Multiple Kernel k-Means with Incomplete Kernels

Xinwang Liu, *Member, IEEE*, Xinzhong Zhu, Miaomiao Li, Lei Wang, *Senior Member, IEEE*, En Zhu, Tongliang Liu, Marius Kloft, Dinggang Shen, *Fellow, IEEE*, Jianping Yin, and Wen Gao, *Fellow, IEEE*

Abstract—Multiple kernel clustering (MKC) algorithms optimally combine a group of pre-specified base kernel matrices to improve clustering performance. However, existing MKC algorithms cannot efficiently address the situation where some rows and columns of base kernel matrices are absent. This paper proposes two simple yet effective algorithms to address this issue. Different from existing approaches where incomplete kernel matrices are first imputed and a standard MKC algorithm is applied to the imputed kernel matrices, our first algorithm integrates imputation and clustering into a unified learning procedure. Specifically, we perform multiple kernel clustering directly with the presence of incomplete kernel matrices, which are treated as auxiliary variables to be jointly optimized. Our algorithm does not require that there be at least one complete base kernel matrix over all the samples. Also, it adaptively imputes incomplete kernel matrices and combines them to best serve clustering. Moreover, we further improve this algorithm by encouraging these incomplete kernel matrices to mutually complete each other. The three-step iterative algorithm is designed to solve the resultant optimization problems. After that, we theoretically study the generalization bound of the proposed algorithms. Extensive experiments are conducted on 13 benchmark data sets to compare the proposed algorithms with existing imputation-based methods. Our algorithms consistently achieve superior performance and the improvement becomes more significant with increasing missing ratio, verifying the effectiveness and advantages of the proposed joint imputation and clustering.

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

1 Introduction

The recent years have seen many effort devoted to designing effective and efficient multiple kernel clustering (MKC) algorithms [1], [2], [3], [4], [5]. They aim to optimally combine a group of pre-specified base kernels to perform data clustering. For example, the work in [1] proposes to find the maximum margin hyperplane, the best cluster labeling, and the optimal kernel simultaneously. A novel optimized kernel k-means algorithm is presented in [2] to combine multiple data sources for clustering analysis. In [3], the kernel combination weights are allowed to adaptively change to capture the characteristics of individual samples. Replacing the squared error in k-means with an $\ell_{2,1}$-norm based one, the work in [4] develops a robust multiple kernel k-means (MKKM) algorithm that simultaneously finds the best clustering labels and the optimal combination of kernels. Observing that existing MKKM algorithms do not sufficiently consider the correlation among base kernels, the work in [5] designs a matrix-induced regularization to reduce the redundancy and enhance the diversity of the selected kernels. These MKC algorithms have been applied to various applications and demonstrated attractive clustering performance [6], [7], [8], [9], [10].

One underlying assumption commonly adopted by the above-mentioned MKC algorithms is that all of the base kernels are complete, i.e., none of the rows or columns of any base kernel shall be absent. In some practical applications such as Alzheimer's disease prediction [11] and cardiac disease discrimination [12], however, it is not uncommon to see that some views of a sample are missing, and this causes the corresponding rows and columns of related base kernels unfilled. The presence of incomplete base kernels makes it difficult to utilize the information of all views for clustering. A straightforward remedy may first impute incomplete kernels with a filling algorithm and then perform a standard MKC algorithm with the imputed kernels. Some widely used filling algorithms include zero-filling, mean value filling, k-nearest-neighbor filling and expectation-maximization (EM) filling [13]. Recently, more advanced imputation algorithms have been developed [14], [15], [16], [17]. The work in [14] constructs a full kernel matrix for an incomplete view

- *X. Liu, M. Li and E. Zhu are with the College of Computer, National University of Defense Technology, Changsha 410073, China. E-mail: {xinwangliu, enzhu}@nudt.edu.cn, miaomiaolinudt@gmail.com.*
- *X. Zhu is with the College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China, and also with the Research Institute of Ningbo Cixing Co. Ltd, Ningbo 315336, China. E-mail: zxz@zjnu.edu.cn.*
- *L. Wang is with the School of Computing and Information Technology, University of Wollongong, NSW 2522, Australia. E-mail: leiw@uow.edu.au.*
- *T. Liu is with the UBTECH Sydney Artificial Intelligence Centre, School of Information Technologies, Faculty of Engineering and Information Technologies, University of Sydney, J12 Cleveland St, Darlington NSW 2008, Australia. E-mail: tongliang.liu@sydney.edu.au.*
- *M. Kloft is with the Department of Computer Science, Technische Universität Kaiserslautern, Kaiserslautern 67653, Germany, and also with the Department of Computer Science, University of Southern California, California 90089. E-mail: kloft@usc.edu.*
- *D. Shen is with the Department of Radiology and BRIC, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599, and also with the Department of Brain and Cognitive Engineering, Korea University, Seoul 02841, Republic of Korea. E-mail: dgshen@med.unc.edu.*
- *J. Yin is with the Dongguan University of Technology, Guangdong 511700, China. E-mail: jpyin@dgut.edu.cn.*
- *W. Gao is with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China. E-mail: wgao@pku.edu.cn.*

Manuscript received 5 June 2017; revised 7 Nov. 2018; accepted 5 Jan. 2019. Date of publication 14 Jan. 2019; date of current version 1 Apr. 2020. (Corresponding author: Xinzhong Zhu.) Recommended for acceptance by P. V. Gehler. Digital Object Identifier no. 10.1109/TPAMI.2019.2892416

with the help of the other complete view (or equally, base kernel). By exploiting the connections of multiple views, the work in [15] proposes an algorithm to accomplish multi-view learning with incomplete views, where different views are assumed to be generated from a shared subspace. In [17], a multi-incomplete-view clustering algorithm is proposed. It learns latent feature matrices for all the views and generates a consensus matrix so that the difference between each view and the consensus is minimized. In addition, by modelling both within-view and between-view relationships among kernel values, an approach is proposed in [16] to predict missing rows and columns of a base kernel. Though demonstrating promising clustering performance in various applications, the above "two-stage" algorithms share a drawback that they disconnect the processes of imputation and clustering, and this prevents the two learning processes from negotiating with each other to achieve the optimal clustering. *Can we design a clustering-oriented imputation algorithm to enhance a kernel for clustering?*

To address this issue, we propose an absent multiple kernel k-means algorithm that integrates imputation and clustering into a single optimization procedure. In our algorithm, the clustering result at the last iteration guides the imputation of absent kernel elements, and the latter is in turn used to conduct the subsequent clustering. These two procedures are alternatively performed until convergence. By this way, the imputation and clustering processes are seamlessly connected, with the aim to achieve better clustering performance. Though being theoretically elegant, we also observe that this algorithm does not sufficiently consider that the imputation of each kernel could benefit from the other kernel matrices, even though they may be incomplete. As a result, we further improve the proposed multiple kernel k-means with incomplete kernels by explicitly allowing these incomplete kernel matrices to mutually impute each other. Both optimization objectives of the proposed absent multiple kernel clustering algorithms are carefully designed and two three-step alternative algorithms are developed to solve the resultant optimization problems, respectively. Extensive experimental study is carried out on 13 multiple kernel learning (MKL) benchmark data sets to evaluate the clustering performance of the proposed algorithm. As indicated, the proposed multiple kernel k-means algorithm with incomplete kernels (MKKM-IK) significantly outperforms existing two-stage imputation methods, and the improvement is particularly significant at high missing ratios, which is desirable. Meanwhile, we observe that the other proposed variant, i.e., MKKM-IK with mutual kernel completion (MKKM-IK-MKC), further improves the clustering performance of MKKM-IK. 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 or kernel matrices are encountered.

This work is a substantially extended version of our original conference paper [18]. Its significant improvement over the previous one can be summarized as follows: (1) We design a new algorithm, termed MKKM-IK-MKC, by incorporating the kernel reconstruction into existing MKKM-IK, and develop an iterative algorithm to efficiently solve the resultant optimization problem. Moreover, the newly proposed MKKM-IK-MKC significantly outperforms MKKM-IK proposed in the previous paper [18]. (2) We provide a theoretical explanation on why utilizing the same kernel coefficients in the kernel reconstruction and the combined kernel for clustering by revealing its connection with kernel alignment maximization. (3) We theoretically study the generalization bound of the proposed MKKM-IK and MKKM-IK-MKC on test data. (4) We design a toy data experiment to explore the sensitivity of the proposed MKKM-IK-MKC in the presence of noisy or uncorrelated kernels. (5) We conduct comprehensive experiments to validate the effectiveness of the proposed algorithms.

2 Related Work

2.1 Kernel k-Means Clustering (KKM)

Let $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ be a collection of n samples, and $\phi(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}$ be a feature mapping that maps $\mathbf{x}$ onto a reproducing kernel Hilbert space $\mathcal{H}$. The objective of kernel k-means clustering is to minimize the sum-of-squares loss over the cluster assignment matrix $\mathbf{Z} \in \{0,1\}^{n\times k}$, which can be formulated as the following optimization problem,

$$\begin{aligned} \min_{\mathbf{Z}\in\{0,1\}^{n\times k}} & \sum_{i=1,c=1}^{n,k} Z_{ic}\|\phi(\mathbf{x}_i) - \boldsymbol{\mu}_c\|_2^2 \\ s.t. & \sum_{c=1}^{k} Z_{ic} = 1, \end{aligned} \tag{1}$$

where $n_c = \sum_{i=1}^n Z_{ic}$ and $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^n Z_{ic}\phi(\mathbf{x}_i)$ are the size and centroid of the cth cluster.

The optimization problem in Eq. (1) can be rewritten as the following matrix-vector form,

$$\min_{\mathbf{Z}\in\{0,1\}^{n\times k}} \ \mathrm{Tr}(\mathbf{K}) - \mathrm{Tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{Z}^\top\mathbf{K}\mathbf{Z}\mathbf{L}^{\frac{1}{2}}) \ \ s.t. \ \ \mathbf{Z}\mathbf{1}_k = \mathbf{1}_n, \tag{2}$$

where $\mathbf{K}$ is a kernel matrix with $K_{ij} = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$, $\mathbf{L} = \mathrm{diag}([n_1^{-1}, n_2^{-1}, \ldots, n_k^{-1}])$ and $\mathbf{1}_\ell \in \mathbb{R}^\ell$ is a column vector with all elements being 1.

The variable $\mathbf{Z}$ in Eq. (2) is discrete, and this makes the optimization problem difficult to solve. A common approach is to relax $\mathbf{Z}$ to take real values. Specifically, by defining $\mathbf{H} = \mathbf{Z}\mathbf{L}^{\frac{1}{2}}$ and letting $\mathbf{H}$ take real values, a relaxed version of the above problem can be obtained as

$$\min_{\mathbf{H}} \ \mathrm{Tr}\big(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big) \ \ s.t. \ \ \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{3}$$

where $\mathbf{I}_k$ is an identity matrix with size $k \times k$. The optimal $\mathbf{H}$ for Eq. (3) can be obtained by taking the eigenvectors corresponding to the top k largest eigenvalues of $\mathbf{K}$ [19].

2.2 Multiple Kernel k-Means Clustering (MKKM)

In a multiple kernel setting, each sample $\mathbf{x}$ has multiple feature representations defined by $\{\mathbf{x}^{(p)}\}_{p=1}^m$. Each sample is represented as $\phi_{\boldsymbol{\beta}}(\mathbf{x}) = [\beta_1\phi_1(\mathbf{x}^{(1)})^\top, \ldots, \beta_m\phi_m(\mathbf{x}^{(m)})^\top]^\top$, where $\{\phi_p(\cdot)\}_{p=1}^m$ is a group of feature mappings and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_m]^\top$ consists of the coefficients of the m base kernels. These coefficients will be optimized during learning. Based on the definition of $\phi_{\boldsymbol{\beta}}(\mathbf{x})$, a kernel function can be expressed as

$$\kappa_{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\beta}}(\mathbf{x}_i)^\top\phi_{\boldsymbol{\beta}}(\mathbf{x}_j) = \sum_{p=1}^m \beta_p^2\kappa_p(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)}). \tag{4}$$

By replacing the kernel matrix $\mathbf{K}$ in Eq. (3) with $\mathbf{K}_{\boldsymbol{\beta}}$ computed via Eq. (4), the objective of MKKM can be written as

$$\begin{aligned} \min_{\mathbf{H},\boldsymbol{\beta}} \ & \mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k,\ \boldsymbol{\beta}^\top\mathbf{1}_m = 1,\ \beta_p \geq 0,\ \forall p. \end{aligned} \tag{5}$$

This problem can be solved by alternatively updating $\mathbf{H}$ and $\boldsymbol{\beta}$: i) *Optimizing* $\mathbf{H}$ *given* $\boldsymbol{\beta}$. With the kernel coefficients $\boldsymbol{\beta}$ fixed, $\mathbf{H}$ can be obtained by solving a kernel k-means clustering optimization problem shown in Eq. (3); ii) *Optimizing* $\boldsymbol{\beta}$ *given* $\mathbf{H}$. With $\mathbf{H}$ fixed, $\boldsymbol{\beta}$ can be optimized via solving the following quadratic programming with linear constraints,

$$\begin{aligned} \min_{\boldsymbol{\beta}} \ & \sum_{p=1}^{m} \beta_p^2 \mathrm{Tr}\big(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big) \\ s.t. \ & \boldsymbol{\beta}^\top\mathbf{1}_m = 1,\ \beta_p \geq 0,\ \forall p. \end{aligned} \tag{6}$$

As noted in [2], [3], using a convex combination of kernels $\sum_{p=1}^{m}\beta_p\mathbf{K}_p$ to replace $\mathbf{K}_{\boldsymbol{\beta}}$ in Eq. (5) is not a viable option, because this could make only one single kernel be activated and all the others assigned with zero weights. Other recent work using ℓ_2-norm combination can be found in [20], [21].

3 The Proposed Algorithms

3.1 Formulation of Multiple Kernel K-Means with Incomplete Kernels

Let $\mathbf{s}_p\,(1 \leq p \leq m)$ denote the sample indices for which the pth view is present and $\mathbf{K}_p^{(cc)}$ be used to denote the kernel sub-matrix computed with these samples. Note that this setting is consistent with the literature, and it is even more general since it does not require that there be at least one complete view across all the samples, as assumed in [14].

The absence of rows and columns from base kernels makes clustering challenging. Existing two-stage approaches first impute these base kernels and then apply a conventional clustering algorithm to them. We have the following two arguments. First, although such imputation is sound from the perspective of "general-purpose", it may not be an optimal option when it has been known that the imputed kernels are used for a clustering task. This is because for most, if not all, practical tasks a belief holds that these employed base kernels or views (when in their complete form) shall, more or less, be able to serve the clustering task. However, such a belief was not exploited by these two-stage approaches as prior knowledge to guide the imputation process. Second, from the perspective that the ultimate goal is to appropriately cluster data, we shall try to directly pursue the clustering result, by treating the absent kernel entries as auxiliary unknowns during this course. In other words, imputed kernels could be merely viewed as the by-products of clustering.

These two arguments motivate us to seek a more natural and reasonable manner to deal with the absence in multiple kernel clustering. That is to perform imputation and clustering in a joint way: 1) impute the absent kernels under the guidance of clustering; and 2) update the clustering with the imputed kernels. 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*. In specific, we propose the multiple kernel k-means algorithm with incomplete kernels as follows,

$$\begin{aligned} \min_{\mathbf{H},\,\boldsymbol{\beta},\,\{\mathbf{K}_p\}_{p=1}^m} \ & \mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k,\ \boldsymbol{\beta}^\top\mathbf{1}_m = 1,\ \beta_p \geq 0, \\ & \mathbf{K}_p(\mathbf{s}_p, \mathbf{s}_p) = \mathbf{K}_p^{(cc)},\ \mathbf{K}_p \succeq 0,\ \forall p, \\ & \mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^{m} \beta_p^2\mathbf{K}_p. \end{aligned} \tag{7}$$

The only difference between the objective function in Eq. (7) and that of traditional MKKM in Eq. (5) lies at the incorporation of optimizing $\{\mathbf{K}_p\}_{p=1}^m$. Note that 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. Though the model in Eq. (7) is simple, it admits the following advantages: 1) Our objective function is more direct and well targets the ultimate goal, i.e., clustering, by integrating kernel completion and clustering into one unified learning framework, where the kernel imputation is treated as a by-product; 2) Our algorithm works in a MKL scenario [22], which is able to naturally deal with a large number of base kernels and adaptively combine them for clustering; 3) Our algorithm does not require any base kernel to be completely observed, which is however necessary for some of the existing imputation algorithms such as [14]. Besides, our algorithm is parameter-free once the number of clusters to form is specified. In [18], a three-step iterative algorithm with proved convergence is designed to solve the optimization problem in Eq. (7). Interested readers can refer to [18] for the detail.

3.2 Incomplete MKKM with Mutual Kernel Completion (MKKM-IK-MKC)

3.2.1 Formulation of Incomplete MKKM with Mutual Kernel Completion

The proposed MKKM-IK in Section 3.1 which jointly performs kernel completion and clustering is effective, and achieves promising clustering performance as shown in the experimental part. However, this algorithm imputes each incomplete kernel by only utilizing the clustering result $\mathbf{H}$, while not sufficiently considering that the available information from other kernels could also contribute to its completion. Meanwhile, the optimization of $\boldsymbol{\beta}$ in Eq. (7) is inherited from existing MKKM framework, which could result in selecting mutually redundant kernels and affect the diversity of information sources utilized for clustering [5]. Both factors could adversely affect the clustering performance.

To address the above issues, we aim to further improve the proposed MKKM-IK by encouraging the incomplete kernel matrices to mutually complete each other. Besides utilizing the clustering result $\mathbf{H}$ to fill each incomplete kernel matrix, the improved algorithm proposes to impute each incomplete kernel matrix by utilizing other incomplete kernel matrices. To this end, we assume that each kernel $\mathbf{K}_p$ resides in the neighborhood of a linear combination of other kernels, i.e., $\sum_{q=1,q\neq p}^{m}\beta_q\mathbf{K}_q$, and minimize $\|\mathbf{K}_p - \sum_{q=1,q\neq p}^{m}\beta_q\mathbf{K}_q\|_{\mathrm{F}}$ to guide the completion of each kernel. It is worth pointing out that the kernel coefficients in this reconstruction term and in the combined kernel for clustering are the same. By doing so, the

reconstruction term naturally induces a regularization on $\boldsymbol{\beta}$ which takes the correlation of base kernels into consideration. Specifically, with given $\{\mathbf{K}_p\}_{p=1}^m$, the optimization w.r.t $\boldsymbol{\beta}$ is equivalent to

$$\min_{\boldsymbol{\beta}} \ \frac{1}{2}\boldsymbol{\beta}^\top \mathbf{A}\boldsymbol{\beta} - \mathbf{f}^\top \boldsymbol{\beta}, \ \ s.t. \ \ \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \geq 0, \forall p, \tag{8}$$

where $\mathbf{M} \in \mathbb{R}^{m\times m}$ with elements $M_{pq} = \mathrm{Tr}(\mathbf{K}_p\mathbf{K}_q)$ to measure the correlation between each pair of kernel matrices $\mathbf{K}_p$ and $\mathbf{K}_q$, $\mathbf{A} = \mathbf{C} \odot \mathbf{M}$ and $\mathbf{f} = \mathbf{M1} - \mathrm{diag}(\mathbf{M})$, $\mathbf{C}$ is a matrix with all elements $m-2$ and diagonal elements $m-1$, $\mathbf{1} \in \mathbb{R}^m$ is column vector with all elements one, and $\mathrm{diag}(\mathbf{M})$ denotes the diagonal elements of $\mathbf{M}$.

Eq. (8) can be treated as a regularization on the kernel combination weights for clustering:

- Its first term, i.e., $\boldsymbol{\beta}^\top \mathbf{A}\boldsymbol{\beta}$ is helpful to reduce the redundancy and enforce the diversity of the selected kernels. A larger $\mathbf{M}_{pq}$ means high correlation between $\mathbf{K}_p$ and $\mathbf{K}_q$, and a smaller one implies that their correlation is low. By minimizing this term, the risk of simultaneously assigning β_p and β_q with large weights can be greatly reduced if $\mathbf{K}_p$ and $\mathbf{K}_q$ are highly correlated. Meanwhile, this regularization increases the probability of jointly assigning β_p and β_q with larger weights as long as $\mathbf{K}_p$ and $\mathbf{K}_q$ are less correlated. As a consequence, this criterion is beneficial to promoting the diversity of selected kernels, and makes the pre-specified kernels more effectively utilized, leading to improved clustering performance. In fact, the theoretical implication of incorporating this regularization can be well justified from the perspective of the following commonly used kernel alignment criterion [5]

$$\max_{\boldsymbol{\beta},\mathbf{H}} \frac{\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\beta}}\big(\mathbf{H}\mathbf{H}^\top\big)\big)}{\big\|\mathbf{H}\mathbf{H}^\top\big\|_{\mathrm{F}}\big\|\mathbf{K}_{\boldsymbol{\beta}}\big\|_{\mathrm{F}}} \ s.t. \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \ \ \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \tag{9}$$

where $\mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^m \beta_p^2\mathbf{K}_p$ and $\|\mathbf{X}\|_{\mathrm{F}} = \sqrt{\mathrm{Tr}\big(\mathbf{X}^\top\mathbf{X}\big)}$.

Eq. (9) is equivalent to

$$\max_{\boldsymbol{\beta},\mathbf{H}} \frac{\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\beta}}\big(\mathbf{H}\mathbf{H}^\top\big)\big)}{\sqrt{\hat{\boldsymbol{\beta}}^\top\mathbf{M}\hat{\boldsymbol{\beta}}}} \ s.t. \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \ \ \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \tag{10}$$

where $\hat{\boldsymbol{\beta}} = [\beta_1^2, \ldots, \beta_m^2]^\top$.

The optimization in Eq. (10) is readily understood. By looking into the numerator and denominator of Eq. (10) in depth, we observe that: i) The negative of the numerator of kernel alignment, i.e., $-\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}\mathbf{H}\mathbf{H}^\top)$, is conceptually equivalent to the objective of MKKM, i.e., $\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big)$; and ii) The denominator, i.e., $\hat{\boldsymbol{\beta}}^\top\mathbf{M}\hat{\boldsymbol{\beta}}$, is a regularization on the kernel coefficients to prevent β_p and β_q from being jointly assigned to a large weight if M_{pq} is relatively high. From the perspective of regularization, the effect of $\boldsymbol{\beta}^\top\mathbf{M}\boldsymbol{\beta}$ and $\hat{\boldsymbol{\beta}}^\top\mathbf{M}\hat{\boldsymbol{\beta}}$ could be treated as the same. Therefore, by using the same kernel coefficients in the regularization term and in the combined kernel for clustering, it is helpful to reduce the redundancy and enforce the diversity of the selected kernels for clustering.
- Its second term, i.e., $-\mathbf{f}^\top\boldsymbol{\beta}$, is helpful to reduce the kernel weights of noisy or irrelevant kernels if there are any such kernels. Note that our objective is to maximize $\mathbf{f}^\top\boldsymbol{\beta}$ with $\mathbf{f} = \mathbf{M1} - \mathrm{diag}(\mathbf{M})$. If $\mathbf{K}_p$ is a noisy or irrelavant kernel, its correlation with other kernels will be low, leading to a small f_p with $\mathbf{f} = [f_1, \ldots, f_m]^\top$. In this case, maximizing $\mathbf{f}^\top\boldsymbol{\beta}$ with ℓ_1-norm constraint would lead to small β_p, as shown in Fig. 7. Consequently, by using the same kernel coefficients in the regularization term and in the combined kernel for clustering, it is helpful to reduce the weights of irrelevant kernels for clustering.

According to the aforementioned analysis, we conclude that the kernel construction term of the proposed MKKM-IK-MKC naturally induces a regularization term on kernel coefficients for clustering, which is helpful to better utilize the pre-specified kernel matrices, leading to significantly improved clustering performance.

By integrating the above mutual kernel completion term into the objective of MKKM-IK in Eq. (7), we obtain the objective function of the proposed algorithm as follows:

$$\begin{aligned}\min_{\mathbf{H},\boldsymbol{\beta},\{\mathbf{K}_p\}_{p=1}^m} \ & \mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big) + \frac{\lambda}{2}\sum_{p=1}^m \Big\|\mathbf{K}_p - \sum_{\substack{q=1\\q\neq p}}^m \beta_q\mathbf{K}_q\Big\|_{\mathrm{F}}^2 \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \ \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \ \beta_p \geq 0, \forall p \\ & \mathbf{K}_p(\mathbf{s}_p, \mathbf{s}_p) = \mathbf{K}_p^{(cc)}, \ \mathbf{K}_p \succeq 0, \forall p, \\ & \mathbf{K}_{\boldsymbol{\beta}} = \sum_{p=1}^m \beta_p^2\mathbf{K}_p,\end{aligned} \tag{11}$$

where λ is a regularization parameter to trade-off the MKKM clustering and mutual kernel completion.

Incorporating the regularization term makes the optimization problem more challenging. In the following, we design a three-step alternative algorithm to solve the optimization problem in Eq. (11).

3.2.2 Alternative Optimization of MKKM-IK-MKC

We design a three-step alternative optimization algorithm to solve the problem in Eq. (11):

i) *Optimizing* $\mathbf{H}$ *with fixed* $\boldsymbol{\beta}$ *and* $\{\mathbf{K}_p\}_{p=1}^m$. Given $\boldsymbol{\beta}$ and $\{\mathbf{K}_p\}_{p=1}^m$, the optimization in Eq. (11) w.r.t. $\mathbf{H}$ reduces to a conventional kernel k-means problem, which can be efficiently solved by existing packages.

ii) *Optimizing* $\{\mathbf{K}_p\}_{p=1}^m$ *with fixed* $\boldsymbol{\beta}$ *and* $\mathbf{H}$. We adopt a coordinate descend manner to optimize each $\mathbf{K}_p$. Specifically, all kernel matrices $\{\mathbf{K}_q\}_{q=1,q\neq p}^m$ are kept as constant during optimizing $\mathbf{K}_p$. Given $\boldsymbol{\beta}$ and $\mathbf{H}$, the optimization in Eq. (11) w.r.t. each $\mathbf{K}_p$ is equivalent to the following optimization problem,

$$\min_{\mathbf{K}_p} \ \frac{1}{2}\big\|\mathbf{K}_p - \mathbf{T}\big\|_{\mathrm{F}}^2 \ \ s.t. \ \ \mathbf{K}_p(\mathbf{s}_p, \mathbf{s}_p) = \mathbf{K}_p^{(cc)}, \ \mathbf{K}_p \succeq 0, \tag{12}$$

where $\mathbf{T} = \sum_{\substack{q=1\\q\neq p}}^m \frac{\beta_p+\beta_q-(m-2)\beta_p\beta_q}{1+(m-1)\beta_p^2}\mathbf{K}_q - \frac{\beta_p^2(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)}{\lambda(1+(m-1)\beta_p^2)}$. As seen, the completion of each $\mathbf{K}_p$ is now dependent on both the clustering result $\mathbf{H}$ and combination of

the other kernels. See the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2019.2892416 for the detailed derivation.

Note that the optimization in Eq. (12) itself is a semi-definite programming (SDP), which can be readily solved by existing convex optimization toolbox such as CVX [23]. However, the high time complexity of SDP problems prevents it from being applied to medium or large scale applications. To relieve the intensive computational burden, we propose to approximately optimize $\mathbf{K}_p$ as follows,

$$\min_{\mathbf{K}_p} \ \left\|\mathbf{K}_p - \mathbf{T}\right\|_{\mathrm{F}}^2 \ s.t. \ \ \mathbf{K}_p(\mathbf{s}_p, \mathbf{s}_p) = \mathbf{K}_p^{(cc)}. \tag{13}$$

The optimal solution in Eq. (13), denoted as $\hat{\mathbf{K}}_p$, can be readily obtained by filling the missing elements of $\mathbf{K}_p$ with the corresponding ones of $\mathbf{T}$. After obtaining the solution of Eq. (13), we project it into the space of positive semi-defined (PSD) matrices by performing an eigen-decomposition to make $\mathbf{K}_p$ satisfy $\mathbf{K}_p \succeq 0$. Specifically, let us denote $\hat{\mathbf{K}}_p = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ as the eigen-decomposition of $\hat{\mathbf{K}}_p$. Then, the optimal PSD approximation of $\hat{\mathbf{K}}_p$ is $\mathbf{U}\mathbf{\Lambda}^+\mathbf{U}^\top$, where $\mathbf{\Lambda}^+$ is obtained by setting the negative diagonal elements of $\mathbf{\Lambda}^+$ as zero. This technique is widely applied in optimization with PSD constraints and usually produces excellent results. The detailed derivation of optimizing $\{\mathbf{K}_p\}_{p=1}^m$ can be found in the appendix, available in the online supplemental material.

iii) *Optimizing $\boldsymbol{\beta}$ with fixed* $\mathbf{H}$ *and* $\{\mathbf{K}_p\}_{p=1}^m$. Given $\mathbf{H}$ and $\{\mathbf{K}_p\}_{p=1}^m$, the optimization in Eq. (11) w.r.t. $\boldsymbol{\beta}$ is the following quadratic programming with linear constraints,

$$\begin{aligned} \min_{\boldsymbol{\beta}} \ & \frac{1}{2}\boldsymbol{\beta}^\top \left((\mathbf{A} \odot \mathbf{M}) + \frac{2}{\lambda}\mathrm{diag}(\mathbf{d}) \right) \boldsymbol{\beta} - \mathbf{f}^\top \boldsymbol{\beta} \\ s.t. \ & \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \geq 0, \forall p, \end{aligned} \tag{14}$$

where $\mathbf{d} = [d_1, \ldots, d_m]^\top$ is a column vector with $d_p = \mathrm{Tr}\big(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big)$, $\mathbf{A} \in \mathbb{R}^{m\times m}$ with all entries $m-2$ and diagonal ones $m-1$, $\mathbf{M} \in \mathbb{R}^{m\times m}$ measures the mutual correlation of each pairwise kernel via $M_{pq} = \mathrm{Tr}\big(\mathbf{K}_p\mathbf{K}_q\big)$, $\mathbf{f} = \mathbf{M}\mathbf{1}_m - \mathrm{diag}(\mathbf{M})$ and $\mathbf{1}_m$ is a m-dimension column vector with all elements one. As seen from Eq. (14), the correlation among base kernels has been incorporated via $\mathbf{M}$, which is helpful to reduce the redundancy and enhance the diversity of selected kernels [5], leading to improved clustering performance. The detailed derivation of optimizing $\boldsymbol{\beta}$ can be found in the appendix, available in the online supplemental material.

In sum, our algorithm for solving Eq. (11) is outlined in Algorithm 1. The computational complexity for the proposed MKKM-IK-MKC is $\mathcal{O}(n^3 + mn^3 + m^3)$ per iteration, where n and m are the total number of whole samples and base kernels, respectively. It is worth pointing out that $\mathbf{K}_p$ can be calculated in parallel since each of them are independent. By this way, our algorithm shall scale well to the number of kernels.

4 Generalization Analysis of the Proposed Algorithms

Generalization error for k-means clustering has been studied by fixing the centroids obtained in the training process and generalizing them for testing; see, e.g., [24], [25]. In this section, we study how the centroids obtained by the proposed MKKM-IK and MKKM-IK-MKC generalize onto test data by deriving generalization bounds via exploiting the reconstruction error.

Algorithm 1. The Proposed MKKM-IK-MKC

Input: $\{\mathbf{K}_p^{(cc)}\}_{p=1}^m$, $\{\mathbf{s}_p\}_{p=1}^m$, λ and ϵ_0.
Output: $\mathbf{H}$, $\boldsymbol{\beta}$ and $\{\mathbf{K}_p\}_{p=1}^m$.
Initialize $\boldsymbol{\beta}^{(0)} = \mathbf{1}_m/m$, $\{\mathbf{K}_p^{(0)}\}_{p=1}^m$ and $t = 1$.
repeat
  $\mathbf{K}_{\boldsymbol{\beta}}^{(t)} = \sum_{p=1}^m \left(\beta_p^{(t-1)}\right)^2 \mathbf{K}_p^{(t-1)}$.
  Update $\mathbf{H}^{(t)}$ by solving kernel k-means with given $\mathbf{K}_{\boldsymbol{\beta}}^{(t)}$.
  Update each $\mathbf{K}_p^{(t)}$ with $\mathbf{H}^{(t)}$ and $\{\mathbf{K}_q^{(t-1)}\}_{q=1,q\neq p}^m$ by Eq. (12).
  Update $\boldsymbol{\beta}^{(t)}$ by solving Eq. (14) with given $\mathbf{H}^{(t)}$ and $\{\mathbf{K}_p^{(t)}\}_{p=1}^m$.
  $t = t + 1$.
until $\max\{|\beta_1^{(t-1)} - \beta_1^{(t)}|, \ldots, |\beta_m^{(t-1)} - \beta_m^{(t)}|\} \leq \epsilon_0$

Before defining the reconstruction error of k-means, we need to model the absence of views. Specifically, let the indicator function $t(\mathbf{x}^{(p)})$ denote the absence of the pth view of the observation $\mathbf{x}$, i.e., if the pth 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 $\hat{\mathbf{C}} = [\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_k]$ be the learned matrix composed of the k centroids and $\hat{\boldsymbol{\beta}}$ the learned kernel weights by the proposed MKKM-IK and MKKM-IK-MKC. Effective k-means clustering algorithms should have the following reconstruction error small

$$\mathbb{E}\left[\min_{\mathbf{y}\in\{\mathbf{e}_1,\ldots,\mathbf{e}_k\}} \left\|\boldsymbol{\phi}_{\hat{\boldsymbol{\beta}}}(\mathbf{x}) - \hat{\mathbf{C}}\mathbf{y}\right\|_{\mathcal{H}}^2\right], \tag{15}$$

where $\boldsymbol{\phi}_{\hat{\boldsymbol{\beta}}}(\mathbf{x}) = [\hat{\beta}_1 t(\mathbf{x}^{(1)})\boldsymbol{\phi}_1^\top(\mathbf{x}^{(1)}), \ldots, \hat{\beta}_m t(\mathbf{x}^{(m)})\boldsymbol{\phi}_m^\top(\mathbf{x}^{(m)})]^\top$, $\mathbf{e}_1, \ldots, \mathbf{e}_k$ form the orthogonal bases of $\mathbb{R}^k$. We show how the proposed algorithms achieve this goal.

Let us define a function class first:

$$\begin{aligned} \mathcal{F} = \Big\{ & f : \mathbf{x} \mapsto \min_{\mathbf{y}\in\{\mathbf{e}_1,\ldots,\mathbf{e}_k\}} \left\|\boldsymbol{\phi}_{\boldsymbol{\beta}}(\mathbf{x}) - \mathbf{C}\mathbf{y}\right\|_{\mathcal{H}}^2 \Big| \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \ \beta_p \geq 0, \\ & \mathbf{C} \in \mathcal{H}^k, \ t(\mathbf{x}_i^{(p)})t(\mathbf{x}_j^{(p)})\boldsymbol{\phi}_p^\top(\mathbf{x}_i^{(p)})\boldsymbol{\phi}_p^\top(\mathbf{x}_j^{(p)}) \leq b, \ \forall p, \forall \mathbf{x}_i \in \mathcal{X} \Big\}, \end{aligned} \tag{16}$$

where $\mathcal{H}^k$ stands for the multiple kernel Hilbert space.

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

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})] \leq & \frac{1}{n}\sum_{i=1}^n f(\mathbf{x}_i) + \frac{4\sqrt{\pi}mb\mathcal{G}_{1n}(\boldsymbol{\beta}, t)}{n} + \frac{4\sqrt{\pi m}b\mathcal{G}_{2n}(\boldsymbol{\beta}, t)}{n} \\ & + \frac{\sqrt{8\pi}bk^2}{\sqrt{n}} + 4b\sqrt{\frac{\log 1/\delta}{2n}}, \end{aligned} \tag{17}$$

where

$$\mathcal{G}_{1n}(\boldsymbol{\beta},t) \triangleq \mathbb{E}_{\gamma}\left[\sup_{\boldsymbol{\beta},t}\sum_{i=1}^{n}\sum_{p,q=1}^{m}\gamma_{ipq} < \beta_p t(\mathbf{x}_i^{(p)}), \beta_q t(\mathbf{x}_i^{(q)}) >\right], \tag{18}$$

$$\mathcal{G}_{2n}(\boldsymbol{\beta},t) = \mathbb{E}_{\gamma}\left[\sup_{\boldsymbol{\beta},t}\sum_{i=1}^{n}\sum_{c=1}^{k}\sum_{p=1}^{m}\gamma_{icp}\beta_p t(\mathbf{x}_i^{(p)})\right], \tag{19}$$

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

Note that if all the views are accessible, we have $\mathcal{G}_{1n}(\boldsymbol{\beta},t) \leq m^2\sqrt{n}$ and $\mathcal{G}_{2n}(\boldsymbol{\beta},t) \leq mk\sqrt{n}$. 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 $\mathcal{G}_{1n}(\boldsymbol{\beta},t)$ and $\mathcal{G}_{2n}(\boldsymbol{\beta},t)$ will become lager, making it more difficult to learn and more training examples are required to secure a given clustering accuracy.

According to Theorem 1, for any learned $\hat{\boldsymbol{\beta}}, \hat{\mathbf{C}}$, to achieve a small

$$\mathbb{E}[f(\mathbf{x})] = \mathbb{E}\left[\min_{\mathbf{y}\in\{e_1,\ldots,e_k\}}\left\|\boldsymbol{\phi}_{\hat{\boldsymbol{\beta}}}(\mathbf{x}) - \hat{\mathbf{C}}\mathbf{y}\right\|_{\mathcal{H}}^2\right],$$

the corresponding $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$ needs to be as small as possible. Assume that $\boldsymbol{\beta}$ and $\mathbf{C}$ are obtained by minimizing $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$ and that $\mathbf{H}$ is constructed according to Eq. (3), we have

$$\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i) \leq \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^{\top})), \tag{20}$$

because the proposed algorithms pose a constraint $\mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_k$ which will make the corresponding centroids non-optimal for minimizing $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$. This means that the proposed objectives are upper bounds of $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$. Thus, minimizing $\mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^{\top}))$ will ensure a small $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$ for good generalization, which also verifies the good generalization ability of the proposed algorithms. The detailed proof are provided in the supplemental material, available online.

5 Experimental Result

5.1 Experimental Settings

The proposed algorithm is experimentally evaluated on 13 widely used MKL benchmark data sets shown in Table 2. They are Cornell, Texas, Washington and Wisconsin[1], Oxford Flower17 and Flower102[2], Columbia Consumer Video (CCV)[3] and Caltech101[4]. The original features for the first four data sets are available. For each of these datasets, we obtain two kernel matrices by applying a linear kernel to the features of each view. For CCV, we generate three base kernels by applying a Gaussian kernel on its SIFT, STIP and MFCC features, where the widths of the three Gaussian kernels are set as the mean of all pairwise sample distances, respectively. For Flower17, Flower102 and Caltech101 data sets, all kernel matrices are pre-computed and can be publicly downloaded from the above websites. Meanwhile, Caltech101-5 means the number of samples belonging to each cluster is 5, and so on.

We compare the proposed algorithms 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 [14] and partial multi-view clustering (PVC) [7]. The algorithms in [15], [17], [26] are not incorporated into our experimental comparison since they only consider the absence of input features while not the rows/columns of base kernels. Compared with [16], the imputation algorithm in [14] is much simpler and more computationally efficient. Therefore, we choose [14] as a representative algorithm to demonstrate the advantages and effectiveness of joint optimization on imputation and clustering. The widely used MKKM [3] is applied with these imputed base kernels. These two-stage methods are termed MKKM+ZF, MKKM+MF, MKKM+KNN and MKKM+AF in this experiment, respectively. We do not include the EM-based imputation algorithm due to its high computational cost, even for small-sized samples. The Matlab codes of kernel k-means and MKKM are publicly downloaded from `https://github.com/mehmetgonen/lmkkmeans`. Additionally, we also provide the results of the proposed MKKM-IK with three different initializations for comprehensive comparison, including MKKM-IK+ZF, MKKM-IK+MF and MKKM-IK+KNN. Meanwhile, the proposed MKKM-IK with mutual kernel completion, is termed MKKM-IK-MKC in comparison.

Following the literature [27], all base kernels are centered and scaled so that we have $\kappa_p(\mathbf{x}_i, \mathbf{x}_i) = 1$ for all i and p. For all data sets, it is assumed that the true number of clusters is known and it is set as the true number of classes. To generate incomplete kernels, we create the index vectors $\{\mathbf{s}_p\}_{p=1}^{m}$ as follows. We first randomly select $\mathrm{round}(\varepsilon * n)$ samples, where $\mathrm{round}(\cdot)$ denotes a rounding function. For each selected sample, a random vector $\mathbf{v} = (v_1,\ldots,v_m) \in [0,1]^m$ and a scalar v_0 ($v_0 \in [0,1]$) are then generated, respectively. The pth view will be present for this sample if $v_p \geq v_0$ is satisfied. In case none of $v_1,\ldots,v_m$ can satisfy this condition, we will generate a new $\mathbf{v}$ to ensure that at least one view is available for a sample. Note that this does not mean that we require a complete view across all the samples. After the above step, we will be able to obtain the index vector $\mathbf{s}_p$ listing the samples whose pth view is present. 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. Intuitively, the larger the value of ε is, the poorer the clustering performance that an algorithm can achieve. In order to show this point in depth, we compare these algorithms with respect to ε. Specifically, ε on all the data sets is set as $[0.1 : 0.1 : 0.9]$.

The widely used clustering accuracy (ACC), normalized mutual information (NMI) and purity 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

1. `http://lamda.nju.edu.cn/code_PVC.ashx`
2. `http://www.robots.ox.ac.uk/~vgg/data/flowers/`
3. `http://www.ee.columbia.edu/ln/dvmm/CCV/`
4. `http://files.is.tue.mpg.de/pgehler/projects/iccv09/`

TABLE 1
Aggregated ACC and NMI Comparison (mean±std) of Different Clustering Algorithms on Cornell, Texas, Washington and Wisconsin Data Sets

Datasets	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF [14]	PVC [7]	MKKM-IK (proposed)			
						ZF	MF	KNN	MKC
ACC									
Cornell	33.47 ± 1.03	33.05 ± 0.81	33.50 ± 1.11	35.84 ± 1.25	35.71 ± 1.21	36.66 ± 1.32	36.86 ± 1.24	36.33 ± 1.36	$\mathbf{47.50 \pm 1.21}$
Texas	35.84 ± 0.71	37.12 ± 1.11	34.67 ± 0.80	37.39 ± 0.99	38.69 ± 1.36	37.83 ± 0.88	38.55 ± 0.82	37.36 ± 0.85	$\mathbf{43.48 \pm 0.93}$
Washington	46.36 ± 1.08	43.66 ± 0.96	45.39 ± 1.13	47.12 ± 1.07	42.65 ± 0.94	46.71 ± 1.01	46.47 ± 1.06	46.37 ± 0.94	$\mathbf{49.69 \pm 0.81}$
Wisconsin	45.75 ± 1.06	43.93 ± 1.13	46.70 ± 0.93	45.75 ± 0.91	34.45 ± 0.86	44.89 ± 1.06	43.52 ± 1.03	44.47 ± 1.13	$\mathbf{49.99 \pm 0.88}$
NMI									
Cornell	9.96 ± 0.57	9.34 ± 0.54	10.18 ± 0.83	12.57 ± 0.89	5.58 ± 0.66	13.25 ± 0.85	13.31 ± 0.93	12.92 ± 0.97	$\mathbf{25.84 \pm 1.19}$
Texas	9.87 ± 0.57	8.15 ± 0.62	9.22 ± 0.57	12.02 ± 0.78	3.42 ± 0.46	12.64 ± 0.81	12.38 ± 0.71	12.16 ± 0.63	$\mathbf{20.81 \pm 0.95}$
Washington	23.23 ± 1.03	22.49 ± 0.96	22.24 ± 1.17	23.36 ± 0.98	11.41 ± 0.60	22.62 ± 0.99	22.60 ± 0.79	22.42 ± 0.94	$\mathbf{25.85 \pm 0.81}$
Wisconsin	20.06 ± 0.79	20.12 ± 1.03	21.22 ± 0.75	19.88 ± 0.76	3.05 ± 0.30	19.21 ± 0.97	19.17 ± 0.93	19.05 ± 0.87	$\mathbf{23.81 \pm 0.82}$

ground-truth label, respectively. Let $\mathbf{c} = [c_1, \ldots, c_n]^\top$ and $\mathbf{y} = [y_1, \ldots, y_n]^\top$ denote the predicted cluster labels of a clustering algorithm and the provided ground-truth labels of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, respectively. The clustering accuracy (ACC) is defined as follows,

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n}, \tag{21}$$

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 [28]. The mutual information between $\mathbf{y}$ and $\mathbf{c}$, denoted as $\mathrm{MI}(\mathbf{y}, \mathbf{c})$, is defined as follows:

$$\mathrm{MI}(\mathbf{y}, \mathbf{c}) = \sum_{y_i \in \mathbf{y}, c'_j \in \mathbf{c}} p(y_i, c'_j) \log_2 \frac{p(y_i, c'_j)}{p(y_i)p(c'_j)}, \tag{22}$$

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:

$$\mathrm{NMI}(\mathbf{y}, \mathbf{c}) = \frac{\mathrm{MI}(\mathbf{y}, \mathbf{c})}{\max(\mathrm{H}(\mathbf{y}), \mathrm{H}(\mathbf{c}))}, \tag{23}$$

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 affect of randomness caused by k-means, and report the best result. Meanwhile, we randomly generate the "incomplete" patterns for 10 times in the above-mentioned way and report the statistical results. The aggregated ACC and NMI are used to evaluate the goodness of the algorithms in comparison. Taking the aggregated ACC for example, it is obtained by averaging the averaged ACC achieved by an algorithm over different ε. All experiments are conducted on a PC machine with an Intel(R) Core(TM)-i7-5820, 3.3 GHz CPU and 16G RAM in MATLAB environment.

5.2 Experimental Results on WebKB Datasets

We conduct experiments on four WebKB datasets, including Cornell, Texas, Washington and Wisconsin, to compare with PVC [7], which requires to access the original features and is only able to handle two views clustering tasks. Table 1 reports the aggregated ACC, NMI and the standard deviation, where the one with the highest performance is shown in bold. From Table 1, we observe that: i) The proposed MKKM-IK with zero, mean and KNN initializations consistently achieve comparable or better clustering performance among the MKKM methods with absent kernels on Cornell, Texas and Washington, and a little inferior to MKKM+KNN on Wisconsin; ii) The proposed MKKM-IK-MKC further significantly improves MKKM-IK and demonstrates the best performance in all the data sets; and iii) The improvement of MKKM-IK-MKC over existing algorithms is more significant. For example, it improves the second best algorithm (PVC) by nearly five percentage points on Texas in terms of aggregated clustering accuracy. We also provide the ACC and NMI comparison of the above algorithms with different missing ratios on Cornell, as shown in Fig. 1. These results are consistent with the ones reported in Table 1. Meanwhile, we provide the results on other three data sets in the appendix, available in the online supplemental material due to space limit.

TABLE 2
Datasets Used in Our Experiments

Dataset	#Samples	#Kernels	#Classes
Cornell	195	2	5
Texas	187	2	5
Washington	230	2	5
Wisconsin	265	2	5
Flower17	1360	7	17
Flower102	8189	4	102
Caltech101-5	510	48	102
Caltech101-10	1020	48	102
Caltech101-15	1530	48	102
Caltech101-20	2040	48	102
Caltech101-25	2550	48	102
Caltech101-30	3060	48	102
CCV	6773	3	20

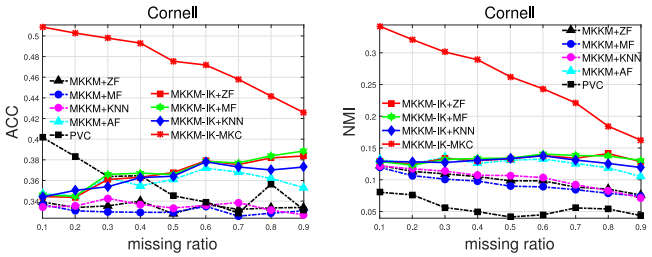


Fig. 1. ACC and NMI comparison with the variation of missing ratios on Cornell dataset. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported. The results on other data sets are provided in the appendix, available in the online supplemental material due to space limit.

5.3 Experimental Results on Caltech101

Caltech101 has been widely used as a benchmark dataset to evaluate the performance of multiple kernel clustering [5]. Here we also compare all the above-mentioned algorithms on this data set where the number of samples for each cluster varies in the range of $5, 10, \cdots, 30$. The PVC algorithm is not included into comparison since it can only handle two views clustering tasks and is required to assess original features.

The clustering results of different algorithms with the variation of missing ratio are reported in Fig. 2. As can be seen, compared with existing two-stage imputation algorithms, three curves corresponding to our proposed MKKM-IK with different initializations are on the top when the missing ratio varies from 0.1 to 0.9 in terms of ACC and NMI, indicating its superior clustering performance. Meanwhile, the proposed MKKM-IK-MKC further significantly improves the performance of MKKM-IK. Taking the results in Fig. (2c) for example. The proposed MKKM-IK with different initializations demonstrate the overall satisfying performance. However, MKKM-IK-MKC further significantly improves its performance. Moreover, from the Figs. 2a, 2b, 2c, 2d, 2e, 2f, 2g, 2h, 2i, 2j, and 2k, we clearly see that the improvement of our algorithms over the compared ones is more significant with the increase of number of samples. The aggregated ACC and NMI are also reported in Table 6. We again clearly see the advantages of our algorithms over the other ones in terms of ACC and NMI. These results have well demonstrated the effectiveness and advantages of incorporating kernel reconstruction in clustering.

5.4 Experimental Results on Flower17 and Flower102

We also compare the clustering performance of the above-mentioned algorithms on flower17 and flower102 data sets, which have been widely used as benchmarks in multiple kernel learning. The clustering results are shown in Fig. 3 and Table 3. Again, we observe that the proposed MKKM-IK outperforms the traditional imputation based algorithms, and MKKM-IK-MKC significantly improves MKKM-IK. Taking the result in Fig. (3a) for example, the proposed MKKM-IK-MKC exceeds the second best one by over ten percentage in terms of clustering accuracy when the missing ratio is 0.1. This superiority is consistently kept with the variation of missing ratio. Similar results can also be found from Figs. (3c) and (3d). Meanwhile, the aggregated ACC and NMI are also reported in Table 3,

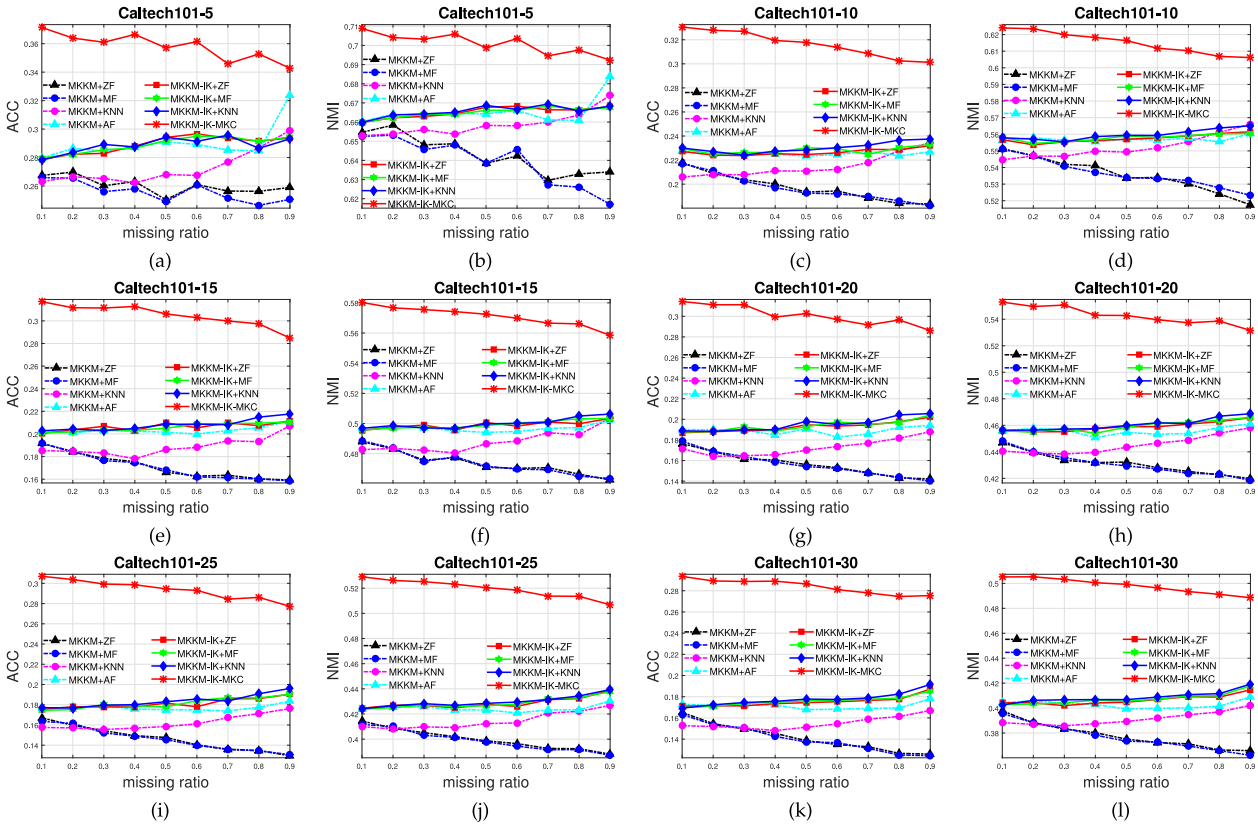


Fig. 2. ACC and NMI comparison with the variation of missing ratios on Caltech101. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported.

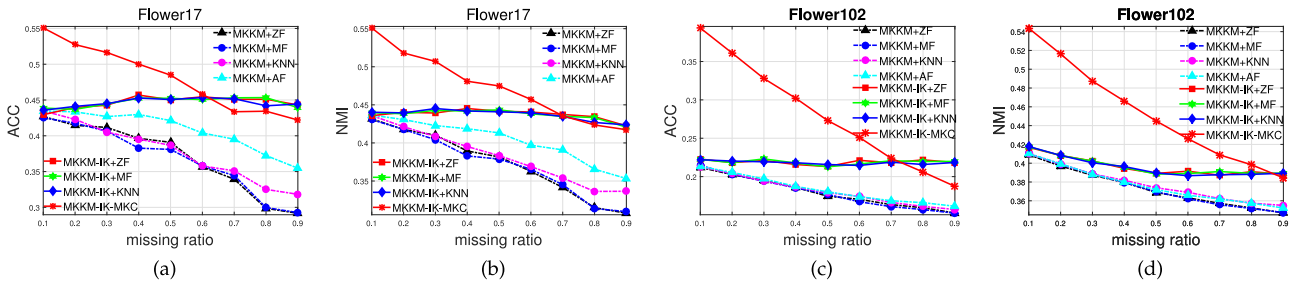


Fig. 3. ACC and NMI comparison with the variation of missing ratios on Flower17 and Flower102. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported.

TABLE 3
Aggregated ACC and NMI Comparison (mean±std) of Different Clustering Algorithms on Flower17 and Flower102

Datasets	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF [14]	MKKM-IK (proposed) ZF	MKKM-IK (proposed) KNN	MKKM-IK (proposed) MF	MKKM-IK (proposed) MKC
				ACC				
Flower17	37.33 ± 0.46	37.19 ± 0.43	38.11 ± 0.43	42.37 ± 0.46	43.84 ± 0.65	43.79 ± 0.57	43.90 ± 0.55	$\mathbf{54.09 \pm 0.49}$
Flower102	17.95 ± 0.12	17.90 ± 0.14	18.17 ± 0.16	18.37 ± 0.18	21.89 ± 0.16	21.90 ± 0.11	21.81 ± 0.14	$\mathbf{28.07 \pm 0.17}$
				NMI				
Flower17	37.63 ± 0.42	37.63 ± 0.40	38.46 ± 0.34	41.86 ± 0.30	42.98 ± 0.48	42.94 ± 0.52	42.98 ± 0.41	$\mathbf{53.10 \pm 0.19}$
Flower102	37.35 ± 0.09	37.37 ± 0.10	37.75 ± 0.12	37.64 ± 0.12	39.65 ± 0.10	39.67 ± 0.06	39.61 ± 0.16	$\mathbf{45.29 \pm 0.07}$

from which we also identify the superiority of the proposed MKKM-IK and MKKM-IK-MKC.

5.5 Experimental Results on CCV

We finally evaluate the performance of the proposed algorithms on CCV dataset, and report the results in Fig. 4 and Table 4. We once again observe that the proposed MKKM-IK and MKKM-IK-MKC significantly outperforms the compared ones in terms of ACC and NMI. Also, we observe that the proposed MKKM-IK-MKC is a little inferior to MKKM-IK from Fig. (4a) when the missing ratio is over 0.6. This is because there might be little useful information available for mutual kernel completion when the missing ratio of kernel matrices is relatively large.

In sum, we attribute the superiority of our algorithms to: 1) the joint optimization on imputation and clustering; and 2) the mutual kernel completion. 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 two learning processes negotiate with each other, leading to improved clustering performance. In contrast, MKKM+ZF, MKKM+MF, MKKM+KNN and MKKM +AF algorithms do not fully take advantage of the connection between the imputation and clustering procedures. This could produce imputation that does not well serve the subsequent clustering as originally expected, affecting the clustering performance. Moreover, the proposed mutual kernel completion well utilizes the available information to complete kernels, which further boosts the clustering performance.

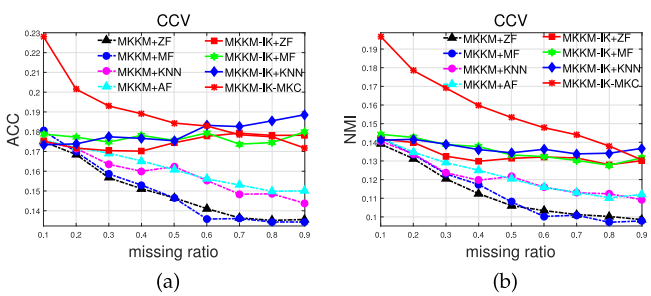


Fig. 4. ACC and NMI comparison with the variation of missing ratios on CCV. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported.

5.6 The Robustness of MKKM-IK-MKC to Noisy or Irrelevant Kernels

To explore the robustness of MKKM-IK-MKC to noisy or irrelevant kernels, we design an additional toy data experiment to explore what will happen if there are noisy or irrelevant kernels in the kernel set. To do so, we generate a random positive semi-definite (PSD) matrix to simulate the kernel matrix obtained with an irrelevant kernel function, and add it into the present kernel set of Flower17 dataset as the last kernel matrix. After that, we perform the aforementioned algorithms on this dataset and report the results in Fig. 6. As observed, the proposed MKKM-IK-MKC significantly outperforms the compared ones when the missing ratio is less than 0.5. When the missing ratio is greater than 0.6, MKKM-IK-MKC demonstrates comparable or slightly inferior performance when compared with the proposed variants without kernel construction. This is because the imputation from other kernel matrices may not be accurate anymore when there are a significant number of missing entries in these kernels, which in turn adversely affects the resultant clustering. Meanwhile, according to the aforementioned analysis, the kernel reconstruction term is able to reduce the kernel weights of irrelevant kernels, which is helpful to achieve robust clustering performance in the presence of irrelevant kernels.

TABLE 4
Aggregated ACC and NMI Comparison (mean±std) of Different Clustering Algorithms on CCV

Datasets	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF [14]	MKKM-IK (proposed) ZF	MKKM-IK (proposed) KNN	MKKM-IK (proposed) MF	MKKM-IK (proposed) MKC
				ACC				
CCV	14.96 ± 0.17	14.99 ± 0.15	15.87 ± 0.19	16.13 ± 0.22	17.50 ± 0.26	17.69 ± 0.31	17.96 ± 0.21	$\mathbf{18.96 \pm 0.24}$
				NMI				
CCV	11.25 ± 0.12	11.34 ± 0.14	12.11 ± 0.17	12.25 ± 0.19	13.30 ± 0.18	13.54 ± 0.23	13.70 ± 0.15	$\mathbf{15.75 \pm 0.16}$

We report the kernel combination weights learned by the aforementioned algorithms in Fig. 7. As can be seen from the Fig. 7h, the kernel combination weight corresponding to the noisy kernel (indexed by 8) learned by the proposed MKKM-IK-MKC is zero. This clearly demonstrates the advantage of incorporating kernel reconstruction into the objective. However, it is not the case for the rest of algorithms in comparison. The kernel weights corresponding to the last kernel learned by these algorithms are considerably greater than zero. This is because the kernel combination weight β_p is updated by Eq. (6) at each iteration, where $a_p = \mathrm{Tr}(\mathbf{K}_p(\mathbf{I} - \mathbf{H}\mathbf{H}^\top))$ and a_p is a limited positive number. This makes its weight β_p usually not zero. From this toy data experiments, we observe that the proposed MKKM-IK-MKC can automatically reduce the kernel weights of noisy or irrelavent kernels and achieve promising clustering performance.

5.7 Alignment Between the Original Kernels and the Imputed Ones

Besides comparing the above-mentioned algorithms in terms of clustering performance, we would like to gain more insight on how close the imputed base kernels (as a by-product of our algorithm) are to the ground-truth, i.e., the original, complete base kernels. To do this, we calculate the alignment between the ground-truth kernels and the imputed ones. The kernel alignment, a widely used criterion to measure the similarity of two kernel matrices, is used to serve this purpose [27]. We compare the alignment resulted from our algorithm with those from existing imputation algorithms. The results under various missing ratios are shown in Fig. 5. As observed, the kernels imputed by the proposed MKKM-IK align with the ground-truth kernels much better than those obtained by the existing imputation algorithms.

In particular, MKKM-IK+KNN wins the MKKM+AF by more than 9 percentage points on Caltech101 when the missing ratio is 0.9, as shown in Fig. (5a). The aggregated alignment and the standard deviation are reported in Table 5. We once again observe the significant superiority of the proposed MKKM-IK to the compared ones. These results indicate that our algorithm can not only achieve better clustering performance, but is also able to produce better imputation result by exploiting the prior knowledge of

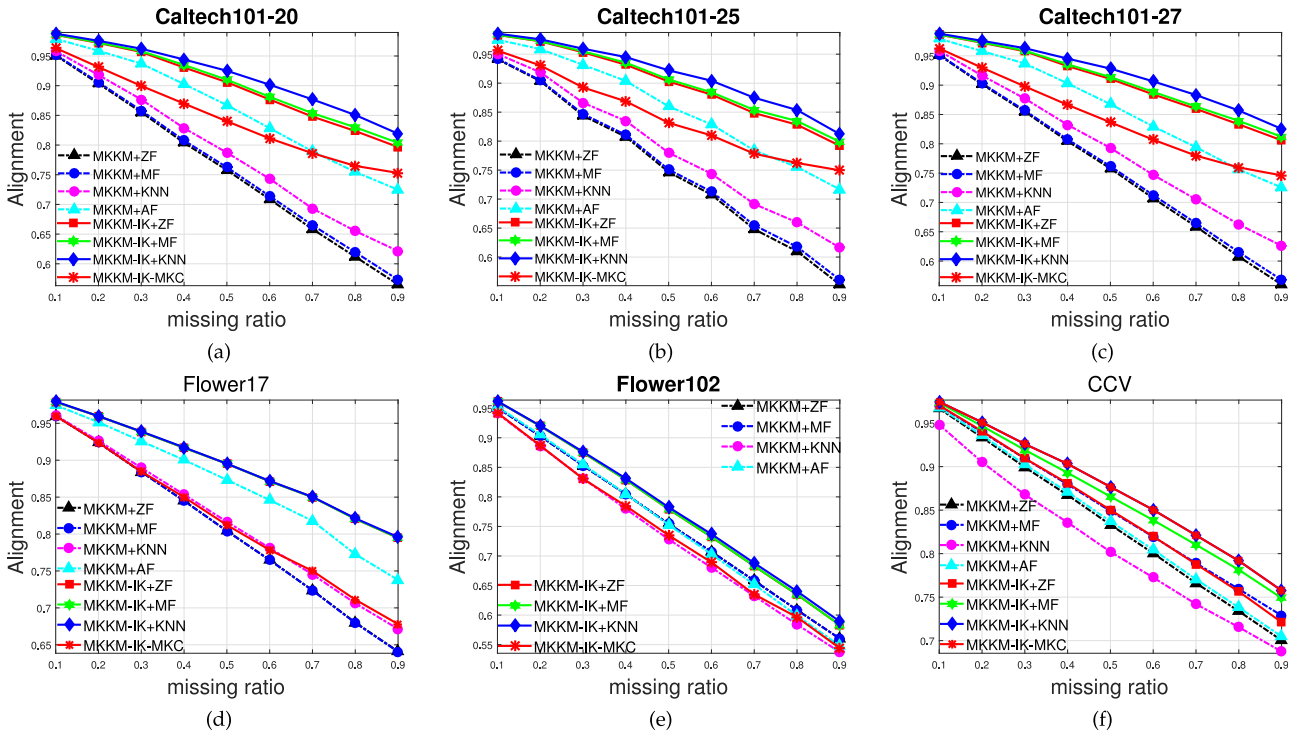


Fig. 5. Kernel alignment between the original kernels and the imputed kernels by different algorithms under different missing ratios. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported. The results on Caltech101-5, Caltech101-10 and Caltech101-15 are provided in the appendix, available in the online supplemental material due to space limit.

TABLE 5
Aggregated Alignment between the Original Kernels and the Imputed Kernels (mean±std) on All Data Sets

Datasets	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF [14]	MKKM-IK (proposed) ZF	MKKM-IK (proposed) KNN	MKKM-IK (proposed) MF	MKKM-IK (proposed) MKC
Flower17	80.05 ± 0.09	80.03 ± 0.09	81.44 ± 0.06	86.49 ± 0.07	89.04 ± 0.07	89.04 ± 0.06	$\mathbf{89.09 \pm 0.06}$	81.42 ± 0.08
Flower102	75.55 ± 0.05	75.55 ± 0.05	73.34 ± 0.03	75.24 ± 0.05	77.75 ± 0.05	77.75 ± 0.05	$\mathbf{78.07 \pm 0.05}$	73.82 ± 0.18
Caltech101-5	74.02 ± 0.32	74.42 ± 0.27	75.50 ± 1.06	84.51 ± 0.16	82.46 ± 0.95	82.93 ± 0.92	84.36 ± 0.98	$\mathbf{84.98 \pm 0.10}$
Caltech101-10	76.16 ± 0.18	76.63 ± 0.15	77.67 ± 0.32	85.89 ± 0.18	88.08 ± 0.24	88.49 ± 0.24	$\mathbf{89.93 \pm 0.20}$	85.39 ± 0.05
Caltech101-15	74.99 ± 0.09	75.47 ± 0.11	77.38 ± 0.25	85.35 ± 0.13	88.85 ± 0.13	89.28 ± 0.15	$\mathbf{90.61 \pm 0.09}$	84.51 ± 0.05
Caltech101-20	75.73 ± 0.13	76.20 ± 0.12	78.68 ± 0.21	86.02 ± 0.10	89.95 ± 0.14	90.34 ± 0.14	$\mathbf{91.59 \pm 0.09}$	84.66 ± 0.02
Caltech101-25	75.12 ± 0.10	75.58 ± 0.11	78.46 ± 0.18	85.71 ± 0.12	89.91 ± 0.17	90.27 ± 0.18	$\mathbf{91.47 \pm 0.14}$	84.22 ± 0.04
Caltech101-30	75.59 ± 0.08	76.01 ± 0.07	79.09 ± 0.12	86.11 ± 0.08	90.47 ± 0.09	90.78 ± 0.07	$\mathbf{91.91 \pm 0.05}$	84.29 ± 0.03
CCV	83.34 ± 0.05	84.94 ± 0.05	80.85 ± 0.05	83.69 ± 0.05	84.86 ± 0.06	86.41 ± 0.06	$\mathbf{87.25 \pm 0.06}$	$\mathbf{87.25 \pm 0.06}$

TABLE 6
Aggregated ACC and NMI Comparison (mean±std) of Different Clustering Algorithms on Caltech101

Datasets	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF [14]	MKKM-IK (proposed) ZF	MKKM-IK (proposed) KNN	MKKM-IK (proposed) MF	MKKM-IK (proposed) MKC
				ACC				
5	26.04 ± 0.34	25.60 ± 0.25	27.28 ± 0.30	29.02 ± 0.31	28.91 ± 0.20	28.91 ± 0.24	28.88 ± 0.38	$\mathbf{35.81 \pm 0.30}$
10	19.71 ± 0.19	19.67 ± 0.23	21.51 ± 0.20	22.53 ± 0.22	22.67 ± 0.18	22.83 ± 0.27	23.04 ± 0.18	$\mathbf{31.65 \pm 0.21}$
15	17.13 ± 0.24	17.09 ± 0.16	18.89 ± 0.13	20.34 ± 0.18	20.64 ± 0.15	20.59 ± 0.22	20.81 ± 0.18	$\mathbf{30.49 \pm 0.25}$
20	15.67 ± 0.12	15.65 ± 0.22	17.29 ± 0.16	18.89 ± 0.20	19.29 ± 0.11	19.37 ± 0.17	19.52 ± 0.12	$\mathbf{30.11 \pm 0.31}$
25	14.65 ± 0.18	14.58 ± 0.13	16.24 ± 0.13	17.71 ± 0.20	18.12 ± 0.15	18.16 ± 0.21	18.36 ± 0.21	$\mathbf{29.38 \pm 0.21}$
30	14.15 ± 0.12	14.05 ± 0.14	15.51 ± 0.16	17.13 ± 0.18	17.54 ± 0.28	17.60 ± 0.18	17.77 ± 0.12	$\mathbf{28.40 \pm 0.19}$
				NMI				
5	64.30 ± 0.16	63.93 ± 0.13	65.89 ± 0.21	66.53 ± 0.14	66.51 ± 0.12	66.50 ± 0.13	66.57 ± 0.21	$\mathbf{70.10 \pm 0.20}$
10	53.57 ± 0.11	53.63 ± 0.08	55.24 ± 0.11	55.70 ± 0.20	55.75 ± 0.15	55.80 ± 0.15	55.98 ± 0.14	$\mathbf{61.52 \pm 0.17}$
15	47.39 ± 0.13	47.38 ± 0.12	48.82 ± 0.11	49.70 ± 0.14	49.90 ± 0.10	49.93 ± 0.10	50.01 ± 0.15	$\mathbf{57.11 \pm 0.21}$
20	43.11 ± 0.10	43.08 ± 0.17	44.54 ± 0.12	45.58 ± 0.15	45.90 ± 0.14	45.94 ± 0.06	46.07 ± 0.11	$\mathbf{54.29 \pm 0.28}$
25	39.98 ± 0.10	39.88 ± 0.11	41.47 ± 0.09	42.45 ± 0.15	42.88 ± 0.15	42.88 ± 0.18	42.99 ± 0.12	$\mathbf{51.96 \pm 0.12}$
30	37.78 ± 0.08	37.66 ± 0.12	39.15 ± 0.13	40.29 ± 0.11	40.65 ± 0.14	40.74 ± 0.10	40.88 ± 0.11	$\mathbf{49.81 \pm 0.12}$

"serve clustering". It is worth pointing out that the kernel matrices imputed by the proposed MKKM-IK-MKC does not algin well with the original ones on some datasets such as Flower17 and Flower102, as shown in Figs. (5d) and (5e). This is because each incomplete kernel matrix is approximately optimized while the equality constraint in Eq. (12) may not be strictly guaranteed to keep anymore. This would reduce the alignment between the imputed kernel matrices and the original ones. The alignment results on Caltech101-5, Caltech101-10 and Caltech101-15 are provided in the appendix, available in the online supplemental material due to space limit.

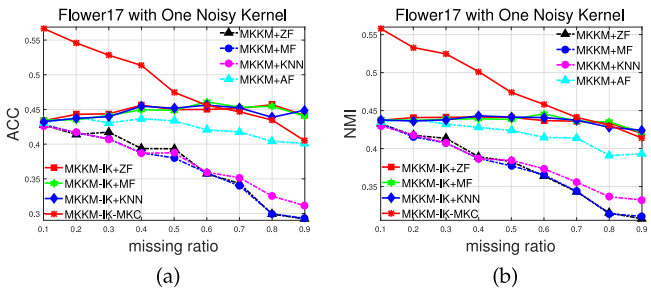


Fig. 6. Clustering accuracy and NMI comparison with the variation of missing ratios on Flower17 with an additional noisy kernel. For each given missing ratio, the "incomplete patterns" are randomly generated for 10 times and their averaged results are reported.

From the above experiments, we conclude that the proposed algorithm: 1) effectively addresses the issue of row/columns absence in multiple kernel clustering; 2) consistently achieves performance superior to the comparable ones, especially in the presence of intensive absence; and 3) can better recover the incomplete base kernels by taking into account the goal of clustering. In short, our algorithm well utilizes the connection between imputation and clustering procedures and mutual kernel completion, bringing forth significant improvements on clustering performance.

5.8 Convergence and Parameter Sensitivity

The proposed MKKM-IK is theoretically guaranteed to converge to a local minimum according to [29]. In our experiments, we observe that the objective value of this algorithm does monotonically decrease at each iteration and that it usually converges in less than 20 iterations. One examples of the evolution of the objective value on Flower17 are demonstrated in Fig. (8a).

Different from MKKM-IK which is parameter-free, the newly proposed MKKM-IK-MKC introduces a parameter λ to balance the objective of incomplete MKKM and kernel reconstruction. We conduct an additional experiment to show the effect of this parameter on the clustering performance. In Fig. 8b, we plot the ACC of MKKM-IK-MKC by varying λ from 2^{-15} to 2^{15} respectively, where the results of

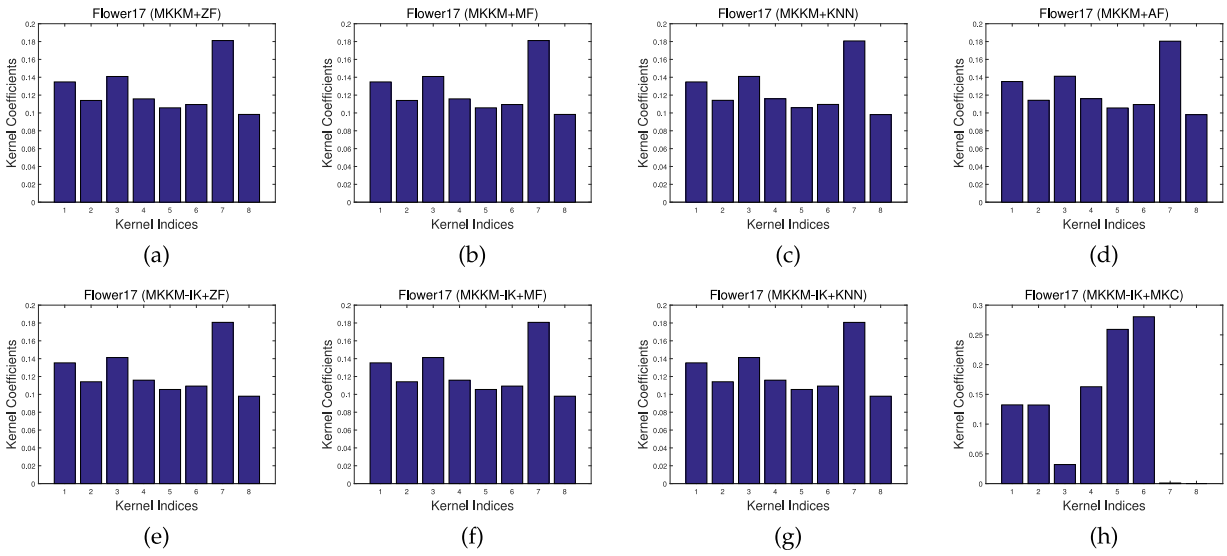


Fig. 7. Kernel coefficients learned by the aforementioned algorithms on Flower17 with an additional noisy kernel (with missing ratio=0.1). The base kernel indexed by 8 is a noisy one. We also observe that the results with other missing ratios are similar.

MKKM-IK+ZF is also incorporated as a baseline. From this figure, we observe that the newly proposed MKKM-IK-MKC significantly outperforms MKKM-IK+ZF and shows stable performance across a wide range of λ values.

We end up this section by discussing the convergence of the proposed MKKM-IK-MKC. Though the objective value of our algorithm cannot be theoretically guaranteed to monotonically decrease at each iteration due to the approximate optimization $\mathbf{K}_p$ in Eq. (12), we experimentally observe that it usually takes less than 10 iterations to satisfy the stopping criterion and demonstrates superior clustering performance.

6 Conclusion

While MKC algorithms have recently demonstrated promising performance in various applications, they are not able to effectively handle the scenario where base kernels are incomplete. This paper proposes to jointly optimize the kernel imputation and clustering to address this issue. It makes these two learning procedures seamlessly integrated to achieve better clustering. The proposed algorithm effectively solves the resultant optimization problem, and it demonstrates well improved clustering performance via extensive experiments on benchmark data sets, especially when the missing ratio is high. In the future, we plan to further improve the clustering performance by considering the correlations of different base kernels [16], [30], [31]. Moreover, the proposed algorithm is generic. We are going to extend it to other MKC algorithms that work with kernel such as spectral clustering [32]. Also, designing proper criteria [33], [34] for mutual kernel completion to satisfy various requirements of clustering tasks is interesting and worth exploring in future.

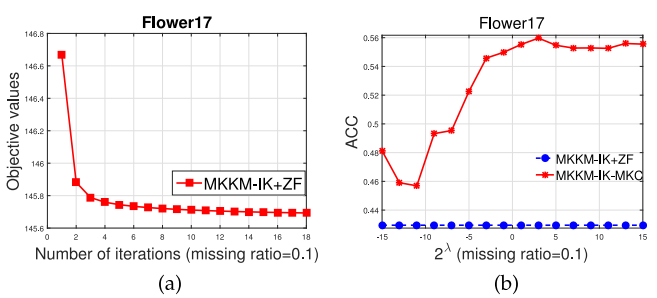


Fig. 8. (a) The objective value of the proposed MKKM-IK at each iteration. (b) The effect of λ on the proposed MKKM-IK-MKC in terms of ACC on Flower17.

Acknowledgments

This work was supported by National Key R&D Program of China 2018YFB1003203, the Natural Science Foundation of China (project no. 61701451 and 61672528), the German Research Foundation (DFG) awards KL 2698/2-1 and GRK1589/2 and the Federal Ministry of Science and Education (BMBF) awards 031L0023A, 01IS18051A. The authors wish to gratefully acknowledge Prof. Huiying Xu from Zhejiang Normal University for her help in the proofreading of this paper. Xinzhong Zhu and Xinwang Liu equally contributed to the paper.

References

[1] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 638–649.

[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 Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.

[3] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1305–1313.

[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 *Proc. 24th Int. Joint Conf. Artif. Intell.*, 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 *Proc. 30th AAAI Conf. Artif. Intell.*, 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 *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1704–1710.

[7] S. Li, Y. Jiang, and Z. Zhou, "Partial multi-view clustering," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1968–1974.

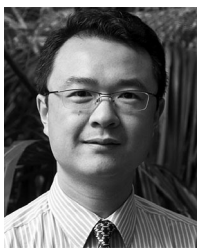
[8] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2408–2414.

[9] 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.*, 2018, pp. 1–1, doi: 10.1109/TPAMI.2018.2879108.
[10] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl.-Based Syst.*, vol. 163, pp. 510–517, 2019.
[11] 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 *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 185–193.
[12] 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 *Proc. IEEE 10th Int. Symp. Biomed. Imaging*, 2013, pp. 764–767.
[13] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.*, 1993, pp. 120–127.
[14] A. Trivedi, P. Rai, H. Daumé III, and S. L. DuVall, "Multiview clustering with incomplete views," in *Proc. Mach. Learn. Social Comput. Workshop*, 2010, pp. 1–4.
[15] C. Xu, D. Tao, and C. Xu, "Multi-view learning with incomplete views," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5812–5825, Dec. 2015.
[16] S. Bhadra, S. Kaski, and J. Rousu, "Multi-view kernel completion," *Mach. Learn.*, vol. 106, no. 5, pp. 713–739, May 2017.
[17] W. Shao, L. He, and P. S. Yu, "Multiple incomplete views clustering via weighted nonnegative matrix factorization with $\ell_{2,1}$ regularization," in *Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2015, pp. 318–334.
[18] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel k-means with incomplete kernels," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2259–2265.
[19] S. Jegelka, A. Gretton, B. Schölkopf, B. K. Sriperumbudur, and U. von Luxburg, "Generalized clustering via kernel embeddings," in *Proc. 32nd Annu. German Conf. AI Advances Artif. Intell.*, 2009, pp. 144–152.
[20] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "l_p-norm multiple kernel learning," *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, 2011.
[21] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 109–116.
[22] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, 2008.
[23] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," [Online]. Available: http://cvxr.com/cvx, Mar. 2014.
[24] A. Maurer and M. Pontil, "k-dimensional coding schemes in Hilbert spaces," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5839–5846, Nov. 2010.
[25] T. Liu, D. Tao, and D. Xu, "Dimensionality-dependent generalization bounds for k-dimensional coding schemes," *Neural Comput.*, vol. 28, no. 10, pp. 2213–2249, 2016.
[26] H. Zhao, H. Liu, and Y. Fu, "Incomplete multimodal visual data grouping," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2392–2398.
[27] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *J. Mach. Learn. Res.*, vol. 13, pp. 795–828, 2012.
[28] L. Lovász and M. D. Plummer, *Matching Theory*, Mathematics Studies, North Holland, 1986.
[29] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural Parallel Sci. Comput.*, vol. 11, no. 4, pp. 351–368, 2003.
[30] X. Liu, L. Wang, J. Yin, E. Zhu, and J. Zhang, "An efficient approach to integrating radius information into multiple kernel learning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 557–569, Apr. 2013.
[31] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2014.
[32] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[33] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210–219, 2017.
[34] Z. Kang, H. Pan, S. C. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, 2019, doi: 10.1109/TCYB.2018.2887094.

Xinwang Liu received the PhD degree from the 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. He has published more than 50 peer-reviewed papers, including those in highly regarded journals and conferences such as the *IEEE Transactions on Image Processing*, the *IEEE Transactions on Neural Networks and Learning Systems*, the *IEEE Transactions on Information Forensics and Security*, ICCV, AAAI, IJCAI, etc. He served on the Technical Program Committees of IJCAI 2016-2018 and AAAI 2016-2019.

Xinzhong Zhu received the PhD degree from Xidian University, China. He is a professor at College of Mathematics and Computer Science, Zhejiang Normal University, and also the president of Research Institute of Ningbo Cixing Co. Ltd, PR, China. His research interests include machine learning, computer vision, manufacturing informatization, robotics and system integration, and intelligent manufacturing. He is a member of the ACM and certified as CCF senior member. He has published more than 30 peer-reviewed papers, including those in highly regarded journals and conferences such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Multimedia*, the *IEEE Transactions on Knowledge and Data Engineering*, AAAI, IJCAI, etc.

Miaomiao Li is working toward the PhD degree at NUDT, China. She is now lecturer of Changsha College, Changsha, China. Her current research interests include kernel learning and multi-view clustering. She has published several peer-reviewed papers such as AAAI, IJCAI, Neurocomputing, etc. She serves on the Technical Program Committees of IJCAI 2017-2019.



Lei Wang received the PhD degree from Nanyang Technological University, Singapore. He is now associate professor at School of Computing and Information Technology of University of Wollongong, Australia. His research interests include machine learning, pattern recognition, and computer vision. He has published more than 120 peer-reviewed papers, including those in highly regarded journals and conferences such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, CVPR, ICCV and ECCV, etc. He was awarded the Early Career Researcher Award by Australian Academy of Science and Australian Research Council. He served as the General co-chair of DICTA 2014 and on the Technical Program Committees of more than 20 international conferences and workshops. He is a senior member of the IEEE.

En Zhu received the PhD degree from the 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. He has published more than 100 peer-reviewed papers, including the *IEEE Transactions on Circuits and Systems for Video Technology*, the *IEEE Transactions on Neural Networks and Learning Systems*, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.

Tongliang Liu received the PhD degree from the University of Technology Sydney. He is currently a lecturer with the School of Information Technologies and the Faculty of Engineering and Information Technologies, and a core member in the UBTECH Sydney AI Centre, at The University of Sydney. His research interests include statistical learning theory, computer vision, and optimisation. He has authored and co-authored more than 40 research papers including the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the the *Transactions on Neural Networks and Learning Systems*, the *IEEE Transactions on Image Processing*, ICML, CVPR, and KDD.

Marius Kloft received the PhD degree from TU Berlin and UC Berkeley. He 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.

Dinggang Shen is Jeffrey Houpt Distinguished Investigator, and a professor of Radiology, Biomedical Research Imaging Center (BRIC), Computer Science, and Biomedical Engineering in the University of North Carolina at Chapel Hill (UNC-CH). He is currently directing the Center for Image Analysis and Informatics, the Image Display, Enhancement, and Analysis (IDEA) Lab in the Department of Radiology, and also the medical image analysis core in the BRIC. He was a tenure-track assistant professor in the University of Pennsylvanian (UPenn), and a faculty member in the Johns Hopkins University. He research interests include medical image analysis, computer vision, and pattern recognition. He serves as an editorial board member for eight international journals. He has also served in the Board of Directors, the Medical Image Computing and Computer Assisted Intervention (MICCAI) Society, in 2012-2015, and will be General chair for MICCAI 2019. He is fellow of the IEEE, AIMBE and IAPR.

Jianping Yin received the PhD degree from the National University of Defense Technology (NUDT), China. He is now the distinguished professor at Dongguan University of Technology. His research interests include pattern recognition and machine learning. He has published more than 150 peer-reviewed papers, including the *IEEE Transactions on Circuits and Systems for Video Technology*, the *IEEE Transactions on Neural Networks and Learning Systems*, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation' Supervisor and National Excellence Teacher. He served on the Technical Program Committees of more than 30 international conferences and workshops.

Wen Gao received the PhD degree from the University of Tokyo, Japan. He is now Boya chair professor and the director of Faculty of Information and Engineering Sciences, Peking University, and the founding director of National Engineering Lab. for Video Technology (NELVT) at Peking University. He works in the areas of multimedia and computer vision, topics including video coding, video analysis, multimedia retrieval, face recognition, multimodal interfaces, and virtual reality. He published seven books, more than 220 papers in refereed journals, and more than 600 papers in selected international conferences. He served or serves on the editorial board for several journals, such as the *IEEE Transactions on Image Processing*, the *IEEE Transactions on Circuits and Systems for Video Technology*, the *IEEE Transactions on Multimedia*, the *IEEE Transactions on Autonomous Mental Development*. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME 2007, ACM Multimedia 2009, IEEE ISCAS 2013, and also served on the advisory and technical committees of numerous professional organizations. He has been featured by IEEE Spectrum in June 2005 as one of the "Ten To Watch" among China's leading technologists. He is a fellow of the IEEE, a fellow of ACM, and a member of Chinese Academy of Engineering.