Hindawi
Computational Intelligence and Neuroscience
Volume 2017, Article ID 3961718, 11 pages
https://doi.org/10.1155/2017/3961718

*Research Article*

# Consensus Kernel $K$-Means Clustering for Incomplete Multiview Data

**Yongkai Ye,[1] Xinwang Liu,[1] Qiang Liu,[1] and Jianping Yin[2]**

[1]*College of Computer, National University of Defense Technology, Changsha, China*
[2]*State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, China*

Correspondence should be addressed to Yongkai Ye; yeyongkai@nudt.edu.cn

Received 28 April 2017; Revised 28 August 2017; Accepted 6 September 2017; Published 22 October 2017

Academic Editor: Ezequiel López-Rubio



Multiview clustering aims to improve clustering performance through optimal integration of information from multiple views. Though demonstrating promising performance in various applications, existing multiview clustering algorithms cannot effectively handle the view's incompleteness. Recently, one pioneering work was proposed that handled this issue by integrating multiview clustering and imputation into a unified learning framework. While its framework is elegant, we observe that it overlooks the consistency between views, which leads to a reduction in the clustering performance. In order to address this issue, we propose a new unified learning method for incomplete multiview clustering, which simultaneously imputes the incomplete views and learns a consistent clustering result with explicit modeling of between-view consistency. More specifically, the similarity between each view's clustering result and the consistent clustering result is measured. The consistency between views is then modeled using the sum of these similarities. Incomplete views are imputed to achieve an optimal clustering result in each view, while maintaining between-view consistency. Extensive comparisons with state-of-the-art methods on both synthetic and real-world incomplete multiview datasets validate the superiority of the proposed method.

## 1. Introduction

The term "multiview data" refers to data that have different sources or modalities. Each source or modality is considered as one "view," and different views have different physical meanings and statistical properties. For example, a web page can be described by the pictures and text it contains, while a news story may be reported by different sites each with its own different viewpoints. A significant number of studies aimed to investigate and learn from multiple views in the past [1, 2]. Multiview clustering, which is one component of multiview learning, aims at grouping samples by utilizing information from different views. Extensive research has been conducted into multiview clustering; these can be roughly categorized into early fusion approaches and late fusion approaches. Early fusion approaches fuse the multi-view information in an early stage of the process and then perform clustering [3–9], while late fusion approaches group data by fusing previously clustered results from separate views [10, 11].

However, in real-world applications, some views may be incomplete for a variety of reasons, which hurts the clustering performance of multiview data. For example, in the context of patient grouping, the data from different tests can serve as different views. If a test is too expensive, some patients may be unable to afford it, which leads to an incomplete view for this particular test. Similarly, in webpage clustering, image data and text data are two modalities that represent a page; however, some pages may not contain any images, which makes the data for the image view incomplete.

Existing studies of incomplete multiview clustering can be roughly divided into two categories: subspace methods and imputation methods. The method outlined in [12], which was the first subspace method for incomplete multiview clustering, learns the common subspace of two views via non-negative matrix factorization. Several variants of this method were proposed following its introduction. In [13], feature learning is integrated into the subspace learning process and the assumption that the data is nonnegative is not required.

The method proposed in [14] learns a latent global graph representation and the subspace simultaneously by adding a novel Laplacian graph regularization term. The other important category of method for incomplete multiview clustering is imputation methods, which handle incomplete views by filling in the missing parts. The method proposed in [15] fills the kernel of an incomplete view according to the Laplacian regularization of the other complete view. Subsequently, the method proposed in [16] tackles the situation where two views are incomplete by alternately updating one view according to the other view. In [17], the incomplete views are imputed via low rank decomposition. As different views are assumed to be generated from a shared subspace, the data matrices of different views can be decomposed using a common factor. Most of these imputation methods simply execute a conventional multiview clustering algorithm after filling the incomplete views. Most recently, a method was proposed in [18], whereby the imputation is not separated from the multiview clustering process. More specifically, the imputation and the multiple kernel clustering are integrated into a unified procedure for better clustering performance.

Integrating imputation and multiview clustering into a unified learning process makes the imputation better serve the clustering objective. This advantage helps the method in [18] to outperform other methods that perform imputation and clustering separately. However, the disadvantage of the method in [18] is that multiview clustering solution it proposes overlooks the consistency between views, which may reduce the final clustering performance. In [18], multiview clustering is achieved by learning a linear combination of kernels that reaches the optimal kernel $k$-means clustering result. Consequently, the linear combination to build the best kernel for clustering is learned without considering the relationships between views. Similarly, the imputation is guided only by the clustering objective and the consistency between views is neglected. However, the consistency between views is one of the inherent properties of multiview data [1]; if this critical property is not considered, the learning of the linear combination of kernels and the imputation in [18] may lead to poor clustering performance. Previous research into multiview clustering has shown that considering the consistency between views helps to boost the performance of multiview clustering [3]. In this study, we wish to build on the advances made in [18] while also considering the consistency between views in order to further improve clustering performance. Therefore, we propose a novel incomplete multiview clustering method that simultaneously fills the incomplete kernels from incomplete views and learns a consistent clustering result. To model the between-view consistency, the similarity between the consistent clustering result and the clustering result of each view is calculated. The consistency between views is measured by the sum of these similarities. The missing parts of kernels and the consistent clustering result are learned in order to achieve the optimal clustering result in each view while keeping consistency between views. Here, the learning process considers both the data structures within views and the consistent relations between views, which benefits the multiview clustering performance. The proposed objective function is then solved by alternately optimizing partial variables. Each subproblem that optimizes the corresponding partial variables either can be solved by means of eigenvector decomposition or has a closed-form solution. To evaluate the performance of the proposed method, we compare it with state-of-the-art methods on three synthetic and one real-world incomplete multiview datasets. Empirical results validate the superiority of the proposed method for incomplete multiview clustering.

The main contributions of this paper can be summarized as follows:

(1) We propose a novel incomplete multiview clustering method, which simultaneously learns a consistent clustering decision and fills the incomplete kernels from incomplete views with explicit modeling of between-view consistency.

(2) We design an alternating optimization algorithm to solve proposed method's optimization problem. Here, the optimization problem is divided into three subproblems. The subproblems either can be solved by means of eigenvector decomposition or has a closed-form solution.

(3) We also provide thorough convergence analysis of the alternating optimization algorithm, including theoretical proof and empirical validations.

## 2. The Proposed Method

Regarding the consistency, we propose that a consistent clustering decision be learned that is similar to each view's kernel $k$-means clustering result. To handle the incomplete views, we simultaneously fill the incomplete views and learn the consistent clustering decision. In the following subsections, we first introduce the notation used in problem formulation, after which kernel $k$-means is briefly reviewed. We then outline how a consistent decision might be found. Next, we introduce the objective function of our method to explain how the kernel filling and decision learning processes are integrated. Finally, we analyse the convergence of the proposed algorithm.

*2.1. Notation.* Assume that there are $N$ samples and $P$ views for the multiview data. For clarity, sample's information in a view is referred to as an instance of the sample in this paper. For incomplete multiview data, a sample's instance in a view could be missing. $\mathbf{S}$ is an $N \times P$ zero-one matrix that indicates which instances are missing; when $\mathbf{S}_{ij} = 0$, sample $i$'s instance in view $j$ is missing. $\mathbf{S}_j$ denotes the $j$th column of $\mathbf{S}$. Because our method is based on kernel $k$-means, we assume that the input multiview data is kernel data. For each view $j$, we have a $N \times N$ kernel matrix $\mathbf{K}_j$. The details of how kernel data are built can be found in Section 3.1, where datasets used in this paper are introduced.

In a view $j$, some instances may be missing, which will lead to an incomplete kernel $\mathbf{K}_j$. To describe the visible and missing parts of the incomplete kernel $\mathbf{K}_j$, we define an operator $\mathbf{K}(\text{row}S, \text{col}S)$, which selects corresponding rows and columns of $\mathbf{K}$ according to zero-one vectors row$S$ and

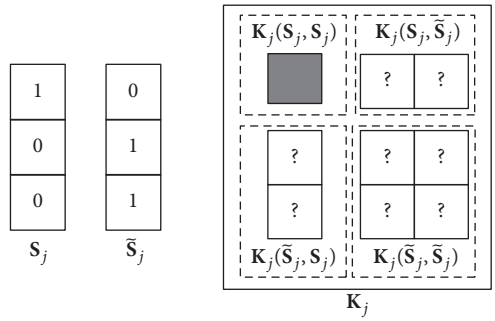


Figure 1: A simple example of notation. $\mathbf{S}_j$ indicates that only the instance of the first sample is visible in view $j$. $\widetilde{\mathbf{S}}_j$ is derived from $\mathbf{S}_j$. The kernel matrix of view $j$ is $\mathbf{K}_j$. $\mathbf{K}_j$ can be divided into four parts according to $\mathbf{S}_j$ and $\widetilde{\mathbf{S}}_j$. It is notable that only $\mathbf{K}_j(\mathbf{S}_j, \mathbf{S}_j)$ is visible.

col$S$ (where 1 indicates selected). Moreover, we define $\widetilde{\mathbf{S}}_j = \mathbf{1} - \mathbf{S}_j$. Thus $\mathbf{K}_j(\mathbf{S}_j, \mathbf{S}_j)$ is the visible part of the kernel matrix, while $\mathbf{K}_j(\mathbf{S}_j, \widetilde{\mathbf{S}}_j)$, $\mathbf{K}_j(\widetilde{\mathbf{S}}_j, \mathbf{S}_j)$, and $\mathbf{K}_j(\widetilde{\mathbf{S}}_j, \widetilde{\mathbf{S}}_j)$ are the missing parts. Figure 1 shows a simple example of notation with three samples.

*2.2. Kernel k-Means.* Here, kernel $k$-means refers to the $k$-means method developed for kernel data. Define a mapping from $\mathcal{X}$ to a reproducing kernel Hilbert space $\mathcal{H} : \phi(\cdot) : x \in \mathcal{X} \to \mathcal{H}$. $\{x_i\}_{i=1}^n \in \mathcal{X}$ is the sample set. A zero-one matrix $\mathbf{Z} \in \{0,1\}^{N\times K}$ is used to store cluster information, where $\mathbf{Z}_{ic} = 1$ indicates that sample $i$ is in cluster $c$. The $k$-means objective in kernel space is as follows:

$$\begin{aligned} \min_{\mathbf{Z}\in\{0,1\}^{N\times K}} \quad & \sum_{c=1}^{K}\sum_{i=1}^{N} \mathbf{Z}_{ic} \left\|\phi\left(x_i\right) - \mu_c\right\|_2^2 \\ \text{s.t.} \quad & \sum_{c=1}^{K} \mathbf{Z}_{ic} = 1, \quad i = 1, \ldots, N, \end{aligned} \tag{1}$$

where $\mu_c$ is the average of samples in cluster $c$. The number of samples in cluster $c$ is $N_c = \sum_{i=1}^{N} \mathbf{Z}_{ic}$, such that $\mu_c = (1/N_c)\sum_{i=1}^{N} \mathbf{Z}_{ic}\phi(x_i)$. The kernel matrix is denoted by $\mathbf{K}$, where $\mathbf{K}_{ij} = \phi(x_i)^T\phi(x_j)$. Define matrix $\mathbf{L} = \mathrm{diag}([N_1^{-1}, N_2^{-1}, \ldots, N_k^{-1}])$, so that the equivalent matrix form of (1) is as follows:

$$\begin{aligned} \min_{\mathbf{Z}\in\{0,1\}^{N\times K}} \quad & \mathrm{tr}\left(\mathbf{K}\right) - \mathrm{tr}\left(\mathbf{L}^{1/2}\mathbf{Z}^T\mathbf{K}\mathbf{Z}\mathbf{L}^{1/2}\right) \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{1}_K = \mathbf{1}_N, \end{aligned} \tag{2}$$

where $\mathrm{tr}(\cdot)$ is the trace operator and $\mathbf{1}_K$ is a $K$-length vector in which all elements are 1.

The discreteness of $\mathbf{Z}$ makes (2) difficult to solve. An approximated problem that is easier to solve can be arrived at by relaxing the discreteness constraints on $\mathbf{Z}$. By denoting $\mathbf{U} = \mathbf{Z}\mathbf{L}^{1/2}$, the approximated problem can be expressed as follows:

$$\begin{aligned} \min_{\mathbf{U}\in\mathbb{R}^{N\times K}} \quad & \mathrm{tr}\left[\mathbf{K}\left(\mathbf{I} - \mathbf{U}\mathbf{U}^T\right)\right] \\ \text{s.t.} \quad & \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{3}$$

The optimal $\mathbf{U}$ can be solved by obtaining eigenvectors corresponding to $k$ larger eigenvalues of $\mathbf{K}$ [9]. Although $\mathbf{U}$ contains the cluster indicator information, $k$-means should be performed on $\mathbf{U}$ to recover the actual clustering label.

*2.3. Finding the Consistent Decision.* So as to consider the consistency between views, we propose to find a consistent clustering decision according to the clustering results of different views. Suppose $\mathbf{U}_j$ is the eigenvector matrix found by kernel $k$-means in view $j$. $\mathbf{U}_j$, while not the actual clustering label of view $j$, does store the cluster information. Accordingly, we can find a matrix $\mathbf{U}^*$ that is consistent with all $\mathbf{U}_j$ and then recover the final decision from $\mathbf{U}^*$.

To find the consistent $\mathbf{U}^*$, it is necessary to define the similarity between $\mathbf{U}^*$ and $\mathbf{U}_j$. Inspired by [3, 19], the similarity is defined as

$$L\left(\mathbf{U}_j, \mathbf{U}^*\right) = \left\langle \frac{\mathbf{U}_j\mathbf{U}_j^T}{\left\|\mathbf{U}_j\mathbf{U}_j^T\right\|_F}, \frac{\mathbf{U}^*\mathbf{U}^{*T}}{\left\|\mathbf{U}^*\mathbf{U}^{*T}\right\|_F} \right\rangle, \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm. Adding regularization $\mathbf{U}^{*T}\mathbf{U}^* = \mathbf{I}$ on $\mathbf{U}^*$, we have

$$L\left(\mathbf{U}_j, \mathbf{U}^*\right) = \mathrm{tr}\left(\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}^*\mathbf{U}^{*T}\right). \tag{5}$$

There may be other possible definitions of similarity between $\mathbf{U}_j$ and $\mathbf{U}^*$. However, (4) is chosen because it allows an easy alternating optimization for the proposed method.

As expected that the consistent decision should be similar to the kernel $k$-means result of each view, we maximize the sum of similarities to find the consistent decision as follows:

$$\sum_{j=1}^{P} L\left(\mathbf{U}_j, \mathbf{U}^*\right) = \sum_{j=1}^{P} \mathrm{tr}\left(\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}^*\mathbf{U}^{*T}\right). \tag{6}$$

It is notable that each view is considered to be equally important in (6). If the importance of each view is prior knowledge, we can weigh the views differently and adapt (6) to a weighted sum of similarities. However, in this paper, we maintain the same weight for all views for model simplicity. Although learning the accurate weights of views is valuable under circumstances where there are some views with heavy noise, it is beyond the scope of this paper.

*2.4. Objective Function.* If all views are complete, it is easy to find the consistent decision by maximizing (6). When some views are incomplete, however we need to fill the corresponding kernels of those views for kernel $k$-means. We expect that these filled kernels will lead to better clustering in each view and a consistent decision. In other words, the

kernel filling is guided by both the clustering objective in each view and the consistency between views. So the filling procedure considers both the data structure within each view and the relationship between views. To achieve this, we propose the objective function as follows:

$$\begin{aligned}\min_{\{\mathbf{U}_j\},\mathbf{U}^*,\{\mathbf{K}_j\}} \quad & \sum_{j=1}^{P}\mathrm{tr}\left[\mathbf{K}_j\left(\mathbf{I}-\mathbf{U}_j\mathbf{U}_j^T\right)\right]-\beta\sum_{j=1}^{P}L\left(\mathbf{U}_j,\mathbf{U}^*\right)\\ \text{s.t.} \quad & \mathbf{U}_j^T\mathbf{U}_j=\mathbf{I}, \quad \forall j=1,\ldots,P,\\ & \mathbf{U}^{*T}\mathbf{U}^*=\mathbf{I},\\ & \mathbf{K}_j\left(\mathbf{S}_j,\mathbf{S}_j\right)=\widehat{\mathbf{K}}_j\left(\mathbf{S}_j,\mathbf{S}_j\right), \quad \forall j=1,\ldots,P,\\ & \mathbf{K}_j\succeq 0, \quad \forall j=1,\ldots,P,\end{aligned} \tag{7}$$

where $L(\mathbf{U}_j,\mathbf{U}^*)=\mathrm{tr}(\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}^*\mathbf{U}^{*T})$. $\mathbf{K}_j$ is the kernel that needs to be learned, which should be positive semidefinite. $\widehat{\mathbf{K}}_j(\mathbf{S}_j,\mathbf{S}_j)$ is the visible part of the original kernel data. The third constraint actually forces $\mathbf{K}_j(\mathbf{S}_j,\mathbf{S}_j)$ to be the same as the original kernel data. However, $\mathbf{K}_j(\widetilde{\mathbf{S}}_j,\mathbf{S}_j)$, $\mathbf{K}_j(\mathbf{S}_j,\widetilde{\mathbf{S}}_j)$, and $\mathbf{K}_j(\widetilde{\mathbf{S}}_j,\widetilde{\mathbf{S}}_j)$ still need to be optimized.

It is notable that the objective function consists of two parts. $\sum_{j=1}^{P}\mathrm{tr}[\mathbf{K}_j(\mathbf{I}-\mathbf{U}_j\mathbf{U}_j^T)]$ is the sum of kernel $k$-means objective in each view, and $\sum_{j=1}^{P}L(\mathbf{U}_j,\mathbf{U}^*)$ is the term designed to model between-view consistency. A parameter $\beta$ is added to balance the importance of single view clustering performance and the consistency between views.

*Remark 1.* Like the method proposed in [18], our method simultaneously fills incomplete kernels and performs multiview clustering. However, there are also major differences between the two methods. In [18], multiview clustering is achieved by learning the best combination of kernels for the best clustering performance, which overlooks the consistency between views. Differently, our method learns a consensus clustering decision from each view's kernel $k$-means result, which explicitly models the consistency. More importantly, our method does not simply revise the method in [18] incrementally by adding a consistency regularization term; instead, we propose a new objective function that inherits the advantages of simultaneously performing imputation and multiview clustering.

*Remark 2.* The strategy of learning consistent clustering decision was also applied in [3, 19]. The former work is based on spectral clustering, while the latter one is based on kernel $k$-means. But it is worth noting that these works cannot deal with the incomplete multiview situation.

*2.5. Optimization.* Optimizing all variables of (7) in one step is difficult. Instead, we develop an algorithm to solve the problem where $\mathbf{U}_j$, $\mathbf{U}^*$, and $\mathbf{K}_j$ are optimized alternatively. The optimal solutions of the subproblems can be found easily, and the whole alternating updating process is guaranteed to converge to a local minimum.

*2.5.1. Updating* $\mathbf{U}_j$. When we only optimize $\mathbf{U}_j$, the subproblem has a similar form to kernel $k$-means and can be solved by means of eigenvalue decomposition in a similar way. The subproblem of updating $\mathbf{U}_j$ is as follows:

$$\begin{aligned}\max_{\mathbf{U}_j} \quad & \mathrm{tr}\left[\mathbf{U}_j^T\left(\mathbf{K}_j+\beta\mathbf{U}^*\mathbf{U}^{*T}\right)\mathbf{U}_j\right]\\ \text{s.t.} \quad & \mathbf{U}_j^T\mathbf{U}_j=\mathbf{I}.\end{aligned} \tag{8}$$

*2.5.2. Updating* $\mathbf{U}^*$. Similarly, the subproblem of updating $\mathbf{U}^*$ can be solved by means of eigenvalue decomposition after reformulation. The subproblem of updating $\mathbf{U}^*$ is as follows:

$$\begin{aligned}\max_{\mathbf{U}^*} \quad & \sum_{j=1}^{P}\mathrm{tr}\left(\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}^*\mathbf{U}^{*T}\right)\\ \text{s.t.} \quad & \mathbf{U}^{*T}\mathbf{U}^*=\mathbf{I}.\end{aligned} \tag{9}$$

Equation (9) is equivalent to the following optimization problem:

$$\begin{aligned}\max_{\mathbf{U}^*} \quad & \mathrm{tr}\left\{\mathbf{U}^{*T}\left(\sum_{j=1}^{P}\mathbf{U}_j\mathbf{U}_j^T\right)\mathbf{U}^*\right\}\\ \text{s.t.} \quad & \mathbf{U}^{*T}\mathbf{U}^*=\mathbf{I}.\end{aligned} \tag{10}$$

*2.5.3. Updating* $\mathbf{K}_j$. The subproblem for $\mathbf{K}_j$ is an optimization problem with positive semidefinite constraint. Let $\mathbf{V}_j=\mathbf{I}-\mathbf{U}_j\mathbf{U}_j^T$, so that the subproblem is as follows:

$$\begin{aligned}\min_{\mathbf{K}_j} \quad & \mathrm{tr}\left(\mathbf{K}_j\mathbf{V}_j\right)\\ \text{s.t.} \quad & \mathbf{K}_j\left(\mathbf{S}_j,\mathbf{S}_j\right)=\widehat{\mathbf{K}}_j\left(\mathbf{S}_j,\mathbf{S}_j\right),\\ & \mathbf{K}_j\succeq 0.\end{aligned} \tag{11}$$

Because $\mathbf{K}_j$ is positive semidefinite, $\mathbf{K}_j$ can be decomposed as $\mathbf{A}_j\mathbf{A}_j^T$, where $\mathbf{A}_j$ is a $N\times 1$ vector [15, 18]. If we obtain $\mathbf{A}_j$, $\mathbf{K}_j$ can be recovered.

For clarity, we divide $\mathbf{A}_j$ into two parts: $\mathbf{A}_j^v=\mathbf{A}_j(\mathbf{S}_j,1)$ and $\mathbf{A}_j^m=\mathbf{A}_j(\widetilde{\mathbf{S}}_j,1)$. $\mathbf{A}_j^v$ is selected according to the indexes of the visible instance in view $j$, and $\mathbf{A}_j^m$ is selected according to the indexes of the missing instance in view $j$. Therefore, the kernel matrix of view $j$ can be divided into four parts as follows:

$$\begin{aligned}\mathbf{K}_j^{vv}&=\mathbf{K}_j\left(\mathbf{S}_j,\mathbf{S}_j\right)=\mathbf{A}_j^v\mathbf{A}_j^{vT},\\ \mathbf{K}_j^{vm}&=\mathbf{K}_j\left(\mathbf{S}_j,\widetilde{\mathbf{S}}_j\right)=\mathbf{A}_j^v\mathbf{A}_j^{mT},\\ \mathbf{K}_j^{mv}&=\mathbf{K}_j\left(\widetilde{\mathbf{S}}_j,\mathbf{S}_j\right)=\mathbf{A}_j^m\mathbf{A}_j^{vT},\\ \mathbf{K}_j^{mm}&=\mathbf{K}_j\left(\widetilde{\mathbf{S}}_j,\widetilde{\mathbf{S}}_j\right)=\mathbf{A}_j^m\mathbf{A}_j^{mT}.\end{aligned} \tag{12}$$

It is notable that $\mathbf{K}_j^{vv}$ is the only visible part. The $N\times N$ matrix $\mathbf{V}_j$ can be divided into four corresponding parts in a similar

way to $\mathbf{K}_j$. According to the first constraint in (11), $\mathbf{A}_j^v \mathbf{A}_j^{vT} = \widehat{\mathbf{K}}_j(\mathbf{S}_j, \mathbf{S}_j)$. To obtain $\mathbf{A}_j^m$, we have a problem equivalent to (11) as follows:

$$\min_{\mathbf{A}_j^m} \quad \mathrm{tr}\left( \left[\mathbf{A}_j^v; \mathbf{A}_j^m\right]^T \begin{bmatrix} \mathbf{V}_j^{vv} & \mathbf{V}_j^{vm} \\ \mathbf{V}_j^{mv} & \mathbf{V}_j^{mm} \end{bmatrix} \left[\mathbf{A}_j^v; \mathbf{A}_j^m\right] \right). \tag{13}$$

Taking the derivative of (13), we can obtain the closed-form solution for $\mathbf{A}_j^m$:

$$\mathbf{A}_j^m = -\left(\mathbf{V}_j^{vm} \mathbf{V}_j^{mm-1}\right)^T \mathbf{A}_j^v. \tag{14}$$

By denoting $\mathbf{V}_j^{vm} \mathbf{V}_j^{mm-1}$ as $\mathbf{V}_j^{vm/mm}$, the missing parts of $\mathbf{K}_j$ can be calculated as

$$\begin{aligned} \mathbf{K}_j^{vm} &= -\widehat{\mathbf{K}}_j\left(\mathbf{S}_j, \mathbf{S}_j\right) \mathbf{V}_j^{vm/mm}, \\ \mathbf{K}_j^{mv} &= -\left(\mathbf{V}_j^{vm/mm}\right)^T \widehat{\mathbf{K}}_j\left(\mathbf{S}_j, \mathbf{S}_j\right), \\ \mathbf{K}_j^{mm} &= \left(\mathbf{V}_j^{vm/mm}\right)^T \widehat{\mathbf{K}}_j\left(\mathbf{S}_j, \mathbf{S}_j\right) \mathbf{V}_j^{vm/mm}. \end{aligned} \tag{15}$$

The overall optimization process is summarized in Algorithm 1.

*2.6. Convergence Property.* In this subsection, we provide a theoretical proof of the convergence of the proposed optimization algorithm. First, we need to prove that the objective value of (7) is lower-bounded.

**Lemma 3.** *if* $\mathbf{K} \succeq 0$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, *then* $\mathrm{tr}[\mathbf{K}(\mathbf{I} - \mathbf{U}\mathbf{U}^T)] \geq 0$.

*Proof.* Denoting $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$, because $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, we have $\mathbf{U}\mathbf{U}^T\mathbf{U} = \mathbf{U}$. So $\mathbf{U}\mathbf{U}^T\mathbf{u}_j = \mathbf{u}_j$, $1 \leq j \leq K$. This implies that $\mathbf{U}\mathbf{U}^T$ has $K$ eigenvalues with 1. Moreover, because the rank of $\mathbf{U}\mathbf{U}^T$ is not larger than $K$, the remaining $N - K$ eigenvalues are 0.

Therefore, $\mathbf{I} - \mathbf{U}\mathbf{U}^T$ is positive semidefinite and can thus be decomposed as $\mathbf{v}\mathbf{v}^T$. Because $\mathbf{K} \succeq 0$, $\mathrm{tr}[\mathbf{K}(\mathbf{I} - \mathbf{U}\mathbf{U}^T)] = \mathrm{tr}(\mathbf{K}\mathbf{v}\mathbf{v}^T) = \mathbf{v}^T\mathbf{K}\mathbf{v} \geq 0$. □

**Lemma 4.** *One has the following:*

$$-\mathrm{tr}\left(\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}^*\mathbf{U}^{*T}\right) \geq -N. \tag{16}$$

*Proof.* According to the definitions of Frobenius norm and trace, we have the following:

$$\begin{aligned} \left\|\mathbf{U}_j\mathbf{U}_j^T - \mathbf{U}^*\mathbf{U}^{*T}\right\|_F^2 &= \mathrm{tr}\left[\left(\mathbf{U}_j\mathbf{U}_j^T - \mathbf{U}^*\mathbf{U}^{*T}\right)^2\right] \\ &= \mathrm{tr}\left[\left(\mathbf{U}_j\mathbf{U}_j^T\right)^2\right] \\ &\quad + \mathrm{tr}\left[\left(\mathbf{U}^*\mathbf{U}^{*T}\right)^2\right] \\ &\quad - 2\mathrm{tr}\left(\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}^*\mathbf{U}^{*T}\right). \end{aligned} \tag{17}$$

**Input**:
Incomplete multi-view data: $\widehat{\mathbf{K}}_1, \widehat{\mathbf{K}}_2, \ldots, \widehat{\mathbf{K}}_P$
Indicator matrix: $\mathbf{S}$
Number of clusters: $K$
Balance parameter: $\beta$
Stopping gap: $\epsilon$
**Output**:
The consistent decision: $\mathbf{U}^*$
Filled kernels: $\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_P$
(1) Initialize filled kernels with zero-filling
(2) Initialize $\{\mathbf{U}_j\}_{j=1}^P$ by performing kernel-$k$ means on initialized filled kernels
(3) Initialize $\mathbf{U}^*$ by Eq. (10)
(4) **repeat**
(5) Update $\{\mathbf{K}_j\}_{j=1}^P$ by solving Eq. (11)
(6) Update $\{\mathbf{U}_j\}_{j=1}^P$ by solving Eq. (8)
(7) Update $\mathbf{U}^*$ by solving Eq. (10)
(8) **until** Objective difference smaller than $\epsilon$
(9) **return** $\mathbf{U}^*$, $\{\mathbf{K}_j\}_{j=1}^P$

Algorithm 1: Consensus kernel $k$-means clustering for incomplete multiview clustering.

Following the constraints in (7), we have $\mathbf{U}_j^T\mathbf{U}_j = \mathbf{I}$ and $\mathbf{U}^{*T}\mathbf{U}^* = \mathbf{I}$. So, $\mathrm{tr}[(\mathbf{U}_j\mathbf{U}_j^T)^2] = \mathrm{tr}[(\mathbf{U}^*\mathbf{U}^{*T})^2] = \mathrm{tr}(\mathbf{I}) = N$. Finally, we have $-\mathrm{tr}(\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}^*\mathbf{U}^{*T}) = (1/2)\|\mathbf{U}_j\mathbf{U}_j^T - \mathbf{U}^*\mathbf{U}^{*T}\|_F^2 - N \geq -N$. □

According to Lemmas 3 and 4, the objective value of (7) is lower-bounded. Moreover, because we obtain the optimal solution to the corresponding subproblem in each step of the alternate updating, the objective value of (7) is therefore nonincreasing during this process. Since the objective value is lower-bounded and nonincreasing, the alternate updating algorithm is guaranteed to converge.

## 3. Experiments

*3.1. Datasets.* One incomplete multiview dataset and three complete multiview datasets are used in the experiments, as shown in Table 1. 3 Sources, the incomplete multiview dataset, has been compiled from three news sites: BBC, Reuters, and the Guardian. The dataset contains 416 news stories, and articles for some stories are missing from each site. More information about 3 Sources can be found in Table 2. Artificial incomplete multiview data are generated from complete multiview datasets using a random missing mechanism. The details of the generating process can be found in Section 3.3. For Digital (https://github.com/HoiYe/DigitalDataset), Flower 17 (http://www.robots.ox.ac.uk/~vgg/data/flowers/17/) and Flower 102 (http://www.robots.ox.ac.uk/~vgg/data/flowers/102/), and precomputed kernel matrices are used. As for 3 Sources, we generate Gaussian kernels with widths set as the mean of sample pair distances.

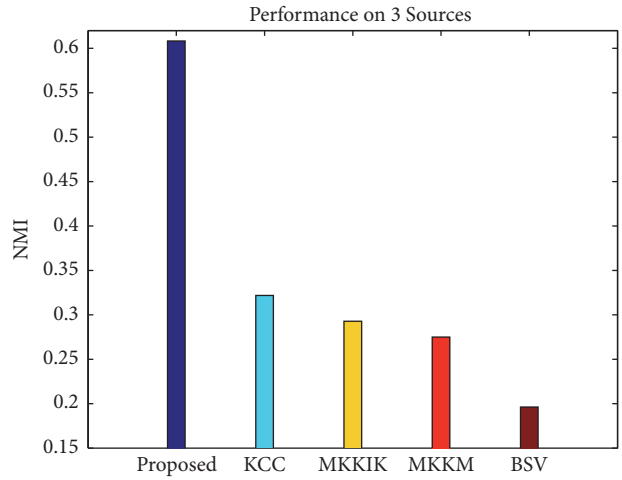


Figure 2: Performance comparison on real-world dataset in terms of NMI.

*3.2. Compared Methods.* The proposed method is compared with three state-of-the-art methods including one of the latest imputation methods and two representative subspace methods. The best clustering result of a single view and the multiview clustering result with zero-filling kernels, as important baselines, are also compared.

*Best Result of a Single View (BSV).* We perform clustering with the remaining samples in each view and choose the best. Because the view is incomplete, the missing samples are assigned random labels, after which the overall performance is reported.

*Multiple Kernel k-Means (MKKM).* Multiple kernel $k$-means is applied to the zero-filling kernels.

*Multiple Kernel k-Means with Incomplete Kernels (MKKIK).* The algorithm proposed in [18] learns the missing parts and performs multiple kernel $k$-means simultaneously.

*Partial View Clustering (PVC).* The subspace method proposed in [12], which learns a subspace where two views' instances of the same sample are similar.

*Incomplete Multimodal Visual Data Grouping (IMG).* The subspace method proposed in [14], which added a graph Laplacian term to learn a latent global graph representation and the subspace simultaneously.

*k-Means-Based Consensus Clustering (KCC).* The work in [20] proposed a unified framework for $k$-means-based consensus clustering that can handle cases with incomplete partitions. Although this work does not focus on incomplete multiview clustering specifically, if we use the clustering results of each of the views as input partitions, it can deal with incomplete multiview clustering.

*3.3. Experimental Settings.* In our experiments, the number of clusters is set as the true number of classes. Kernels are centralized and scaled during the preprocessing procedure following the suggestion put forward in [21]. Incomplete multiview data is manually produced for the complete multiview datasets. If the incomplete samples ratio (ISR) is $\epsilon$, then $\epsilon \times N$ samples are randomly selected as incomplete. We keep the probability that a view is missing set at $q_0 = 0.5$. A random vector $\mathbf{g} = (\mathbf{g}_1, \ldots, \mathbf{g}_P) \in [0,1]^P$ is generated for each incomplete sample. The $p$th view of an incomplete sample exists only if $\mathbf{g}_p > q_0$. Because at least one view should always exist for a sample, a random vector is accepted until there is one view available for this sample. $\epsilon$ is varied from 0.1 to 0.9 to produce different missing patterns. For each value of $\epsilon$, 10 random missing patterns are generated and the average performance reported. For the proposed method, the parameter $\beta$ is searched for in $[10^{-5}, 10^{-4}, \ldots, 10^4, 10^5]/P$. $P$ represents the number of views, which is divided to avoid the scale difference caused by view number. For the relatively large dataset Flower 102, we search a smaller range: $[10^{-3}, 10^{-2}, \ldots, 10^2, 10^3]/P$. For PVC, we use the code provided by the authors, and the parameter is tuned from $[10^{-6}, 10^5, \ldots, 1]$ following the suggestion in [12]. For IMG, the same parameter as in PVC is set as the tuned value in PVC, and the other two parameters are set as advised in [14]. We use normalized mutual information (NMI) as the clustering evaluation [3, 12].

*3.4. Experimental Results.* Figure 2 shows the results on 3 Sources, the real-world incomplete multiview dataset. BSV performs worse as it only considers information from one view. Using the multiview information fusion, MKKM with zero-filling reaches a better NMI than BSV, while MKKIK outperforms MKKM for a more reasonable imputation. The proposed method achieves a significant NMI boost of about 30% compared with MKKIK. Our method fills the incomplete kernels to make the clustering result of each view consistent, while MKKIK does not consider the consistency. We suggest that there may be a strong underlying consistency between views on 3 Sources, so the proposed method outperforms MKKIK in part due to the fact that

Table 1: Overview of datasets.

| Dataset | Number of samples | Number of views | Number of clusters |
|---|---|---|---|
| 3 Sources | 416 | 3 | 6 |
| Digital | 2000 | 3 | 10 |
| Flower 17 | 1360 | 7 | 17 |
| Flower 102 | 8189 | 4 | 102 |

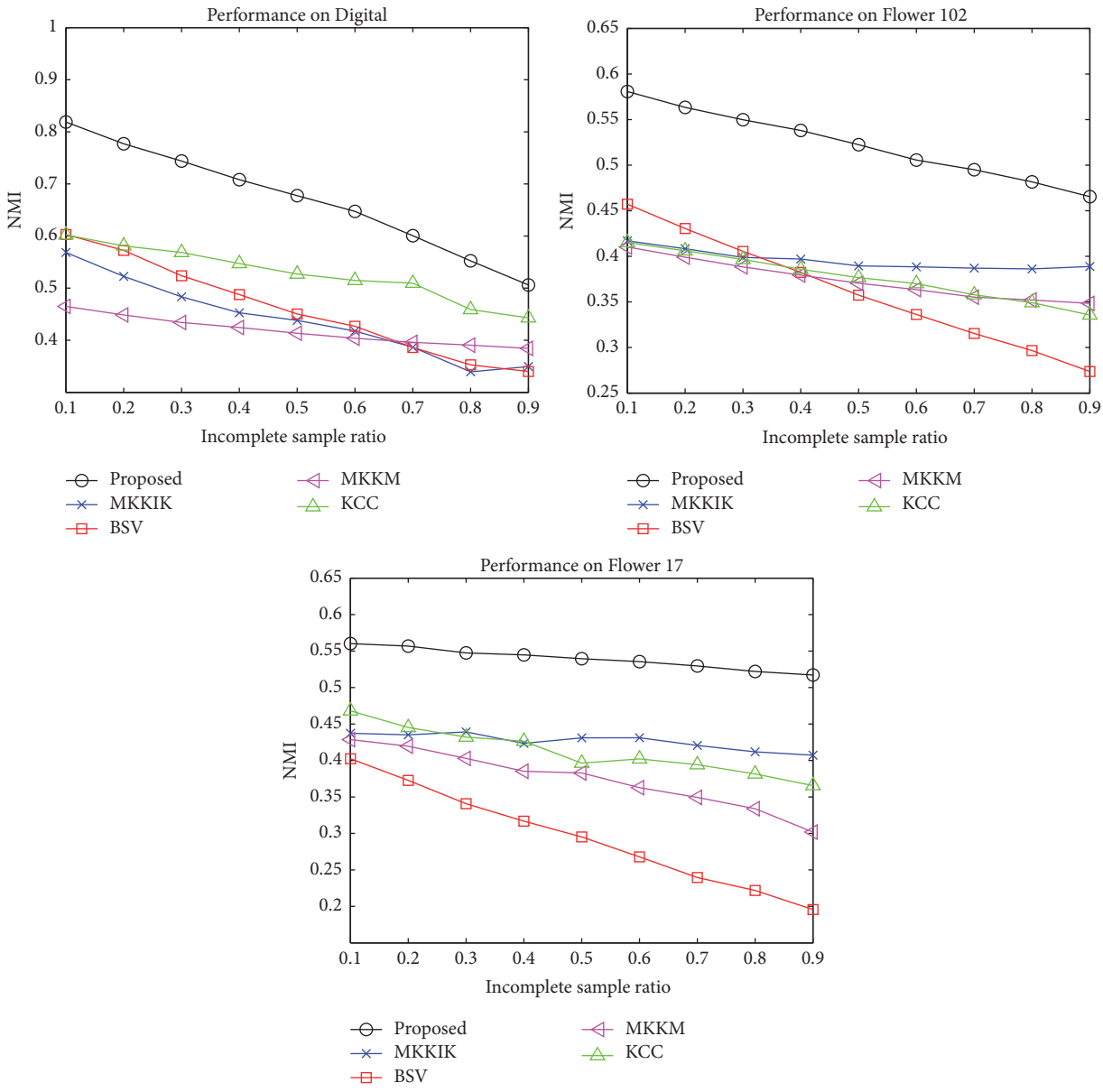


Figure 3: Performance comparison on synthetic incomplete multiview datasets in terms of NMI.

this consistency is considered. Moreover, our method also outperforms KCC, which is a method that does consider the consistency; we suggest that this occurs because KCC does not have an imputation process. In KCC, the consensus clustering decision is learned from the remaining incomplete partitions.

Figure 3 summarizes the results on the three artificial incomplete multiview datasets: Flower 17, Flower 102, and

Table 2: Details of the 3 Sources dataset.

| | Articles | Missing ratio |
|---|---|---|
| BBC | 352 | 0.1538 |
| The Guardian | 302 | 0.2740 |
| Reuters | 294 | 0.2933 |

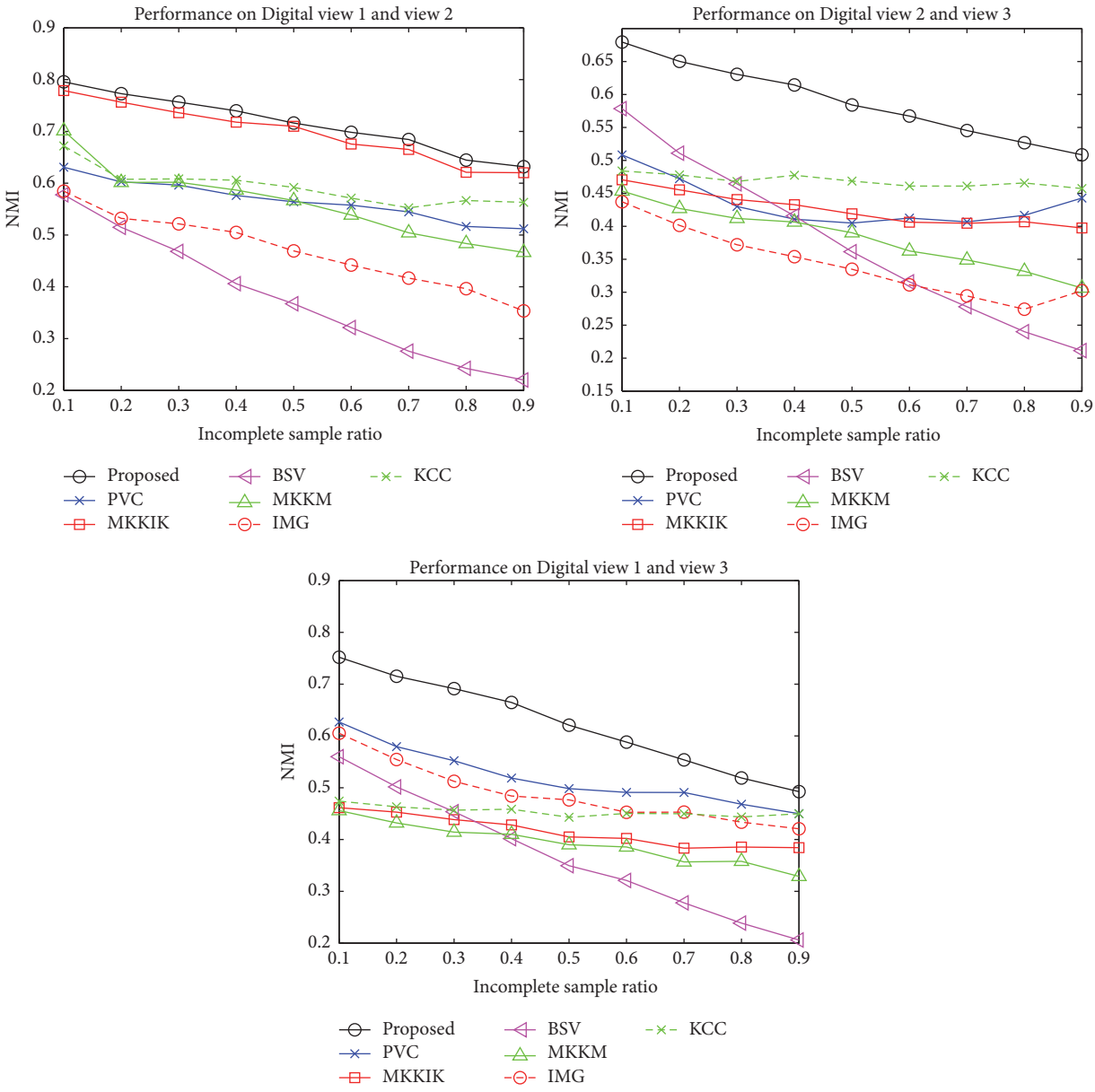


Figure 4: Performance comparisons on synthetic incomplete two-view data from Digital in terms of NMI.

Digital. It can be observed that the proposed method constantly achieves the best NMI compared with the state-of-the-art methods with different ISRs. Moreover, the proposed method significantly outperforms the second-best method with different ISRs. For example, the proposed method outperforms the second-best method by around 20% on Digital when ISR is 0.1. It is also notable that when ISR increases, the performance of all methods decreases, which validates the degenerating effect of incomplete views. In Figure 4, we compare our method with two additional

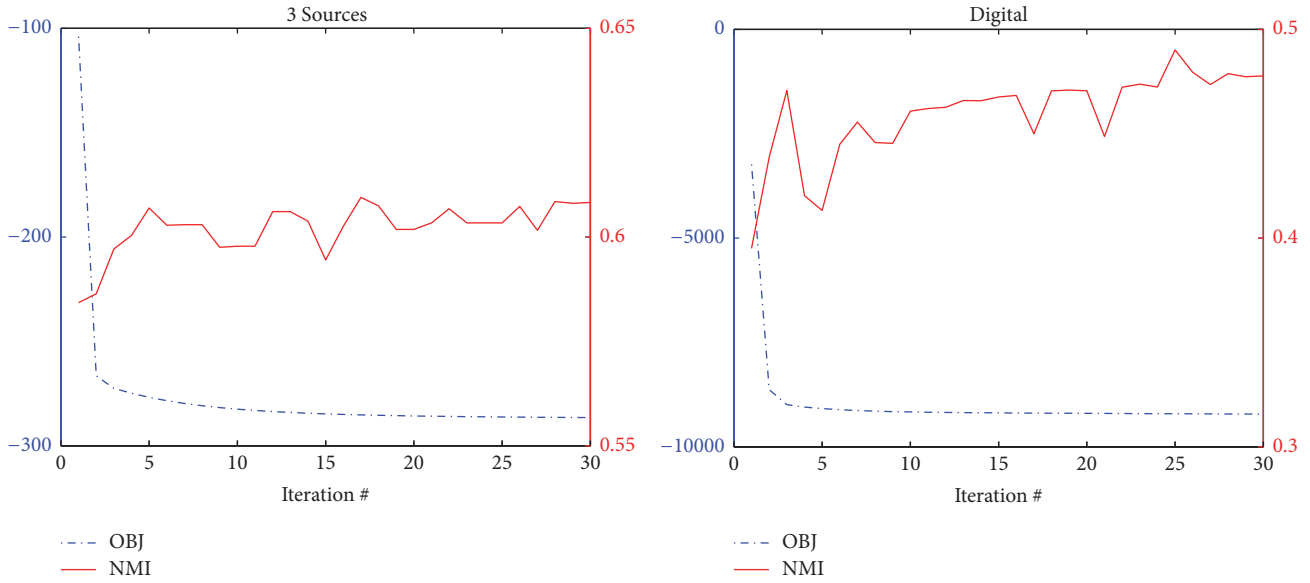


Figure 5: Objective value curve and NMI curve against iteration number.

methods, PVC and IMG, which are representative subspace methods that focus on two views. We report the results on the view pairs of Digital. The proposed method constantly exhibits better performance than all other methods on all view pairs. This shows that the proposed method can also perform better than the state-of-the-art subspace methods in a two-view situation.

To summarize, the proposed method demonstrates its superiority against the state-of-the-art methods on both synthetic and real-world multiview datasets. We suggest that the imputation in the proposed method considers both clustering performances in each view and the consistency between views, which contributes to the superiority of the proposed method.

*3.5. Convergence Study.* As was proved in the previous section, the proposed algorithm is guaranteed to be convergent. Here we empirically validate the convergence property, as illustrated in Figure 5. Due to space limitations, we show the objective curve and NMI curve when incomplete sample ratio is 0.9 for 3 Sources and Digital. The objective values decrease as the iteration number increases, and the objective values converge within 30 iterations. Although the NMIs do not grow monotonically, they achieve relatively large value when the number of iterations reaches 30.

*3.6. Parameter Study.* Figure 6 illustrates how parameter $\beta$ influences clustering performance. On Digital, Flower 102, and Flower 17, we present the performance curves for three ISRs: 0.3, 0.5, and 0.7. On 3 Sources, performance is optimal when $\beta = 10^2/P$. On Digital, the performance remains relatively stable as the parameter changes. On Flower 102, the performance maintains a relatively high level when $\beta$ is greater than $10^{-2}/P$. On Flower 17, the performance is sensitive to the parameter when $\beta$ is larger than $10/P$. Overall, across the four datasets, the performance tends to be better when $\beta$ is larger. According to (7), when $\beta$ is larger, the clustering results between views should have greater consistency. Thus, better performance when $\beta$ is larger indicates relatively strong consistency between views on these datasets. It should also be emphasized that although the performance on Flower 17 is relatively sensitive to $\beta$, Figure 3 indicates that the proposed method still outperforms other methods for worse choices of $\beta$. When applying the proposed method on other datasets, we recommend a comparatively large value of $\beta$ in cases when views share a substantial amount of common information.

## 4. Conclusion

In this paper, we have proposed a consensus kernel $k$-means clustering method for incomplete multiview data in which a consensus clustering decision and the missing parts of the incomplete kernels are learned. In this way, the imputation of incomplete kernels leads to better clustering of each view and maintains consistency between views, which benefits the final clustering decision. Comprehensive experiments validate the clustering performance improvement of the proposed method compared with state-of-the-art methods.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

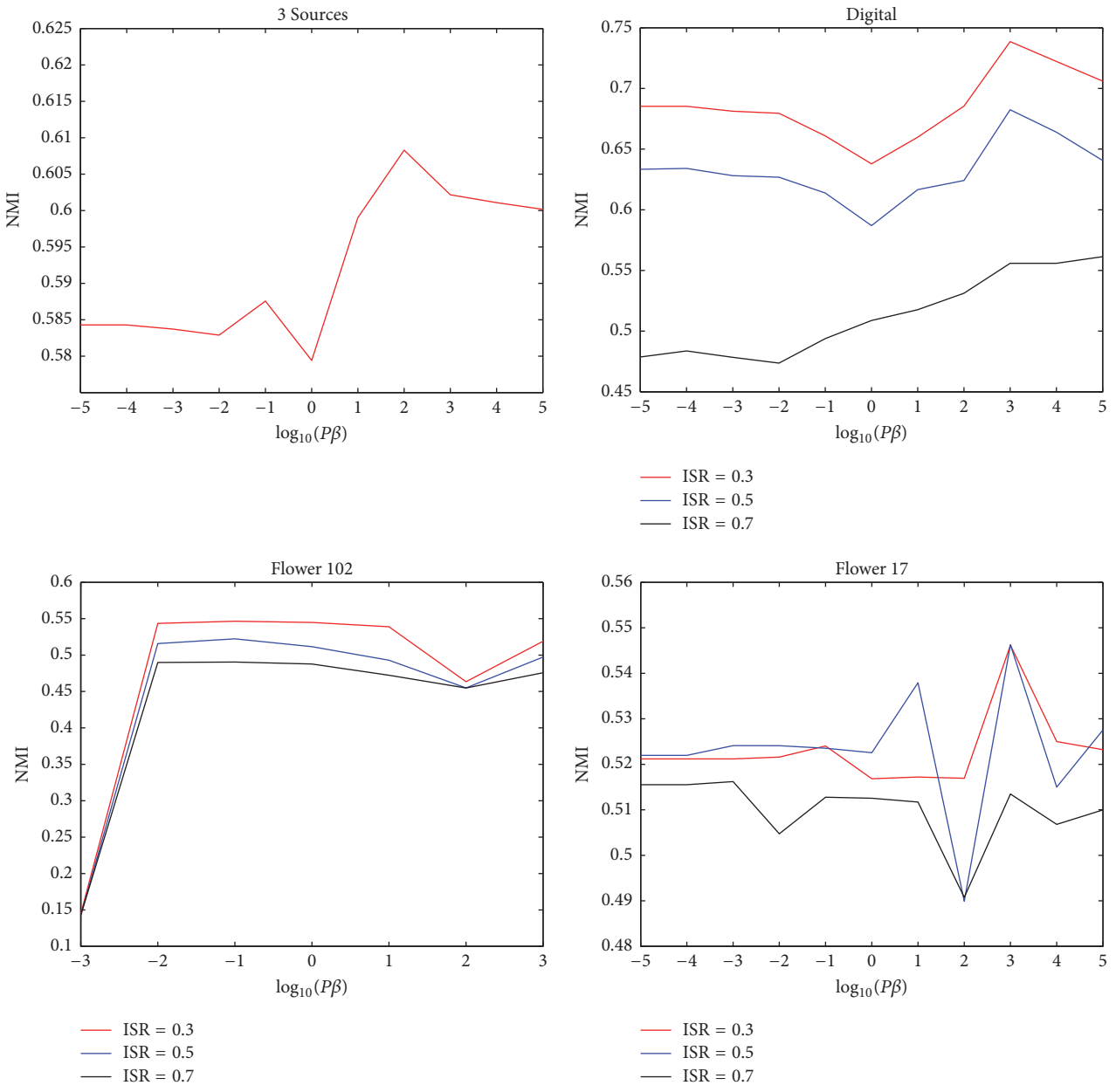


Figure 6: Parameter studies on different datasets.

## Acknowledgments

This work is supported by National Natural Science Foundation of China (no. 61672528 and no. 61403405).

## References

[1] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," 2013, https://arxiv.org/abs/1304.5634.

[2] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[3] A. Kumar, P. Rai, and H. Daumé, "Co-regularized multi-view spectral clustering," in *Advances in Neural Information Processing Systems*, pp. 1413–1421, 2011.

[4] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 252–260, 2013.

[5] Q. Yin, S. Wu, R. He, and L. Wang, "Multi-view clustering via pairwise sparse subspace representation," *Neurocomputing*, vol. 156, pp. 12–21, 2015.

[6] Q. Wang, Y. Dou, X. Liu, Q. Lv, and S. Li, "Multi-view clustering with extreme learning machine," *Neurocomputing*, vol. 214, pp. 483–494, 2016.

[7] L. Zhang and X. Hu, "Locally adaptive multiple kernel clustering," *Neurocomputing*, vol. 137, pp. 192–197, 2014.

[8] Y. Lu, L. Wang, J. Lu, J. Yang, and C. Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognition*, vol. 47, no. 11, pp. 3656–3664, 2014.

[9] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple dernel k-means clustering with matrix-induced regularization," in *Proceedings of the in Proceedings of the 30th AAAI Conference on Artifial Inteligence*, 2016.

[10] E. Bruno and S. Marchand-Maillet, "Multiview clustering: a late fusion approach using latent models," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09*, pp. 736-737, July 2009.

[11] S. F. Hussain, M. Mushtaq, and Z. Halim, "Multi-view document clustering via ensemble method," *Journal of Intelligent Information Systems*, vol. 43, no. 1, pp. 81–99, 2014.

[12] S. Li, Y. Jiang, and Z. Zhou, "Partial multi-view clustering," in *Proceedings of the in Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.

[13] Q. Yin, S. Wu, and L. Wang, "Incomplete multi-view clustering via subspace learning," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM '15*, pp. 383–392, October 2015.

[14] H. Zhao, H. Liu, and Y. Fu, "Incomplete multi-modal visual data grouping," in *Proceedings of the in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

[15] P. Rai, A. Trivedi, H. Daumé, and S. L. DuVall, "Multiview clustering with incomplete views," in *Proceedings of the NIPS Workshop on Machine Learning for Social Computing*, 2010.

[16] W. Shao, X. Shi, and P. S. Yu, "Clustering on multiple incomplete datasets via collective kernel learning," in *Proceedings of the 13th IEEE International Conference on Data Mining, ICDM '13*, pp. 1181–1186, December 2013.

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

[18] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel k-means clustering with incomplete kernels," in *Proceedings of the in AAAI Conference on Artificial Intelligence*, 2017.

[19] Y. Ye, X. Liu, J. Yin, and E. Zhu, "Co-regularized kernel *k*-means for multi-view clustering," in *Proceedings of the International Conference on Pattern Recognition*, 2016.

[20] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen, "*K*-means-based consensus clustering: a unified view," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 155–169, 2015.

[21] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, pp. 795–828, 2012.