NMI are also similar.

We also report the aggregated ACC, NMI, purity and rand index, and the standard deviation in Table 2, where the one with the highest performance is shown in bold. Again, we observe that the proposed EE-R-IMVC significantly outperforms MKKM+ZF, MKKM+MF, MKKM+KNN, MKKM+AF, MKKM-IK and LF-IMVC. For example, EE-R-IMVC exceeds the second best one (LF-IMVC) by 5.9%, 2.0%, 10.9%, 1.5%, 0.8% and 2.9% in terms of ACC on Flower17, Flower102, UCI-Digital, CCV, Caltech102 and ProteinFold, respectively. These results are consistent with our observations in Figures 1, 2, 3, 4, 5.

The above experimental results on these datasets have well demonstrated that EE-IMVC and EE-R-IMVC are superior to some state-of-the-art in terms of ACC, NMI, purity and rand index. We attribute the superiority of EE-IMVC and EE-R-IMVC as three aspects: i) Completing the incomplete base clustering matrices with the consensus one. Different from MKKM-IK where the consensus clustering matrix $\mathbf{H}$ is utilized to fill incomplete base kernels, EE-IMVC and EE-R-IMVC impute each incomplete base clustering matrix with $\mathbf{H}$. The latter is more natural and reasonable since both $\mathbf{H}$ and incomplete base clustering matrices reside in the same clustering space, leading to more suitable imputation. ii) The joint optimization on imputation and clustering. On one hand, the imputation is guided by the clustering results, which makes the imputation more directly targeted at the ultimate goal. On the other hand, this meaningful imputation is beneficial to refine the clustering results. These factors bring forth the significant improvements on clustering performance. iii) The regularization on the consensus clustering matrix. We can incorporate useful prior knowledge to help the learning of $\mathbf{H}$, which in turn boosts the imputation of incomplete base clustering matrices, leading to improved clustering performance.

5.3 Effectiveness of the Learned Consensus Matrix

We conduct extra experiments to show the evolution of the learned consensus clustering matrix $\mathbf{H}$ during the learning procedure. Specifically, we evaluate the ACC, NMI, purity and rand index of EE-IMVC and EE-R-IMVC based on the $\mathbf{H}$ learned at each iteration on the aforementioned datasets, and plot the curves in Figure 6. Taking the results in terms of ACC for example, we observe that i) the ACC of EE-IMVC and EE-R-IMVC gradually increases to a maximum and generally maintains it up to slight variation, and ii) the curves corresponding to EE-R-IMVC is usually on the above of EE-IMVC. These observations have clearly demonstrated the effectiveness of learned consensus clustering matrix, indicating the advantage of regularizing the consensus clustering matrix. Other curves in terms of NMI, purity and rand index have similar trend.

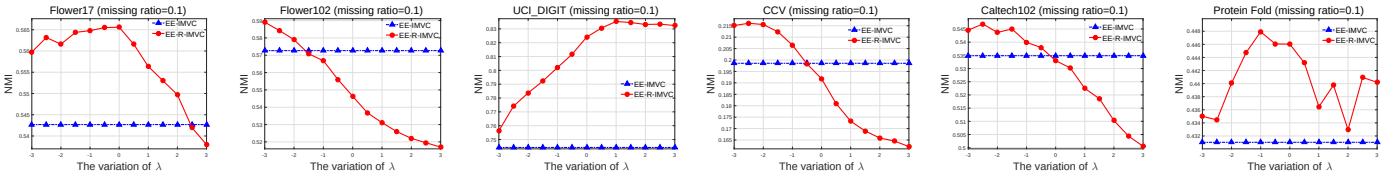


Fig. 7: The sensitivity of EE-R-IMVC with the variation of λ with missing ratio 0.1 on Flower17, Flower102, UCI-Digtal, CCV, Caltech102-30 and ProteinFold datasets. The results of EE-IMVC are also provided as a reference. The results in terms of ACC, Purity and Rand Index with other missing ratios are similar and omitted due to space limit.

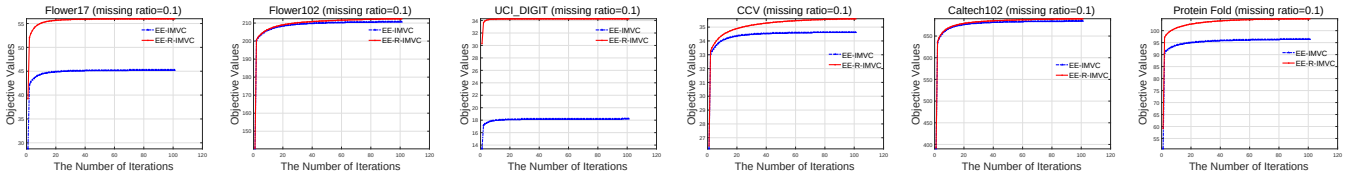


Fig. 8: The objective values of EE-IMVC and EE-R-IMVC with iterations with missing ratio 0.1 on all datasets. The curves with other missing ratios are similar and we omit them due to space limit.

5.4 Empirical Study on Regularizing $\mathbf{H}$

In this subsection, we firstly clarify the motivation of incorporating prior knowledge to improve the clustering by conducting an ablation study on all benchmark datasets. Secondly, we try the best to explore what kind of prior knowledge is expected by designing different $\mathbf{H}_0$s.

We empirically observe that, apart from the orthogonal constraint, some prior knowledge on $\mathbf{H}$ may be helpful to boost its optimization, leading to improved clustering performance. To see this point in depth, we design an ablation study to verify the effectiveness of incorporating $\mathbf{H}_0$ on all benchmark datastes. The clustering algorithms include: 1) clustering data with only prior knowledge $\mathbf{H}_0$, 2) clustering data without prior knowledge (i.e., EE-IMVC), and 3) clustering data with EE-IMVC and prior knowledge (i.e., EE-R-IMVC). The experimental results are reported in Table 3. From these results, we have the following observations.

- The clustering performance with only prior knowledge $\mathbf{H}_0$ is usually inferior to that of EE-IMVC and EE-R-IMVC. This indicates that only prior knowledge about the clusters is far from enough to well partition the data. As a result, we still need clustering the data even though we have prior knowledge about the clusters.
- The clustering performance of EE-IMVC is inferior to that of EE-R-IMVC. This demonstrates that the prior knowledge about the clusters is indeed helpful to improve the clustering, indicating the necessarity of incorporating prior knowledge.

These experimental results well explain the effectiveness of incorporating prior knowledge in optimizing $\mathbf{H}$ and improving clustering.

We then try to explore what kind of prior knowledge is expected by designing two different $\mathbf{H}_0$s, i.e., $\mathbf{H}_0^{(1)}$ and $\mathbf{H}_0^{(2)}$. 1) $\mathbf{H}_0^{(1)}$: We first impute the missing parts of each base kernel matrix with zeros, and combine them with unified weight. It is then taken as the input of kernel k-means to generate $\mathbf{H}_0^{(1)}$, and 2) $\mathbf{H}_0^{(2)}$: The incomplete parts of base kernels are firstly filled with zeros. These imputed base kernel matrices are then taken as the input of multiple kernel k-means (MKKM) to output $\mathbf{H}_0^{(2)}$. The experimental results with different $\mathbf{H}_0$s are reported in Table 3. From these results, we observe that:

- Different prior knowledge encoded by $\mathbf{H}_0$s produces different clustering performance.
- By integrating different $\mathbf{H}_0$s, EE-R-IMVC consistently and significantly outperforms EE-IMVC in terms of ACC, NMI, purity and rand index.

These results indicate that the prior knowledge on $\mathbf{H}$ is able to boost its optimization, leading to improved clustering performance. Also, there are other choices to generate $\mathbf{H}_0$. For example, $\mathbf{H}_0$ could be the output of MKKM-IK [22]. We will further explore the affect of different $\mathbf{H}_0$s on clustering in the future work.

5.5 Parameter Sensitivity

As can be seen in Eq. (6), EE-R-IMVC introduces the regularization parameter λ to trade off the clustering and regularization. In the following, we conduct experiments to show the effect of this parameter on the clustering performance on all datasets. Figure 7 presents the NMI of EE-R-IMVC by varying λ from 2^{-3} to 2^3, where the EE-IMVC is also provided as a baseline. From these figures, we observe that the NMI first increases to a high value and generally maintains it up to slight variation with the increasing value of λ. EE-R-IMVC demonstrates stable performance across a wide range of λ. These experiments have well shown that EE-R-IMVC is insensitive to the variation of the parameter.

5.6 Convergence

Our algorithms are theoretically guaranteed to converge according to Theorem 2. We record the objective values of

TABLE 3: ACC, NMI, purity and rand index comparison (mean±std) of different clustering algorithms on all benchmark datasets (with missing ratio=0.1).

Datasets	Clustering with only $\mathbf{H}_0$		EE-IMVC	EE-R-IMVC	
	$\mathbf{H}_0^{(1)}$	$\mathbf{H}_0^{(2)}$		$\mathbf{H}_0^{(1)}$	$\mathbf{H}_0^{(2)}$
ACC					
Flower17	52.63 ± 2.19	42.69 ± 1.16	54.88 ± 2.59	62.47 ± 2.41	58.56 ± 1.30
Flower102	32.02 ± 0.46	21.41 ± 0.32	42.70 ± 0.96	43.45 ± 0.78	43.20 ± 0.59
UCI-Digital	90.86 ± 4.42	47.41 ± 0.76	82.14 ± 2.01	93.74 ± 1.86	82.39 ± 1.86
CCV	17.94 ± 0.25	17.49 ± 0.36	24.67 ± 0.70	25.81 ± 0.53	25.64 ± 0.58
Caltech102-30	26.64 ± 0.73	16.05 ± 0.41	32.33 ± 1.21	33.41 ± 0.55	33.37 ± 0.46
ProteinFold	29.08 ± 0.83	26.77 ± 1.10	34.35 ± 2.89	36.28 ± 1.60	36.35 ± 1.61
NMI					
Flower17	52.93 ± 1.29	43.16 ± 0.74	54.26 ± 1.36	59.36 ± 0.88	56.12 ± 0.90
Flower102	50.81 ± 0.27	41.00 ± 0.18	57.22 ± 0.41	57.57 ± 0.35	57.56 ± 0.33
UCI-Digital	86.35 ± 2.03	46.42 ± 0.62	74.34 ± 1.35	87.73 ± 1.34	74.57 ± 1.34
CCV	15.34 ± 0.30	14.35 ± 0.31	19.83 ± 0.34	20.40 ± 0.25	20.19 ± 0.29
Caltech102-30	48.88 ± 0.51	39.44 ± 0.32	53.27 ± 0.57	54.02 ± 0.27	53.88 ± 0.29
ProteinFold	39.91 ± 0.64	36.43 ± 0.86	43.05 ± 1.23	44.68 ± 0.84	44.28 ± 0.80
Purity					
Flower17	55.63 ± 1.64	44.46 ± 0.95	56.63 ± 2.23	63.22 ± 2.04	59.94 ± 1.36
Flower102	39.51 ± 0.43	26.32 ± 0.18	48.98 ± 0.81	49.70 ± 0.75	49.53 ± 0.68
UCI-Digital	91.51 ± 3.15	50.11 ± 0.92	82.15 ± 2.00	93.74 ± 1.34	82.41 ± 1.84
CCV	21.60 ± 0.20	21.72 ± 0.28	27.78 ± 0.54	28.94 ± 0.45	28.87 ± 0.54
Caltech102-30	29.07 ± 0.56	17.43 ± 0.48	34.60 ± 1.14	35.64 ± 0.44	35.70 ± 0.49
ProteinFold	37.12 ± 1.10	33.04 ± 1.00	41.50 ± 1.78	43.29 ± 1.29	43.10 ± 1.02
Rand Index					
Flower17	35.46 ± 2.33	25.45 ± 0.75	38.16 ± 2.00	44.70 ± 1.66	40.91 ± 1.31
Flower102	19.33 ± 0.37	11.07 ± 0.19	29.34 ± 1.04	29.86 ± 0.61	29.77 ± 0.66
UCI-Digital	84.02 ± 3.79	30.22 ± 0.69	69.46 ± 1.94	86.85 ± 2.33	69.69 ± 1.92
CCV	6.04 ± 0.09	5.43 ± 0.15	9.04 ± 0.27	9.59 ± 0.27	9.42 ± 0.31
Caltech102-30	13.39 ± 0.45	5.66 ± 0.22	18.35 ± 0.94	19.48 ± 0.51	19.33 ± 0.44
ProteinFold	13.19 ± 0.69	11.42 ± 0.71	17.89 ± 1.94	19.91 ± 1.12	19.75 ± 0.91

EE-IMVC and EE-R-IMVC with iterations on all datasets and plot them in Figure 8. As observed, the objective value of EE-IMVC and EE-R-IMVC does monotonically increase at each iteration and that it usually converges in less than 50 iterations.

6 Conclusion

While the recently proposed MKKM-IK [22] is able to handle incomplete multi-view clustering, the relatively high computational and space complexities prevent it from large scale clustering tasks. This paper firstly proposes the EE-IMVC to simultaneously clustering and imputing the incomplete base clustering matrices. We further improve EE-IMVC by incorporating prior knowledge to regularize the learning of the consensus clustering matrix. We develop two four-step algorithms to effectively and efficiently solves the resultant optimization problems. In addition, we analyze and derive the generalization error bound of the proposed EE-IMVC and EE-R-IMVC. Extensive experiments on benchmark datasets have been conducted and the results well demonstrate the superiority of our algorithms.

Although demonstrating improvements compared to others, the proposed EE-IMVC and EE-R-IMVC can be further improved from the following aspects: 1) As shown in Eq. (9) in the manuscript, the update of each $\mathbf{H}_p^{(u)}$ only depends on $\mathbf{H}(\hat{\mathbf{s}}_p, :)$. We plan to sufficiently utilize other views $\{\mathbf{H}_q\}_{q=1,q\neq p}^m$ to improve the imputation of $\mathbf{H}_p^{(u)}$, leading to improved clustering performance. 2) As pointed by reviewers, the prior knowledge encoded by $\mathbf{H}_0$ has significant affect on clustering performance. In the future, we will further explore how to automatically learn an optimal $\mathbf{H}_0$ from data by following the idea of optimal neighborhood kernel learning [7].

Acknowledgements

This work was supported by the Natural Science Foundation of China (project no. 61773392, 61672528 and 61701451).

References

[1] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004)*, 2004, pp. 19–26.

[2] S. Yu, L.-C. Tranchevent, X. Liu, W. Glänzel, J. A. K. Suykens, B. D. Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE TPAMI*, vol. 34, no. 5, pp. 1031–1039, 2012.

[3] S. Li, Y. Jiang, and Z. Zhou, "Partial multi-view clustering," in *AAAI*, 2014, pp. 1968–1974.

[4] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means clustering using ℓ_{21}-norm." in *IJCAI*, 2015, pp. 3476–3482.

[5] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *AAAI*, 2016, pp. 1888–1894.

[6] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *IJCAI*, 2016, pp. 1704–1710.

[7] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *AAAI*, 2017, pp. 2266–2272.

[8] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *AAAI*, 2015, pp. 2750–2756.

[9] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *IJCAI*, 2013, pp. 2598–2604.

[10] Z. Tao, H. Liu, and Y. Fu, "Simultaneous clustering and ensemble," in *AAAI*, 2017, pp. 1546–1552.

[11] J. Liu, C. Wang, M. Danilevsky, and J. Han, "Large-scale spectral clustering on graphs," in *IJCAI*, 2013, pp. 1486–1492.

[12] R. Zhang, S. Li, T. Fang, S. Zhu, and L. Quan, "Joint camera clustering and surface segmentation for large-scale multi-view stereo," in *ICCV*, 2015, pp. 2084–2092.

[13] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *IJCAI*, 2017, pp. 2843–2849.

[14] R. Kumar, T. Chen, M. Hardt, D. Beymer, K. Brannon, and T. F. Syeda-Mahmood, "Multiple kernel completion and its application to cardiac disease discrimination," in *ISBI*, 2013, pp. 764–767.

[15] S. Xiang, L. Yuan, W. Fan, Y. Wang, P. M. Thompson, and J. Ye, "Multi-source learning with block-wise missing data for alzheimer's disease prediction," in *ACM SIGKDD*, 2013, pp. 185–193.

[16] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *NIPS*, 1993, pp. 120–127.

[17] A. Trivedi, P. Rai, H. Daumé III, and S. L. DuVall, "Multiview clustering with incomplete views," in *NIPS 2010: Machine Learning for Social Computing Workshop, Whistler, Canada*, 2010.

[18] C. Xu, D. Tao, and C. Xu, "Multi-view learning with incomplete views," *IEEE Trans. Image Processing*, vol. 24, no. 12, pp. 5812–5825, 2015.

[19] W. Shao, L. He, and P. S. Yu, "Multiple incomplete views clustering via weighted nonnegative matrix factorization with $\ell_{2,1}$ regularization," in *ECML PKDD*, 2015, pp. 318–334.

[20] S. Bhadra, S. Kaski, and J. Rousu, "Multi-view kernel completion," in *arXiv:1602.02518*, 2016.

[21] Q. Yin, S. Wu, and L. Wang, "Incomplete multi-view clustering via subspace learning," in *ACM CIKM*, 2015, pp. 383–392.

[22] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel k-means with incomplete kernels," in *AAAI*, 2017, pp. 2259–2265.

[23] X. Zhu, X. Liu, M. Li, E. Zhu, L. Liu, Z. Cai, J. Yin, and W. Gao, "Localized incomplete multiple kernel k-means," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, 2018, pp. 3271–3277.

[24] X. Liu, X. Zhu, C. Tang, M. Li, E. Zhu, J. Yin, and W. Gao, "Efficient and effective incomplete multi-view clustering," in *AAAI*, 2019, pp. 1–8.

[25] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, 2019.

[26] T. Kato and R. Rivero, "Mutual kernel matrix completion," in *arXiv:1702.04077v2*, 2017.

[27] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[28] A. Maurer and M. Pontil, "k-dimensional coding schemes in Hilbert spaces," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5839–5846, 2010.

[29] T. Liu, D. Tao, and D. Xu, "Dimensionality-dependent generalization bounds for k-dimensional coding schemes," *Neural computation*, vol. 28, no. 10, pp. 2213–2249, 2016.

[30] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *NIPS*, 2014, pp. 1305–1313.

[31] L. Lovász and M. D. Plummer, *Matching Theory*. Akadémiai Kiadó, North Holland, 1986.

[32] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *CVPR*, vol. 2, 2006, pp. 1447–1454.

[33] T. Damoulas and M. A. Girolami, "Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection," *Bioinformatics*, vol. 24, no. 10, pp. 1264–1270, 2008.

[34] F. Yan, J. Kittler, K. Mikolajczyk, and M. A. Tahir, "Non-sparse multiple kernel fisher discriminant analysis," *Journal of Machine Learning Research*, vol. 13, pp. 607–642, 2012.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Assistant Researcher of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc.

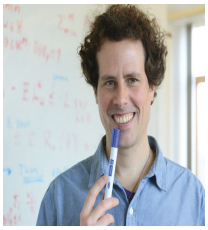
Miaomiao Li is in pursuit of her PhD degree at National University of Defense Technology, China. She is now Lecture of Changsha College, Changsha, China. Her current research interests include kernel learning and multi-view clustering. Miaomiao Li has published several peer-reviewed papers such as IEEE T-PAMI, IEEE T-NNLS, AAAI, IJCAI, Neurocomputing, etc. She serves on the Technical Program Committees of IJCAI 2017-2020.

Chang Tang received his Ph.D. degree from Tianjin University, Tianjin, China in 2016. He joined the AMRL Lab of the University of Wollongong between Sep. 2014 and Sep. 2015. He is currently an associate professor at the School of Computer Science, China University of Geosciences, Wuhan, China. His current research interests include machine learning and data mining. Dr. Tang served on the Technical Program Committees of IJCAI 2018 and ICME 2018.

Jingyuan Xia Jingyuan Xia received the B.Sc. and M.Sc. degree from the National University of Defense Technology, Hunan, China. He is currently working toward the Ph.D. degree with the Department of Electrical and Electronic Engineering, Imperial College London at London, UK. His current research interests include bi-linear convex optimization, low-rank matrix completion, sparse signal processing, intelligent transportation estimation.

Jian Xiong received the B.S. degree in engineering, and the M.S. and Ph.D. degrees in management from National University of Defense Technology, Changsha, China, in 2005, 2007, and 2012, respectively. He is an Associate Professor with the School of Business Administration, Southwestern University of Finance and Economics. His research interests include data mining, multiobjective evolutionary optimization, multiobjective decision making, project planning, and scheduling.

Li Liu received the BSc degree in communication engineering, the MSc degree in photogrammetry and remote sensing and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), China, in 2003, 2005 and 2012, respectively. She joined the faculty at NUDT in 2012, where she is currently an Associate Professor with the College of System Engineering. She was a cochair of seven International Workshops at CVPR, ICCV, and ECCV. She is going to lecture a tutorial at CVPR'19. She was a guest editor of special issues for IEEE TPAMI and IJCV. Her current research interests include facial behavior analysis, texture analysis, image classification, object detection and recognition. Her papers have currently over 1800 citations in Google Scholar. She currently serves as Associate Editor of the Visual Computer Journal.



Marius Kloft is a professor of computer science at TU Kaiserslautern and an adjunct faculty member of the University of Southern California. Previously he was a junior professor at HU Berlin and a joint postdoctoral fellow at the Courant Institute of Mathematical Sciences and Memorial Sloan-Kettering Cancer Center, New York. He earned his PhD at TU Berlin and UC Berkeley.

En Zhu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer Science, NUDT, China. His main research interests are pattern recognition, image processing, machine vision and machine learning. Dr. Zhu has published 60+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.