Efficient and Effective Incomplete Multi-View Clustering

Xinwang Liu,[1] Xinzhong Zhu,[2,3] Miaomiao Li,[1] Chang Tang,[4] En Zhu,[1] Jianping Yin,[5] Wen Gao[6]
[1]School of Computer Science, National University of Defense Technology, Changsha, China, 410073
[2]College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua, Zhengjiang, China, 321004
[3]Research Institute of Ningbo Cixing Co., Ltd, Ningbo, China, 315336
[4]School of Computer Science, China University of Geosciences, Wuhan, China, 430074
[5]Dongguan University of Technology, Guangdong, China
[6]School of Electronics Engineering and Computer Science, Peking University, Beijing, China, 100871

Abstract

Incomplete multi-view clustering (IMVC) optimally fuses 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, overcomplicated optimization and limitedly improved clustering performance. In this paper, we 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. We carefully develop a three-step iterative algorithm to solve the resultant optimization problem with linear computational complexity and theoretically prove its convergence. Further, we conduct comprehensive experiments to study the proposed EE-IMVC in terms of clustering accuracy, running time, evolution of the learned consensus clustering matrix and the convergence. As indicated, our algorithm significantly and consistently outperforms some state-of-the-art algorithms with much less running time and memory.

Introduction

Multi-view clustering (MVC) optimally integrates features from different views to improve clustering performance (Bickel and Scheffer 2004). It has been intensively studied and widely applied into various applications during the last few decade (Yu et al. 2012; Li, Jiang, and Zhou 2014; Du et al. 2015; Liu et al. 2016; Li et al. 2016; Liu et al. 2017b; Li et al. 2015; Cai, Nie, and Huang 2013; Tao, Liu, and Fu 2017; Liu et al. 2013; Zhang et al. 2015; Tao et al. 2017). All these MVC algorithms assume that the views of samples are observable. However, in some practical applications (Kumar et al. 2013; Xiang et al. 2013), 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 (Ghahramani and Jordan 1993) and other advanced ones (Trivedi et al. 2010; Xu, Tao, and Xu 2015; Shao, He, and Yu 2015; Bhadra, Kaski, and Rousu 2016; Yin, Wu, and Wang 2015). 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 (Liu et al. 2017a). 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 (Liu et al. 2017a; Zhu et al. 2018), 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-scale clustering tasks. ii) *Overcomplicated 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 (Liu et al. 2017a). 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. We design a simple and computationally efficient algorithm to solve the resultant optimization problem by three singular value decomposition (SVD) per iteration, and analyze its computational and storage complexities and theoretically prove its convergence. After that, we conduct comprehensive experiments on four benchmark datasets to study the properties of the proposed algorithm, including the clustering accuracy with the various missing ratios, the running time with the various number of samples, 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.

Related Work

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 k and k is the number of clusters.

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.

The recently proposed MKKM-IK (Liu et al. 2017a) 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, (Liu et al. 2017a) 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 (Liu et al. 2017a).

Although MKKM-IK demonstrates excellent clustering performance in handling incomplete multi-view clustering tasks (Liu et al. 2017a), 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 (Liu et al. 2017a). 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. As a result, instead of imputing incomplete similarity matrices $\{\mathbf{K}_p\}_{p=1}^m$, we propose to impute the incomplete base clustering matrices to address the aforementioned issues. Moreover, we argue that this imputation is more natural and reasonable since

all of them reside in a common clustering partition space, which would produce better imputation and finally improve the clustering.

Efficient and Effective Incomplete Multi-view Clustering (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 matrices simultaneously. We firstly define the p-th $(1 \leq p \leq m)$ base clustering matrix as

$$\mathbf{H}_p = [\mathbf{H}_p^{(o)^\top}, \mathbf{H}_p^{(u)^\top}]^\top \in \mathbb{R}^{n \times k}, \tag{3}$$

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,

$$\begin{aligned} \max_{\mathbf{H}, \{\mathbf{W}_p, \mathbf{H}_p^{(u)}, \beta_p\}_{p=1}^m} & \operatorname{Tr}\left[\mathbf{H}^\top \sum\nolimits_{p=1}^m \beta_p \begin{pmatrix} \mathbf{H}_p^{(o)} \\ \mathbf{H}_p^{(u)} \end{pmatrix} \mathbf{W}_p\right] \\ s.t.\ & \mathbf{H} \in \mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ & \mathbf{W}_p \in \mathbb{R}^{k\times k},\ \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_k, \\ & \mathbf{H}_p^{(u)} \in \mathbb{R}^{(n-n_p)\times k},\ \mathbf{H}_p^{(u)^\top}\mathbf{H}_p^{(u)} = \mathbf{I}_k, \\ & \boldsymbol{\beta} \in \mathbb{R}^m,\ \sum\nolimits_{p=1}^m \beta_p^2 = 1,\ \beta_p \geq 0, \end{aligned} \tag{4}$$

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 (Liu et al. 2017a), the objective function of EE-IMVC in Eq. (4) 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 (Liu et al. 2017a), 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.

Alternate Optimization

Jointly optimizing $\mathbf{H}$, $\{\mathbf{H}_p^{(u)}, \mathbf{W}_p\}_{p=1}^m$ and $\boldsymbol{\beta}$ in Eq. (4) 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. (4) is equivalent to

$$\max_{\mathbf{H}}\ \operatorname{Tr}\left(\mathbf{H}^\top \mathbf{T}\right)\ s.t.\ \mathbf{H} \in \mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{5}$$

where $\mathbf{T} = \sum_{p=1}^m \beta_p \mathbf{H}_p \mathbf{W}_p$. It 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. (4) equivalently reduces to,

$$\max_{\mathbf{W}_p} \operatorname{Tr}\left(\mathbf{W}_p^\top \mathbf{Q}_p\right)\ s.t. \mathbf{W}_p \in \mathbb{R}^{k\times k},\ \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_k, \tag{6}$$

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. (4) is equivalent to

$$\begin{aligned} \max_{\mathbf{H}_p^{(u)}}\ & \operatorname{Tr}\left(\mathbf{H}_p^{(u)^\top} \mathbf{U}_p\right) \\ & s.t.\ \mathbf{H}_p^{(u)} \in \mathbb{R}^{(n-n_p)\times k},\ \mathbf{H}_p^{(u)^\top}\mathbf{H}_p^{(u)} = \mathbf{I}_k, \end{aligned} \tag{7}$$

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. (4) is equivalent to

$$\max_{\boldsymbol{\beta}}\ \boldsymbol{\nu}^\top \boldsymbol{\beta}\ \ s.t.\ \ \boldsymbol{\beta} \in \mathbb{R}^m,\ \sum\nolimits_{p=1}^m \beta_p^2 = 1,\ \beta_p \geq 0, \tag{8}$$

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. (8) 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. (8) is* $\boldsymbol{\beta}^\star = \boldsymbol{\nu}/\|\boldsymbol{\nu}\|$.

Proof 1 *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* $\nu_p^{(t)} = \mathrm{Tr}((\mathbf{H}^{(t)})^\top \mathbf{H}_p^{(t)} \mathbf{W}_p^{(t)}) = \max_{\mathbf{H}_p^{(u)}} \mathrm{Tr}\big((\mathbf{H}^{(t)})^\top [\mathbf{H}_p^{(o)^\top}, \mathbf{H}_p^{(u)^\top}]^\top \mathbf{W}_p^{(t)}\big) \geq \max_{\mathbf{W}_p} \mathrm{Tr}\big((\mathbf{H}^{(t)})^\top [\mathbf{H}_p^{(o)^\top}, \big(\mathbf{H}_p^{(u)(t-1)}\big)^\top]^\top \mathbf{W}_p\big) > 0, \forall p$. *The proof is completed by taking the derivative of the Lagrangian function of Eq. (8) on* β_p *and letting it vanish.*

Algorithm 1 The Proposed EE-IMVC

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

In sum, our algorithm for solving Eq. (4) is outlined in Algorithm 1, where $\mathrm{obj}^{(t)}$ 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 2 *Note that for* $\forall p, \mathrm{Tr}(\mathbf{H}^\top [\mathbf{H}_p^{(o)^\top}, \mathbf{H}_p^{(u)^\top}]^\top \mathbf{W}_p) \leq \frac{1}{2}[\mathrm{Tr}(\mathbf{H}^\top \mathbf{H}) + \mathrm{Tr}(\mathbf{W}_p^\top [\mathbf{H}_p^{(o)^\top}, \mathbf{H}_p^{(u)^\top}][\mathbf{H}_p^{(o)^\top}, \mathbf{H}_p^{(u)^\top}]^\top \mathbf{W}_p)] = \frac{1}{2}[2k + \mathrm{Tr}(\mathbf{W}_p^\top \mathbf{H}_p^{(o)^\top} \mathbf{H}_p^{(o)} \mathbf{W}_p)]$. *Note that the maximum of* $\mathrm{Tr}(\mathbf{W}_p^\top \mathbf{H}_p^{(o)^\top} \mathbf{H}_p^{(o)} \mathbf{W}_p)$ *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* $\mathrm{Tr}\big(\mathbf{H}^\top [\mathbf{H}_p^{(o)^\top}, \mathbf{H}_p^{(u)^\top}]^\top \mathbf{W}_p\big) \leq \frac{1}{2}[2k + \sum_{j=1}^k \lambda_p^j] \triangleq a_p$. *Correspondingly,* $\sum_{p=1}^m \beta_p \mathrm{Tr}(\mathbf{H}^\top [\mathbf{H}_p^{(o)^\top}, \mathbf{H}_p^{(u)^\top}]^\top \mathbf{W}_p) \leq \sum_{p=1}^m \beta_p a_p$, *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.*

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 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 has a linear computational complexity with number of samples, which enables it more efficiently to handle large scale clustering tasks when compared with MKKM-IK (Liu et al. 2017a).

Storage complexity: During the learning procedure, EE-IMVC needs 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 in our experiments. Further exploring other initializations and studying their influence on the clustering performance will be an interesting future work.

Extensions: EE-IMVC can be extended from the following aspects. Firstly, EE-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 (Kato and Rivero 2017) 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 (von Luxburg 2007). 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.

Experiments

Experimental settings

The proposed algorithm is experimentally evaluated on four widely used multiple kernel benchmarkdata sets shown in Table 2. They are Oxford Flower17 and Flower102[1], Caltech102[2] and Columbia Consumer Video (CCV)[3]. For these datasets, all kernel matrices are pre-computed and can be

[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/

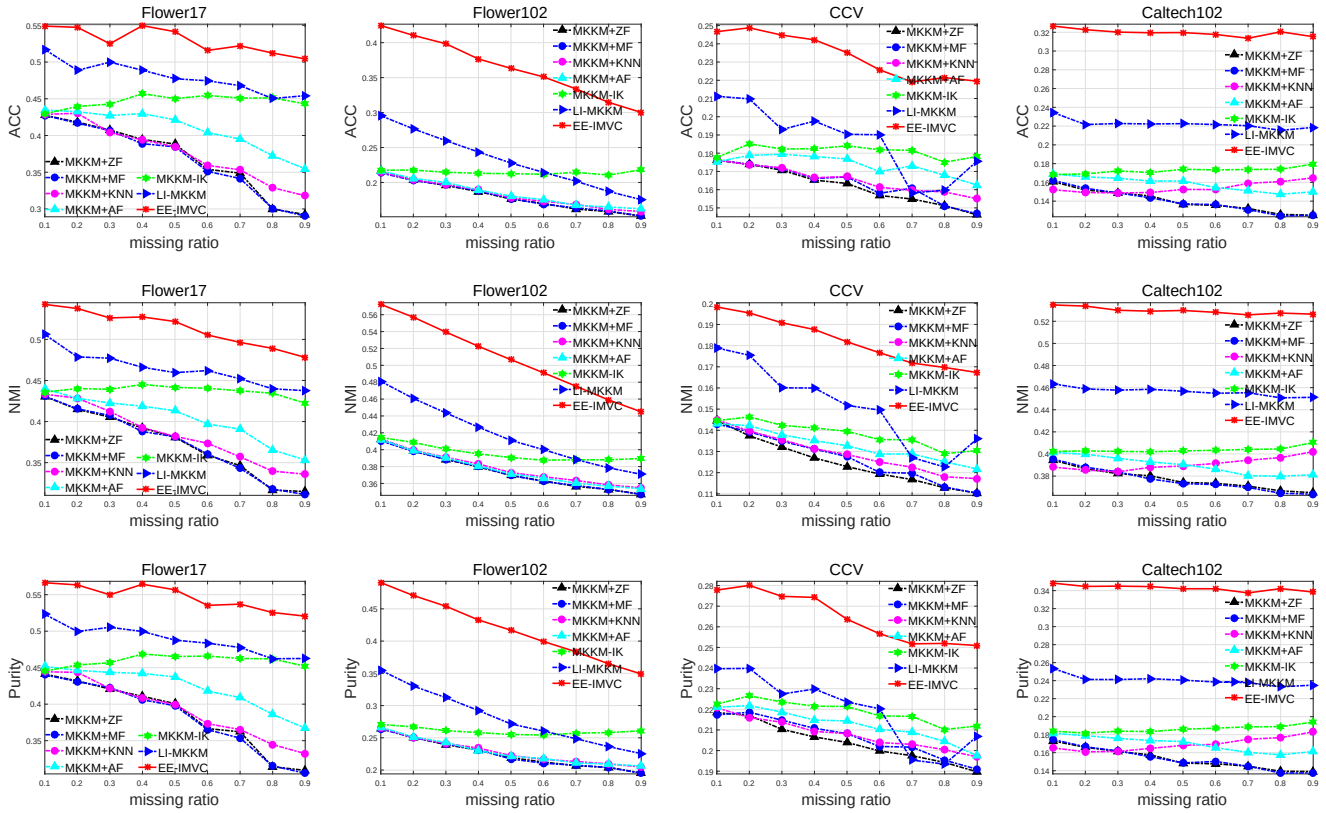

Figure 1: ACC, NMI, Purity and Rand Index comparison with the variation of missing ratios on four benchmark datasets.

Table 1: Aggregated ACC, NMI, purity and rand index comparison (mean±std) of different clustering algorithms on benchmark datasets.

Datasets	MKKM+ZF	MKKM+MF	MKKM+KNN	MKKM+AF (Trivedi et al. 2010)	MKKM-IK (Liu et al. 2017a)	LI-MKKM (Zhu et al. 2018)	EE-IMVC Proposed
ACC							
Flower17	37.0 ± 0.7	36.8 ± 0.6	37.8 ± 0.3	40.8 ± 0.3	44.7 ± 0.3	47.9 ± 0.3	$\mathbf{52.9 \pm 0.7}$
Flower102	18.0 ± 0.2	18.0 ± 0.2	18.3 ± 0.1	18.5 ± 0.1	21.5 ± 0.2	23.1 ± 0.2	$\mathbf{36.4 \pm 0.2}$
CCV	16.2 ± 0.1	16.4 ± 0.1	16.6 ± 0.2	17.4 ± 0.1	18.1 ± 0.2	18.7 ± 0.3	$\mathbf{23.4 \pm 0.4}$
Caltech102	14.0 ± 0.1	14.0 ± 0.1	15.4 ± 0.2	15.8 ± 0.1	17.3 ± 0.2	22.1 ± 0.2	$\mathbf{31.9 \pm 0.2}$
NMI							
Flower17	37.3 ± 0.5	37.3 ± 0.4	38.4 ± 0.2	40.3 ± 0.3	43.7 ± 0.3	46.3 ± 0.2	$\mathbf{51.4 \pm 0.6}$
Flower102	37.4 ± 0.1	37.4 ± 0.1	37.8 ± 0.1	37.7 ± 0.1	39.6 ± 0.1	41.7 ± 0.1	$\mathbf{50.8 \pm 0.1}$
CCV	12.5 ± 0.1	12.7 ± 0.1	12.9 ± 0.1	13.3 ± 0.1	13.8 ± 0.2	15.1 ± 0.2	$\mathbf{18.2 \pm 0.2}$
Caltech102	37.7 ± 0.1	37.7 ± 0.1	39.1 ± 0.1	38.9 ± 0.1	40.4 ± 0.2	45.6 ± 0.1	$\mathbf{52.9 \pm 0.1}$
Purity							
Flower17	38.4 ± 0.6	38.2 ± 0.5	39.3 ± 0.3	42.2 ± 0.3	45.9 ± 0.7	48.8 ± 0.3	$\mathbf{54.7 \pm 0.7}$
Flower102	22.5 ± 0.1	22.4 ± 0.1	22.9 ± 0.1	22.9 ± 0.2	26.0 ± 0.2	28.0 ± 0.2	$\mathbf{41.8 \pm 0.2}$
CCV	20.4 ± 0.1	20.7 ± 0.1	20.8 ± 0.1	21.2 ± 0.1	21.9 ± 0.2	22.0 ± 0.2	$\mathbf{26.5 \pm 0.4}$
Caltech102	15.3 ± 0.1	15.3 ± 0.1	16.9 ± 0.1	17.0 ± 0.1	18.6 ± 0.2	23.9 ± 0.1	$\mathbf{34.3 \pm 0.2}$

publicly downloaded from the above websites. Their number of samples varies from one thousand to over eight thousands, and views from four to 48.

We compare the proposed EE-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 (Trivedi et al. 2010). The widely used MKKM (Gönen and Margolin 2014) is applied with these imputed base kernels. These two-stage methods are termed MKKM+ZF, MKKM+MF,

Table 2: Datasets used in our experiments.

Dataset	#Samples	#Kernels	#Classes
Flower17	1360	7	17
Flower102	8189	4	102
CCV	6773	6	20
Caltech102	3060	48	102

MKKM+KNN and MKKM+AF, respectively. In addition, we compare with the recently proposed MKKM-IK (Liu et al. 2017a), which jointly optimizes the imputation and clustering.

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 (Liu et al. 2017a; Zhu et al. 2018) 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 $[0.1 : 0.1 : 0.9]$.

The widely used clustering accuracy (ACC), normalized mutual information (NMI) and purity are applied to evaluate the clustering performance. For 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 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 from the following four aspects: clustering performance, running time, the evolution of the learned consensus clustering matrix and convergence.

Clustering Performance

Figure 1 presents the ACC and NMI comparison of the above algorithms with different missing ratios on all datasets. We have the following observations: 1) The recently proposed MKKM-IK (Liu et al. 2017a) (in green) outperforms existing two-stage imputation methods. For example, it exceeds the best two-stage imputation method (MKKM+AF) by 0.1%, 1.2%, 1.7%, 2.7%, 2.8%, 4.4%, 4.7%, 6.9% and 6.9% in terms of NMI, with the variation of missing ratios in $[0.1, \cdots, 0.9]$ on Flower17. These results verify the effectiveness of its joint optimization on imputation and clustering. 2) The proposed EE-IMVC significantly and consistently outperforms MKKM-IK. For example, it improves the latter by 10.7%, 9.7%, 8.7%, 8.2%, 8.0%, 6.5%, 5.9%, 5.5% and 5.6% 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. 3) The superiority of EE-IMVC is more significant when the missing ratio is relatively small. For example, EE-IMVC improves the second best algorithm (MKKM-IK) by 10.7% on Flower17 in terms of NMI when the missing ratio is 0.1 (see Figure 1(c)). The curves in terms of purity and rand index are provided in the supplemental material due to space limit.

We also report the aggregated ACC, NMI, purity and rand index, and the standard deviation in Table 1, where the one with the highest performance is shown in bold. Again, we observe that the proposed algorithm significantly outperforms MKKM+ZF, MKKM+MF, MKKM+KNN, MKKM+AF and MKKM-IK. For example, EE-IMVC exceeds the second best one (MKKM-IK) by 8.3%, 14.9%, 5.3% and 14.7% in terms of clustering accuracy on Flower17, Flower102, CCV and Caltech102, respectively. These results are consistent with our observations in Figure 1.

The above experimental results on these datasets have well demonstrated that EE-IMVC is superior to some state-of-the-art in terms of clustering accuracy, NMI, purity and rand index. We attribute the superiority of EE-IMVC as two 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 imputes 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.

Running Time

To compare the computational efficiency of the above-mentioned algorithms, we design another experiment to study the relationship between running time and the number of samples. To see this point in depth, we randomly select samples from the four benchmark datasets, run the aforementioned algorithms, record their running time, and plot them in Figure 4. We have the following observations from these figures: 1) The running time of EE-IMVC is nearly linear with the number of samples. 2) The superiority of EE-IMVC is more significant with the increase of samples, indicating its computational efficiency in handling large-scale clustering tasks. In sum, the experimental results in Figure 4 have well demonstrated the computational advantage of EE-IMVC.

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 NMI of EE-IMVC

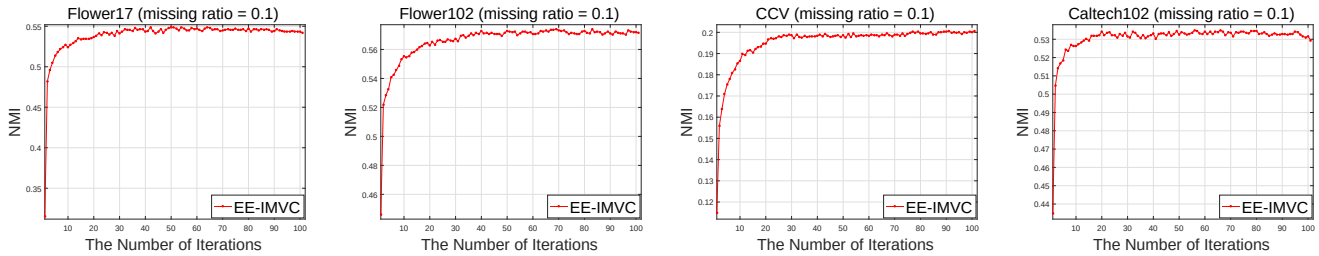

Figure 2: The evolution of the learned consensus clustering matrix $\mathbf{H}$ with missing ratio 0.1 on all datasets. The curves with other missing ratios are similar and we omit them due to space limit.

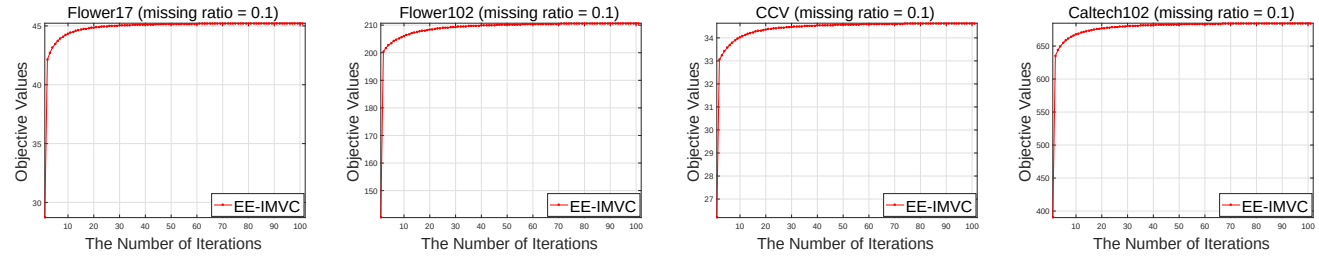

Figure 3: The objective value of EE-IMVC with iterations with missing ratio 0.1 on all datasets. The curves with other missing ratios are similar and we omit them due to space limit.

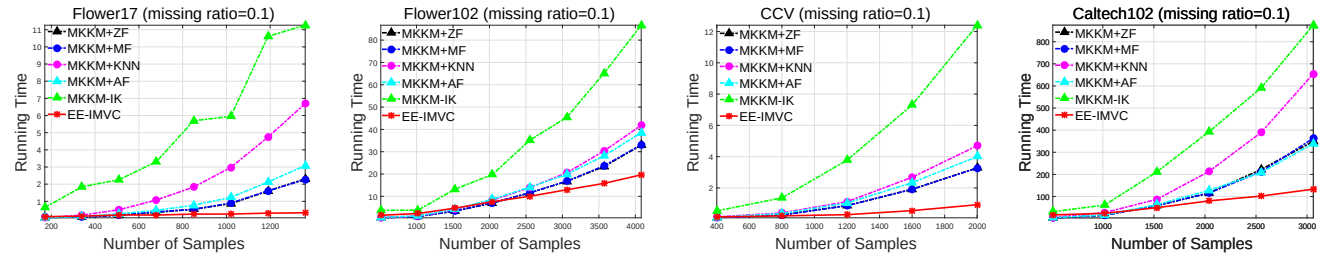

Figure 4: Running time comparison of different algorithms with various number of samples with missing ratio 0.1 on all datasets. The curves with other missing ratios are similar and omitted due to space limit.

based on the $\mathbf{H}$ learned at each iteration on Flower102 and CCV and plot the curves in Figure 2. From these figures, we observe that the NMI of EE-IMVC gradually increases to a maximum and generally maintains it up to slight variation. These experiments have clearly demonstrated the effectiveness of learned consensus clustering matrix, indicating the advantage of imputing incomplete base clustering matrices, instead of imputing incomplete kernel matrices. Other curves in terms of ACC and purity have similar trend and are omitted due to space limit.

Convergence

Our algorithm is theoretically guaranteed to converge according to Theorem 2. We record the objective values of EE-IMVC with iterations on all datasets and plot them in Figure 3. As observed, the objective value of EE-IMVC does monotonically increase at each iteration and that it usually converges in less than 50 iterations.

Conclusion

While the recently proposed MKKM-IK (Liu et al. 2017a) is able to handle incomplete multi-view clustering, the relatively high computational and space complexities prevent it from large scale clustering tasks. This paper proposes a late fusion approach to simultaneously clustering and imputing the incomplete base clustering matrices. The proposed algorithm effectively and efficiently solves the resultant optimization problem, and demonstrates well improved clustering performance via extensive experiments on benchmark datasets. In the future, we plan to explore the correlation among base clustering matrices and use it to further improve the imputation.

Acknowledgements

This work was supported by National Key R&D Program of China 2018YFB1003203, the Natural Science Foundation of China (project no. 61773392). The authors wish to grate-

fully acknowledge Prof. Huiying Xu from Zhejiang Normal University for her help in the proofreading of this paper. Xinwang Liu and Xinzhong Zhu equally contribute to the paper. Xinzhong Zhu is the corresponding author of this paper.

References

Bhadra, S.; Kaski, S.; and Rousu, J. 2016. Multi-view kernel completion. In *arXiv:1602.02518*.

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

Cai, X.; Nie, F.; and Huang, H. 2013. Multi-view k-means clustering on big data. In *IJCAI*, 2598–2604.

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

Ghahramani, Z., and Jordan, M. I. 1993. Supervised learning from incomplete data via an EM approach. In *NIPS*, 120–127.

Gönen, M., and Margolin, A. A. 2014. Localized data fusion for kernel k-means clustering with application to cancer biology. In *NIPS*, 1305–1313.

Kato, T., and Rivero, R. 2017. Mutual kernel matrix completion. In *arXiv:1702.04077v2*.

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

Li, Y.; Nie, F.; Huang, H.; and Huang, J. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*, 2750–2756.

Li, M.; Liu, X.; Wang, L.; Dou, Y.; Yin, J.; and Zhu, E. 2016. Multiple kernel clustering with local kernel alignment maximization. In *IJCAI*, 1704–1710.

Li, S.; Jiang, Y.; and Zhou, Z. 2014. Partial multi-view clustering. In *AAAI*, 1968–1974.

Liu, J.; Wang, C.; Danilevsky, M.; and Han, J. 2013. Large-scale spectral clustering on graphs. In *IJCAI*, 1486–1492.

Liu, X.; Dou, Y.; Yin, J.; Wang, L.; and Zhu, E. 2016. Multiple kernel k-means clustering with matrix-induced regularization. In *AAAI*, 1888–1894.

Liu, X.; Li, M.; Wang, L.; Dou, Y.; Yin, J.; and Zhu, E. 2017a. Multiple kernel k-means with incomplete kernels. In *AAAI*, 2259–2265.

Liu, X.; Zhou, S.; Wang, Y.; Li, M.; Dou, Y.; Zhu, E.; and Yin, J. 2017b. Optimal neighborhood kernel clustering with multiple kernels. In *AAAI*, 2266–2272.

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

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2017. From ensemble clustering to multi-view clustering. In *IJCAI*, 2843–2849.

Tao, Z.; Liu, H.; and Fu, Y. 2017. Simultaneous clustering and ensemble. In *AAAI*, 1546–1552.

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

von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17(4):395–416.

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

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view learning with incomplete views. *IEEE Trans. Image Processing* 24(12):5812–5825.

Yin, Q.; Wu, S.; and Wang, L. 2015. Incomplete multi-view clustering via subspace learning. In *ACM CIKM*, 383–392.

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

Zhang, R.; Li, S.; Fang, T.; Zhu, S.; and Quan, L. 2015. Joint camera clustering and surface segmentation for large-scale multi-view stereo. In *ICCV*, 2084–2092.

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