Consensus Guided Multi-View Clustering

HONGFU LIU and YUN FU, Northeastern University

In recent decades, tremendous emerging techniques thrive the artificial intelligence field due to the increasing collected data captured from multiple sensors. These multi-view data provide more rich information than traditional single-view data. Fusing heterogeneous information for certain tasks is a core part of multi-view learning, especially for multi-view clustering. Although numerous multi-view clustering algorithms have been proposed, most scholars focus on finding the common space of different views, but unfortunately ignore the benefits from partition level by ensemble clustering. For ensemble clustering, however, there is no interaction between individual partitions from each view and the final consensus one. To fill the gap, we propose a Consensus Guided Multi-View Clustering (CMVC) framework, which incorporates the generation of basic partitions from each view and fusion of consensus clustering in an interactive way, i.e., the consensus clustering guides the generation of basic partitions, and high quality basic partitions positively contribute to the consensus clustering as well. We design a non-trivial optimization solution to formulate CMVC into two iterative *k*-means clusterings by an approximate calculation. In addition, the generalization of CMVC provides a rich feasibility for different scenarios, and the extension of CMVC with incomplete multi-view clustering further validates the effectiveness for real-world applications. Extensive experiments demonstrate the advantages of CMVC over other widely used multi-view clustering methods in terms of cluster validity, and the robustness of CMVC to some important parameters and incomplete multi-view data.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Multi-view clustering, ensemble clustering, utility function

ACM Reference format:
Hongfu Liu and Yun Fu. 2018. Consensus Guided Multi-View Clustering. *ACM Trans. Knowl. Discov. Data.* 12, 4, Article 42 (April 2018), 21 pages.
https://doi.org/10.1145/3182384

1 INTRODUCTION

Multi-view data with heterogeneous representations are becoming more and more popular in both academic and industrial areas (Bickel and Scheffer 2004; Ying et al. 2007; Chen et al. 2010; Zhang et al. 2008). For example, images can be extracted different descriptors, such as RGB values, Fourier coefficient, SIFT, HOG or deep features; the news might be reported by different media and different channels; and the literary work is broadcasted via multiple translations in different languages.

This research is supported by the NSF IIS Award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Award W911NF-17-1-0367.
Authors' addresses: H. Liu and Y. Fu, 360 Huntington Ave, Boston, MA 02115; emails: liu.hongf@husky.neu.edu, yunfu@ece.neu.edu.

https://doi.org/10.1145/3182384

These multi-view data provide more rich information to uncover the intrinsic structure than the traditional single-view data. Moreover, it has been widely recognized that the multi-view learning helps to handle outliers and noisy features for better performance (Wang et al. 2013; Sun 2013; Guo 2013).

The goal of multi-view clustering is to make use of heterogeneous information from different views to provide a comprehensive clustering result (Zhou and Liu 2008; Kim et al. 2010; Eaton et al. 2010). The key problem of multi-view clustering is how to fuse the heterogeneous information. One naive way is directly to concatenate the features from different views together and apply traditional single-view learning methods on the new representation. However, data collected from different views might have little in common within the original feature spaces. For example, the photic features and acoustic features, they share high similarity in high level for cluster structure or partition level, rather than the photic or acoustic spaces. In light of this, some scholars aim to seek a unified common space to represent multi-view data (Blaschko and Lampert 2008; Chaudhuri et al. 2009). However, it becomes more challenging when it comes to unsupervised tasks. Some views containing irrelevant or noisy representation might severely damage the common space and lead to degraded performance. Besides, with the increase of the number of views, there is little common space shared by all the views.

Ensemble clustering, also known as consensus clustering (Strehl and Ghosh 2003), aims to fuse several partitions into an integrated one. Different from clustering problem, ensemble clustering is formulated as a fusion problem in essence. In light of this, numerous methods have been proposed including graph-based methods (Strehl and Ghosh 2003; Fern and Brodley 2004), co-association matrix based (Fred and Jain 2005) and k-means-based methods (Wu et al. 2013). Especially, k-means-based Consensus Clustering (KCC) transforms the ensemble clustering problem into a k-means clustering problem and provides flexible utility functions for different scenarios (Wu et al. 2015). For a long time, ensemble clustering has not been paid much attention in the multi-view clustering area due to different research problems. Actually, there are several benefits of fusing multi-view information in partition level, such as meaningful cluster structure and robustness to outliers. However, there is no interaction between individual partitions from each view and the final consensus one.

Therefore, the existing studies on multi-view clustering either pay little attention to fusing multi-view information in partition level (Kumar et al. 2011; Liu et al. 2013), or overlook the interaction between individual partitions from each view and the final result (Wu et al. 2013). In response to this, we propose a novel framework, Consensus Guided Multi-View Clustering (CMVC) to integrate heterogeneous information and to seek a consensus partition from different views. In essence, CMVC is an extension of our previous work KCC for multi-view clustering, which also achieves the multi-view clustering in the partition space by integrating the individual basic partitions from each view. Consequently, CMVC inherits the robustness and empirical good performance of consensus clustering. However, it is a kind of waste that the high-quality consensus partition is not further utilized. In light of this, beyond fusing basic partitions, CMVC further employs the high-quality consensus partition to guide the updating of basic partitions, which later contribute to a new consensus partition. Fusing basic partitions into a consensus one and updating basic partitions with the guidance of the consensus one are iteratively optimized for multi-view clustering. In such a way, CMVC updates basic partitions and the consensus one in a joint fusion way.

Different from existing algorithms, CMVC has two advantages: (1) multi-view information is fused in partition level (in the experimental part, we showcase the large margin of the methods that fuses information in partition level over others); (2) the basic partitions and consensus clustering are iteratively updated in a mutually promotional way. Figure 1 shows the pipelines of the proposed

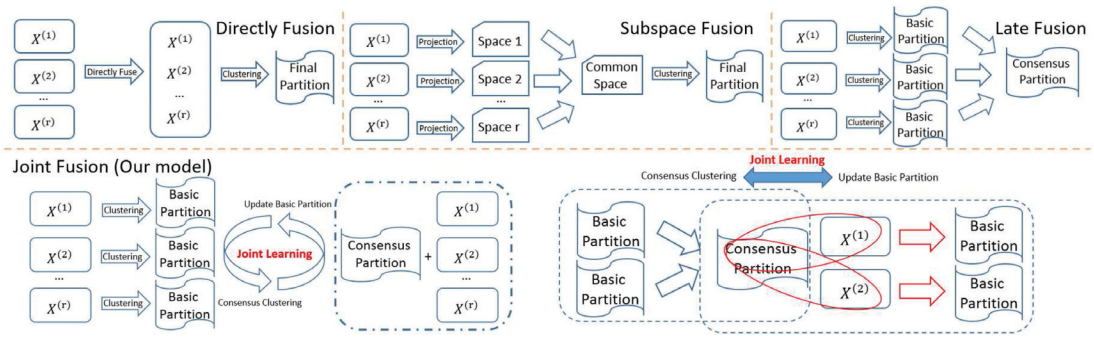


Fig. 1. Different pipelines for multi-view clustering.

CMVC and other types of multi-view clustering. In our framework, several basic partitions are generated from each view, followed by the ensemble clustering to obtain the consensus partition; the consensus partition further supervises the generation of the basic partitions. The above processes are iteratively updated for multi-view clustering. The intuition here is that consensus clustering guides the generation of basic partitions, and higher quality basic partitions positively contribute to the consensus clustering as well. Calculations are approximated to speedup and the process leads to the generalization of CMVC, indicating that complex multi-view clustering can be handled by two iterative simple k-means clusterings with high efficiency and rich feasibility. Next, we extend CMVC to handle the incomplete multi-view clustering. Extensive experiments demonstrate the advantages of CMVC over other widely used multi-view clustering methods. Beyond accurate clustering performance, CMVC is insensitive to some important parameters, such as the trade-off parameter λ, the random feature selection rate and the number of sub-views. Finally, CMVC shows appealing merits in learning from incomplete multi-view data with high robustness, which validates its effectiveness for real-world applications. Our contributions are highlighted as follows:

—We propose a novel framework CMVC to fuse heterogeneous information in partition level. Generally speaking, the high-quality consensus partition is further used to guide the generation of basic ones, which also are conductive to the consensus partition.
—By an approximate calculation, CMVC is formulated into two iterative k-means clusterings with high efficiency, which leads to the generalization with different utility and distance functions and provides rich feasibility for different scenarios.
—Experimental results demonstrate the effectiveness of joint fusion. Especially, CMVC shows appealing merits in learning from incomplete multi-view data with high robustness.

The rest of this article is organized as follows. Section 2 gives a brief related work on multi-view clustering. We illustrate the objective function and its corresponding solution in Section 3. Sections 4 and 5 provide the approximation calculation and the generalization of CMVC, respectively. In Section 6, we extend CMVC to handle incomplete multi-view clustering. Section 7 demonstrates the experimental results compared with the state-of-the-art. Finally, we conclude this article in Section 8.

2 RELATED WORK

Much progress has been made over the past decade in developing effective multi-view clustering algorithms, whose goals are to fuse multiple representations and partition instances into different clusters. Here, we summarize the existing multi-view clustering algorithms in the literature roughly into four groups. (1) The simplest way is to treat the multi-view data as the single-view

Table 1. Comparisons Among Different Multi-View Clustering Algorithms

		Features for fusion			Fusion way	
Types	Representative methods	Low level	Middle level	High level	One time fusion	Joint fusion
Directly fusion	ConKM, ConNMF	✓			✓	
Subspace fusion	CCA, PVC		✓		✓	
Late fusion	HCC, KCC			✓	✓	
Joint fusion	CRSC, Multi NMF		✓			✓
Our model	CMVC	✓		✓		✓

data by concatenating multi-view features directly and conduct the traditional clustering process via optimizing certain loss functions (Bickel and Scheffer 2004; Kumar and Daume 2011). Li et al. (2015) employed local manifold fusion to integrate heterogeneous features with approximated bipartite graphs for accelerating the speed. (2) The second category aims to project the data in different views into a common low-dimensional latent subspace with the structure preserved, then any clustering algorithms on the common space can be applied to obtain the final partition (Blaschko and Lampert 2008; Chaudhuri et al. 2009). For instance, Singh and Gordon (2008) proposed Collective NMF (ColNMF) to employ the same bases to present multi-view data; Li et al. (2014) established the latent space for partial multi-view data with the same example in different views close to each other; Cai et al. (2013) leveraged $l_{2,1}$ norm to obtain a shared indicator matrix; Ying et al. (2007) found all non-redundant clustering views of the data; in Guo (2013), the authors formulated the subspace learning of multiple views as a joint optimization problem with a group sparsity constraint. (3) The third category is named late integration or late fusion (Bruno and Marchand-Maillet 2009; Tao et al. 2017), which generates basic partitions from each view individually, and fuses them into an integrated one. Actually, the third category is the well-known ensemble clustering (Strehl and Ghosh 2003). Numerous methods have been proposed including graph-based methods (Fern and Brodley 2004; Tao et al. 2016, 2017), co-association matrix based (Fred and Jain 2005; Liu et al. 2015, 2017b) and k-means-based methods (Wu et al. 2013; Liu et al. 2015, 2016, 2018). (4) The above three categories are regarded as one-time fusion. Differently, the fourth category belongs to joint fusion. Some methods interactively learn basic indictors and the consensus subspace by making the basic ones consistent with the consensus one. In Kumar et al. (2011), within the spectral clustering framework, the authors integrated eigenvectors learnt from different views via co-regularization. Similarly, Liu et al. (2013) achieved the consistency between individual matrix factorizations and the consensus one. Other multi-view clustering methods include local learning (Zhang et al. 2008), pareto optimization (Wang et al. 2008), spectral embedding (Xia et al. 2010), and so on.

We summarize the differences among these multi-view clustering algorithms in Table 1. Compared to subspace fusion methods, the late fusion methods usually achieve better performance since the more informative partitions are used to obtain the final result. However, this kind of methods employs only one-time fusion, which means that there is no interaction between basic partitions. Recently, Co-regularized Spectral Clustering (CRSC) (Kumar et al. 2011), MultiNMF (Liu et al. 2013) alternatively fuse basic information and the consensus one; they use middle-level features such as eigen features or latent space features, which are less effective than the one of partition level. In our model, CMVC fuses multi-view information of partition level to obtain the consensus one. Then, we update basic ones with the consensus partition and the original low-level features from each view.

In essence, CMVC is an extension of KCC (Wu et al. 2015) for multi-view clustering. The major difference lies in the interaction between basic partitions and the consensus one. Although partition-level fusion brings high-quality performance, it is a pity if we cannot further employ the

Table 2. Contingency Matrix

π \ $\pi^{(i)}$	$C_1^{(i)}$	$C_2^{(i)}$	$\cdots$	$C_K^{(i)}$	$\sum$
C_1	$n_{11}^{(i)}$	$n_{12}^{(i)}$	$\cdots$	$n_{1K}^{(i)}$	n_{1+}
C_2	$n_{21}^{(i)}$	$n_{22}^{(i)}$	$\cdots$	$n_{2K}^{(i)}$	n_{2+}
$\cdot$	$\cdot$	$\cdot$	$\cdots$	$\cdot$	$\cdot$
C_K	$n_{K1}^{(i)}$	$n_{K2}^{(i)}$	$\cdots$	$n_{KK}^{(i)}$	n_{K+}
$\sum$	$n_{+1}^{(i)}$	$n_{+2}^{(i)}$	$\cdots$	$n_{+K}^{(i)}$	n

consensus one to obtain better results. That is one of our motivations in this article. In CMVC, the consensus partition and basic ones are updated in a mutually promotional way.

3 CONSENSUS GUIDED MULTI-VIEW CLUSTERING

We first present the objective function of CMVC and give the corresponding solution and approximate calculation. Then, the generalization of CMVC with different distance functions and utility functions is derived, and finally we apply CMVC to handle the incomplete multi-view clustering.

3.1 Objective Function

Let $X = \{X^{(1)}, X^{(2)}, \ldots, X^{(r)}\}$ be the data with r multiple representations or views, and $X^{(v)} = \{x_1^{(v)}, x_2^{(v)}, \ldots, x_n^{(v)}\}$ denote n instances in the vth view. The objective function of *CMVC* is as follows:

$$\min_{H^*} \sum_{v=1}^{r} ||X^{(v)} - H^{(v)}C^{(v)}||_{\mathrm{F}}^2 - \lambda \sum_{v=1}^{r} U_c(H^*, H^{(v)}), \tag{1}$$

where $H^{(v)}$ is the cluster assignment matrix derived from $X^{(v)}$, $C^{(v)}$ is the corresponding centroid matrix, H^* is the final consensus cluster indicator matrix, and λ is a tradeoff parameter between standard k-means and disagreement with the consensus clustering. U_c is the categorical utility function (Mirkin 2001) measuring the similarity between two partitions, which is widely used in ensemble clustering (Wu et al. 2015; Liu et al. 2017b) and constrained clustering (Liu and Fu 2015; Liu et al. 2017a).

To better understand U_c, we next introduce the contingency matrix, which counts the co-occurrence for two discrete random variables. Table 2 shows a typical example for two partitions, π and $\pi^{(i)}$ containing K clusters. In the table, $n_{kj}^{(i)}$ denotes the number of data objects belonging to both cluster $C_j^{(i)}$ in $\pi^{(i)}$ and cluster C_k in π, $n_{k+} = \sum_{j=1}^{K} n_{kj}^{(i)}$, and $n_{+j}^{(i)} = \sum_{k=1}^{K} n_{kj}^{(i)}$, $1 \le j, k \le K$. Let $p_{kj}^{(i)} = n_{kj}^{(i)}/n$, $p_{k+} = n_{k+}/n$, and $p_{+j}^{(i)} = n_{+j}^{(i)}/n$. Based on the variables in Table 2, we have the following equation for U_c:

$$U_c(H^*, H^{(v)}) = \sum_{k=1}^{K} p_{k+} \sum_{j=1}^{K} \left(\frac{p_{kj}^{(v)}}{p_{k+}} \right)^2 - \sum_{j=1}^{K} \left(p_{+j}^{(v)} \right)^2, \tag{2}$$

where $p_{kj}^{(v)}$ is the joint probability of one instance belonging to both the kth cluster in H^* and the jth cluster in $H^{(v)}$, p_{k+} and $p_{+j}^{(v)}$ are the portion of the kth cluster in H^* and the jth cluster in $H^{(v)}$, respectively.

The objective function incorporates the generation of basic partitions and fusion of consensus clustering into a one-step framework. Basic partitions from different views and the consensus clustering are conductive to the generation of each other. Therefore, the objective function has two benefits: (1) multi-view information is fused in partition level; (2) the consensus clustering guides the generation of basic partitions as side information, further high-quality basic partitions positively contribute to the consensus clustering as well.

3.2 Solutions

To solve the optimization problem in Equation (1), we propose an iterative update procedure. Generally speaking, we first apply k-means on each individual view to generate $H^{(v)}$, $1 \le v \le r$, and then the two following steps are repeated until convergence: (1) fixing $H^{(v)}$, update H^*; (2) fixing H^*, update each $H^{(v)}$.

Fixing $H^{(v)}$, Update H^*: When all $H^{(v)}$ are fixed, the optimization problem becomes a consensus clustering problem. Here, we introduce Theorem 3.1 to transfer the ensemble clustering into a k-means clustering problem.

THEOREM 3.1. *Given r basic indicator matrix $H^{(1)}, \ldots, H^{(r)}$, and let $H = [H^{(1)}, \ldots, H^{(r)}]$ be the $n \times rK$ matrix concatenating all the indicator matrices from each view, we have*

$$\sum_{v=1}^{r} U_c(H^*, H^{(v)}) \propto -||H - H^*G||_{\mathrm{F}}^2, \tag{3}$$

where $G = [G^{(1)}, G^{(2)}, \ldots, G^{(r)}]$ is the centroid of H.

The proof can be found in our previous work (Wu et al. 2013, 2015).

Remark 1. Theorem 3.1 converts the complex consensus clustering problem into a simple k-means clustering with squared Euclidean distance, which has the neat formulation. Moreover, Theorem 3.1 also gives a new insight into the objective function of *CMVC* as follows:

$$\min_{H^*} \sum_{v=1}^{r} ||X^{(v)} - H^{(v)}C^{(v)}||_{\mathrm{F}}^2 + \lambda||H^{(v)} - H^*G^{(v)}||_{\mathrm{F}}^2. \tag{4}$$

Beyond the utility function measuring the similarity in the partition level, we can also use the distance to measure the disagreement and form it into the k-means framework. It is worthy to note that different from the loss function in Liu et al. (2013), we have one more variable $G^{(v)}$, which is learnable and plays a role in shuffling the order of clusters in H^*.

Fixing H^*, Update $H^{(v)}$: According to Equation (4), the optimization problem has the following format with fixed H^*:

$$\min_{H^{(v)}} ||X^{(v)} - H^{(v)}C^{(v)}||_{\mathrm{F}}^2 + \lambda||H^{(v)} - H^*G^{(v)}||_{\mathrm{F}}^2. \tag{5}$$

At this point, we can also iteratively update unknown variables $C^{(v)}$, $G^{(v)}$, and $H^{(v)}$ by three subproblems.

(1) When $C^{(v)}$ and $H^{(v)}$ are fixed, we only care about the term that is relevant to $G^{(v)}$ and minimize $J_1 = ||H^{(v)} - H^*G^{(v)}||_{\mathrm{F}}^2$, we have

$$J_1 = \mathrm{tr}((H^{(v)} - H^*G^{(v)})(H^{(v)} - H^*G^{(v)})^\top). \tag{6}$$

Next, we take the derivative of J_2 over $G(v)$, and have

$$\frac{\partial J_1}{\partial G^{(v)}} = -2H^{*\top}H^{(v)} + 2H^{*\top}H^*G^{(v)} = 0. \tag{7}$$

ALGORITHM 1: The Algorithm of *CMVC*

Input: $X^{(1)}, X^{(2)}, \ldots, X^{(r)}$: data matrices for r views;
K: number of clusters;
λ: tradeoff parameter.
Output: optimal H^*;
1: Initialize $H^{(v)}$ by running k-means on $X^{(v)}$;
2: **repeat**
3: let $H = [H^{(1)}, H^{(2)}, \ldots, H^{(r)}]$;
4: Run k-means on H to update H^* by Equation (3);
5: For each view, update $H^{(v)}$ by Algorithm 2;
6: **until** the objective value of Equation (1) remains unchanged.

ALGORITHM 2: Update $H^{(v)}$ with Fixed H^*

Input: $X^{(v)}$: the vth view data matrix;
H^*: consensus clustering;
K: number of clusters;
λ: tradeoff parameter.
Output: optimal $H^{(v)}$;
1: **repeat**
2: Update $G^{(v)}$ by Equation (8);
3: Update $H^{(v)}$ by Equation (9);
4: Update $C^{(v)}$ by Equation (12);
5: **until** the objective value of Equation (5) remains unchanged.

The solution leads to the update rule of $G^{(v)}$ as follows:

$$G^{(v)} = (H^{*\top} H^*)^{-1} H^{*\top} H^{(v)}. \tag{8}$$

(2) When $C^{(v)}$ and $G^{(v)}$ are fixed, the derivative method is not suitable for the binary variable $H^{(v)}$; thus, we exhaustively calculate the distance between each instance and centers, then find the label that makes the objective function minimized:

$$k = \arg\min_j ||X_i^{(v)} - C_j^{(v)}||_2^2 + \lambda ||z_j - H_i^* G^{(v)}||_2^2, \tag{9}$$

where $X_i^{(v)}$ is the ith instance in view $X^{(v)}$, $C_j^{(v)}$ and H_i^* are the jth row and ith row in $C^{(v)}$ and H^*, respectively, and z_j is a $1 \times K$ vector with 1 in the jth position and 0 in other places.

(3) When $G^{(v)}$ and $H^{(v)}$ are fixed, let $J_2 = ||X^{(v)} - H^{(v)} C^{(v)}||_{\mathrm{F}}^2$, we have

$$J_2 = \mathrm{tr}((X^{(v)} - H^{(v)} C^{(v)})(X^{(v)} - H^{(v)} C^{(v)})^\top), \tag{10}$$

where $\mathrm{tr}(\cdot)$ means the trace of a matrix. Then, by taking derivative of $C^{(v)}$, we get

$$\frac{\partial J_2}{\partial C^{(v)}} = -2H^{(v)\top} X^{(v)} + 2H^{(v)\top} H^{(v)} C^{(v)}. \tag{11}$$

Setting Equation (11) to be 0, we can update $C^{(v)}$ as follows:

$$C^{(v)} = (H^{(v)\top} H^{(v)})^{-1} H^{(v)\top} X^{(v)}. \tag{12}$$

By these two steps, we alternatively update H^* and $H^{(v)}$ and repeat the process until the objective function value converges. We summarize the algorithm of CMVC in Algorithm 1. In the iterative fashion, the basic partitions are fused for the robust consensus partition H^*; then, the consensus partition is involved to guide the generation of basic partitions, which successively contribute to a new consensus partition. Therefore, basic partitions from different views and the consensus clustering are conductive to the generation of each other.

3.3 Convergence Analysis and Discussion

From the solutions, we can see that H^* and $H^{(v)}$ are iteratively updated. When $H^{(v)}$ is fixed, we transfer the optimization problem over H^* into a k-means clustering, which has a good convergence property. And, given fixed H^*, we decompose the optimization problem over $H^{(v)}$ into three subproblems, and each of them is a convex problem with respect to one variable. Therefore, our proposed algorithm guarantees that CMVC can converge to a local optimum.

4 APPROXIMATE CALCULATION

From the above solution, the first step employs k-means to optimize H^* in an efficient way. However, in the second step, a lot of heavy matrix computation, including multiplication and inverse, are needed to update $H^{(v)}$. Thus, we wonder if we could also employ k-means to approximately calculate $H^{(v)}$. Here, we use the following equation to substitute Equation (5) for an efficient solution:

$$\min_{H^{(v)}} ||X^{(v)} - H^{(v)}C^{(v)}||_{\mathrm{F}}^2 + \lambda||H^* - H^{(v)}G^{(v')}||_{\mathrm{F}}^2. \tag{13}$$

Compared to Equation (5), in Equation (13), just the second term has been changed. Note that $||H^* - H^{(v)}G^{(v')}||_F^2$ is just $U_c(H^{(v)}, H^*)$. Although $U_c(H^{(v)}, H^*) \neq U_c(H^*, H^{(v)})$, here, we replace $U_c(H^*, H^{(v)})$ by $U_c(H^{(v)}, H^*)$ due to the fact that both have the same function measuring the similarity between H^* and $H^{(v)}$. In addition, Lemma 4.1 demonstrates that $U_c(H^{(v)}, H^*)$ and $U_c(H^*, H^{(v)})$ share the consistent order in the solution space.

Lemma 4.1. *Given three partitions H^*, H_1 and H_2, if $U_c(H^*, H_1) \gtreqless U_c(H^*, H_2)$, then we have $U_c(H_1, H^*) \gtreqless U_c(H_2, H^*)$.*

Proof. It is self-evident that

$$\max_{H^*} U_c(H^*, H_1) \Leftrightarrow \max_{H^*} U_c(H_1, H^*), \tag{14}$$

holds for any feasible region F. Therefore, if let $F = \{H', H''\}$, we have

$$U_c(H', H_1) \gtreqless U_c(H'', H_2) \Leftrightarrow U_c(H_1, H') \gtreqless U_c(H_2, H''), \tag{15}$$

which indicates that $U_c(H^*, H_1)$ and $U_c(H_1, H^*)$ have the same ranking over all the possible partitions in the universal set F. We complete the proof. □

By Lemma 4.1, we have the following theorem to update $H^{(v)}$ in a fast way.

Theorem 4.2. *Let $X^{(v)}$ be the matrix of the vth data, H^* be the consensus clustering matrix, and $D^{(v)}$ is a concatenated matrix consisting of $X^{(v)}$ and H^*, then we have*

$$\min_{H^{(v)}} ||X^{(v)} - H^{(v)}C^{(v)}||_{\mathrm{F}}^2 + \lambda||H^* - H^{(v)}G^{(v')}||_{\mathrm{F}}^2 \Leftrightarrow \min_{H^{(v)}} \sum_{k=1}^{K} \sum_{i \in \mathbf{C}_k} f(d_i, m_k), \tag{16}$$

ALGORITHM 3: The Algorithm of Generalized CMVC

Input: $X^{(1)}, X^{(2)}, \ldots, X^{(r)}$: data matrices for r views;
K: number of clusters;
λ: tradeoff parameter.
Output: optimal H^*;
1: Initialize $H^{(v)}$ by k-means on $X^{(v)}$ with the distance function in Equation (20);
2: **repeat**
3: let $H = [H^{(1)}, H^{(2)}, \ldots, H^{(r)}]$;
4: Run k-means on H to update H^* with the distance function in Equation (21);
5: For each view, update $H^{(v)}$ by k-means with distance function in Equation (22);
6: **until** H^* remains unchanged.

where $\mathbf{C}_k$ *is the* k*th cluster in* $H^{(v)}$*, and* m_k *is its corresponding centroid,* d_i *is the* i*th row in* $D^{(v)}$*, and* f *is a k-means distance, which can be computed as follows:*

$$f(d_i, m_k) = ||d_{i,1} - m_{k,1}||_2^2 + \lambda ||d_{i,2}^{(v)} - m_{k,2}||_2^2, \tag{17}$$

where $d_i = \langle d_{i,1}, d_{i,2} \rangle$ *with* $d_{i,1} = \langle d_{i,11}, \ldots, d_{i,1m} \rangle$ *and* $d_{i,1} = \langle d_{i,1m+1}, \ldots, d_{i,1m+K} \rangle$*, and* $m_{k,1}$*,* $m_{k,2}$ *have the similar definitions.*

Proof. We start from the right side of the objective function of k-means:

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{i \in \mathbf{C}_k} f(d_i, m_k) \\
&= \sum_{k=1}^{K} \sum_{i \in \mathbf{C}_k} ||d_{i,1} - m_{k,1}||_2^2 + \lambda ||d_{i,2} - m_{k,2}||_2^2 \\
&= \sum_{k=1}^{K} \sum_{i \in \mathbf{C}_k} ||d_{i,1} - m_{k,1}||_2^2 + \lambda \sum_{k=1}^{K} \sum_{i \in \mathbf{C}_k} ||d_{i,2} - m_{k,2}||_2^2 \\
&= ||X^{(v)} - H^{(v)} C^{(v)}||_{\mathrm{F}}^2 + \lambda ||H^* - H^{(v)} G^{(v')}||_{\mathrm{F}}^2.
\end{aligned} \tag{18}$$

We finish the proof. □

Remark 2. Theorem 4.2 provides a way to transfer the optimization problem of Equation (13) into a k-means clustering. By taking a close look, we can see that the k-means distance function is nothing but the weighted squared Euclidean distance. When $\lambda = 1$, the weighed squared Euclidean distance degenerates into the traditional Euclidean distance.

Remark 3. The goal of the objective function in Theorem 4.2 not only aims to uncover the cluster structure in the vth view, but also makes use of H^* to guide the clustering process. Different from the traditional side information, which applies the pairwise constraints for clustering, the guidance is directly employed on the partition level, rather than the instance level. Such kind of partition level side information (Liu and Fu 2015; Liu et al. 2017a) provides more consistency than pairwise constraints and leads to better performance.

Benefits of the approximate calculation lie in three points. (1) The optimization problem over $H^{(v)}$ can be solved by a k-means clustering with high efficiency. (2) *CMVC* can be solved via two

Table 3. Sample Instances of the Point-to-Centroid Distance

Distance	$\phi(x)$	$f(x, y)$
Squared Euclidean distance	$\|\|x\|\|_2^2$	$\|\|x - y\|\|_2^2$
KL-divergence	$-H(x)$	$\sum_{j=1}^{d} x_j \log \frac{x_j}{y_j}$
Cosine distance	$\|\|x\|\|_2$	$\|\|x\|\|_2 - \sum_{j=1}^{d} x_j y_j / \|\|y\|\|_2$

Note: H means Shannon entropy.

iterative k-means clusterings, which indicates that such complex multi-view clustering can be handled by the simplest clustering algorithm in a neat mathematical way. (3) The integrated k-means framework leads to a good generalization with different k-means distances and utility functions. Although it is difficult to strictly prove the convergency of the CMVC with the approximate version, it has fast converge speed in practice.

Next, we provide the time complexity for our proposed CMVC, which consists of two iterative phases, consensus partition fusion and basic partition updating. For the consensus partition fusion, KCC is employed to transform it into a k-means clustering on the concatenated basic partitions $H = [H^{(1)}, \ldots, H^{(r)}]$, which leads the time complexity for this phase $O(t_1 nK^2 r)$. Here, t_1 is the average iteration number. Recall that H is a binary matrix and only r elements in each row are non-zero. With a matrix indexing the non-zero elements, the time complexity drops to $O(t_1 nKr)$. For the basic partition updating, we still formulate it as a k-means clustering on $D^{(v)}$ and the time complexity is $O(t_2 nK(m_v + K))$, where t_1 is the average iteration number and m_v is the number of features for the v-th view data matrix. Thus, the overall time complexity for CMVC is $O(q(t_1 nKr + t_2 nK(m + rK)))$, where $m = \sum_{i=1}^{r} m_v$ and it is linear to the number of instances and features, which could be a candidate tool for large-scale multi-view data clustering.

5 GENERALIZATION OF CMVC

So far we use squared Euclidean distance or Frobenius norm to derive the objective function of CMVC. In practice, squared Euclidean distance is not powerful enough to capture the complex structure of multi-view data. In this section, we demonstrate that there exist rich distance functions and utility functions, which can be involved in the CMVC framework. Before giving the generalization of CMVC, we introduce the Point-to-Centroid Distance (P2C-D) (Wu et al. 2012), which is an extension of Bergman divergence (Banerjee et al. 2005).

Definition 1. Let $S \in \mathbb{R}$ be a non-empty open convex set. A twice continuously differentiable function $f: S \times S \to \mathbb{R}_+$ is called P2C-D, if there exists some higher order continuously differentiable convex function $\phi : S \to \mathbb{R}$ such that

$$f(x, y) = \phi(x) - \phi(y) - (x - y)^\top \nabla\phi(y). \tag{19}$$

It has been shown that Bregman divergence (Banerjee et al. 2005) as a family of distances fits the classic k-means with arithmetic centroids, which makes the objective function value decrease in the iterative instance assignment and centroid update. If we relax the requirement of ϕ in Bregman divergence from the strictness of the convexity to the continuously differentiable convexity, this leads to the more general P2C-D (Wu et al. 2012), which also guarantees the convergency of k-means-like algorithms. Table 3 gives some examples of the P2C-D. It is worth noting that cosine similarity is a widely used metric in high-dimensional clustering. However, it cannot be generalized by Bregman divergence.

Table 4. Sample KCC Utility Functions

	$U(H^*, H^{(v)})$	$\phi(m_k)$	$f(h_l, m_k)$
U_C	$\sum_{k=1}^{K} p^*_{k+}\|\|P_k^{(v)}\|\|_2^2 - \|\|P^{(v)}\|\|_2^2$	$\sum_{i=1}^{r} \|\|m_{k,i}\|\|_2^2 - \|\|P^{(v)}\|\|_2^2$	$\sum_{i=1}^{r} \|\|h_{l,i} - m_{k,i}\|\|_2^2$
U_H	$\sum_{k=1}^{K}(-H(P_k^{(v)})) - (H(P^{(v)}))$	$\sum_{i=1}^{r}(-H(m_{k,i})) - (H(P^{(v)}))$	$\sum_{i=1}^{r} D(h_{l,i}\|\|m_{k,i})$
$U_{\cos}$	$\sum_{k=1}^{K} p^*_{k+}\|\|P_k^{(v)}\|\|_2 - \|\|P^{(v)}\|\|_2$	$\sum_{i=1}^{r} \|\|m_{k,i}\|\|_2 - \|\|P^{(v)}\|\|_2$	$\sum_{i=1}^{r}(1 - \cos(h_{l,i}, m_{k,i}))$

Note: D means KL-divergence, cos means cosine similarity, $m_{k,v} = P_k^{(v)} = \langle p_{k1}/p^*_{k+}, \ldots, p_{kk}/p^*_{k+}\rangle$, and $P^{(v)}$ is the cluster distribution of $H^{(v)}$.

Based on the P2C-D, we studied the KCC (Wu et al. 2015) and derived a necessary and sufficient condition for KCC utility functions. Note that the category utility function U_c is just a special case of KCC utility functions. By giving different convex functions ϕ, a rich group of KCC utility functions can be derived. Table 4 shows some examples of KCC utility functions derived from various ϕ. We can see that ϕ is just the summation of several convex functions. This indicates that the rich k-means distance functions can derive from not only various convex functions, but also the combination of different convex functions.

By this means, a rich group of objective functions of CMVC can be formatted with different convex functions. Here, let f_1 be the k-means distance during the generation basic partition with ϕ_1, f_2 with ϕ_2 be the k-means distance when optimizing H^* and f_3 can be used to update $H^{(v)}$. Then, we have the following distances:

$$f_1(x, y) = \phi_1(x) - \phi_1(y) - (x - y)^\top \nabla\phi_1(y), \tag{20}$$

$$f_2(x, y) = \sum_{i=1}^{r}(\phi_2(x_i) - \phi_2(y_i) - (x_i - y_i)^\top \nabla\phi_2(y_i)), \tag{21}$$

$$f_3(x, y) = \phi_1(x_1) - \phi_1(y_1) - (x_1 - y_1)^\top \nabla\phi_1(y_1) + \lambda(\phi_2(x_2) - \phi_2(y_2) - (x_2 - y_2)^\top \nabla\phi_2(y_2)). \tag{22}$$

In Equations (21) and (22), a vector is decomposed into several blocks, on each block one k-means distance function is used to calculate the part of distance, the final result can be obtained by the summation or linear combination. In the generalization of CMVC, we use f_1 to initialize $H^{(v)}$, apply f_2 to get the consensus clustering H^*, and employ f_3 to approximately optimize $H^{(v)}$. The algorithm of generalized CMVC is given by Algorithm 3.

It is also worth noting that the generalization of CMVC provides new insights to solve some complex objective functions. For the objective function in Equation (1), we can use the gradient method to solve it; however, if we use cosine similarity or KL-divergence as the distance function or utility function, the objective function cannot be summarized in the matrix formulation. That means these complex objective functions cannot be solved by Algorithms 1 and 2.

6 HANDLING INCOMPLETE MULTI-VIEW DATA

In real-world scenarios, it is common that multi-view data could be corrupted by faulty device or transmission loss. This results in incomplete multi-view data with missing feature values of some instances. While much efforts have been taken in the studies of the multi-view clustering problem, handling incomplete multi-view data is an underlying problem, i.e., when the data from one view or more views are inaccessible. In light of this, we extend CMVC to handle this challenging, i.e., the Incomplete Multi-view Clustering (IMC) problem (Zhao et al. 2016).

To solve the IMC problem, one naive way is to remove these data points with missing values. However, it is quite inappropriate to shrink the size of training samples due to a few missing entries, which wastes the rich information from other views. Another natural way is to fill in the

missing elements by the average value or the value from nearest neighbors, or the value from a predict model (Williams and Carin 2015; Bhadra et al. 2017). Although these strategies facilitate the missing issue to some extent, these new artificial points would disturb the original feature space and lead to a skewed cluster structure with large missing ratio (Shao et al. 2015). Back to *CMVC*, the missing values affect the updating of H^* and basic partition $H^{(v)}$. In the following, we handle the missing values in terms of updating H^* and $H^{(v)}$, respectively.

Let $X_s^{(v)}$ with $n_s^{(v)}$ instances ($n_s^{(v)} \leq n$) be a subset of $X^{(v)}$. We first initialize $H^{(v)}$ for each view. Due to $n_s^{(v)} \leq n$, missing values result in missing labels in $H^{(v)}$. When updating H^*, these missing labels do not provide any utility for consensus. That means these missing labels do not contribute the centroid matrix G in Equation (3). Therefore, the centroids are no more the arithmetic average of assigned elements of $H = [H^{(1)}, \ldots, H^{(r)}]$, which has the following computation formulation:

$$g_k = \frac{\sum_{i \in \mathbf{C}_k \cap X_s^{(v)}} h_i}{|\mathbf{C}_k \cap X_s^{(v)}|}, \tag{23}$$

where h_i is the ith row of H.

When updating $H^{(v)}$, Theorem 4.2 concatenates the data from the vth view and the consensus partition H^* as a whole matrix. The corresponding centroids consist of two parts $m_k = \langle m_{k,1}, m_{k,2} \rangle$. Similar to Equation (23), the missing values in the vth view are not involved in the computation of $m_{k,1}$. Note that there is no missing label in H^*. Equation (24) gives the updating rules of centroids with missing values:

$$m_{k,1} = \frac{\sum_{i \in \mathbf{C}_k \cap X_s^{(v)}} d_{i,1}}{|\mathbf{C}_k \cap X_s^{(v)}|}, m_{k,2} = \frac{\sum_{i \in \mathbf{C}_k} d_{i,2}}{|\mathbf{C}_k|}. \tag{24}$$

By this means, we can still make use of k-means to update H^* and $H^{(v)}$ with the above centroids updating rules and modified k-means distance. In the phase of initialization, the missing instances are labeled as a vector with all zeros in $H^{(v)}$, and the rest instances are organized for clustering with labels from 1 to K.

In the phase of consensus, we adjust the distance function of f_2 as follows:

$$f_2(x, y) = \sum_{v=1}^{r} \mathbf{1}\left(x_i \in X_s^{(v)}\right) f'(x_i, y_i), \tag{25}$$

where $f'(x_i, y_i) = \phi_2(x_i) - \phi_2(y_i) - (x_i - y_i)^T \nabla \phi_2(y_i)$ and $\mathbf{1}(x_i \in X_s^{(v)}) = 1$ if x_i is in $X_s^{(v)}$ and 0 otherwise. By this means, the centroids are also updated by arithmetic mean, with the denominator representing the number of non-missing instances.

Similarly, in the phase of updating individual partitions, the distance function of f_3 can be adjusted as follows:

$$f_3(x, y) = \mathbf{1}\left(x_i \in X_s^{(v)}\right) f_1(x_1, y_1) + \lambda f'(x_2, y_2). \tag{26}$$

Note that there exist missing elements in $H^{(v)}$, but each instance in H^* has a label from 1 to K. For the incomplete multi-view CMVC, we have the following theorem.

Theorem 1. *The incomplete multi-view clustering problems with objective function Equations (4) and (13) can be guaranteed to converge in finite two-phase iterations with the distance function in Equations (25) and (26) and the centroid updating rules in Equations (23) and (24), respectively.*

Proof. Here, we give the proof when updating $H^{(v)}$. The proof for updating H^* has the similar procedure. Starting from the objective function of k-means, we have

$$\sum_{k=1}^{K}\sum_{i\in \mathbf{C}_k} f_3(d_i, m_k) = \sum_{k=1}^{K}\Bigg(\sum_{i\in \mathbf{C}_k\cap X_s^{(v)}} (\phi_1(d_{i,1}) - \phi_1(m_{k,1}) - (d_{i,1} - m_{k,1})^\top \nabla\phi_1(m_{k,1})) + \sum_{i\in \mathbf{C}_k} (\phi_2(d_{i,2}) - \phi_2(m_{k,2}) - (d_{i,2} - m_{k,2})^\top \nabla\phi_2(m_{k,2}))\Bigg). \tag{27}$$

The k-means includes two iterations of assignment phase and updating centroids. In the assignment phase, each instance is assigned to the nearest centroid so that the objective function decreases. Thus, we analyze the change of the objective function value during updating centroids with missing values. Next, we prove that the computation of centroid by Equation (24) is the optimal.

Let $q_k = \langle q_{k,1}, q_{k,2}\rangle$ be any centroid of the concatenated matrix $D^{(v)}$ consisting of $X^{(v)}$ and H^*. Then, we calculate the difference of objective function value difference with the centroids q_k and m_k:

$$\begin{aligned}
\Delta &= \sum_{k=1}^{K}\sum_{i\in \mathbf{C}_k}\Big(f_3(d_i, q_k) - f_3(d_i, m_k)\Big)\\
&= \sum_{k=1}^{K}\Bigg(\sum_{i\in \mathbf{C}_k\cap X_s^{(v)}} (\phi_1(d_{i,1}) - \phi_1(q_{k,1}) - (d_{i,1} - q_{k,1})^\top \nabla\phi_1(q_{k,1}) - \phi_1(d_{i,1})\\
&\quad + \phi_1(m_{k,1}) + (d_{i,1} - m_{k,1})^\top \nabla\phi_1(m_{k,1})) + \sum_{i\in \mathbf{C}_k} (\phi_2(d_{i,2}) - \phi_2(q_{k,2})\\
&\quad -(d_{i,2} - q_{k,2})^\top \nabla\phi_2(q_{k,2}) - \phi_2(d_{i,2}) + \phi_2(m_{k,2}) + (d_{i,2} - m_{k,2})^\top \nabla\phi_2(m_{k,2}))\Bigg).
\end{aligned}$$

According to Equation (24), we have $\sum_{i\in \mathbf{C}_k\cap X_s^{(v)}} d_{i,1} = |\mathbf{C}_k \cap X_s^{(v)}| m_{k,1}$ and $\sum_{i\in \mathbf{C}_k} d_{i,2} = |\mathbf{C}_k| m_{k,2}$. Therefore, the above equation can be simplified as

$$\begin{aligned}
\Delta &= \sum_{k=1}^{K}\Bigg(\sum_{i\in \mathbf{C}_k\cap X_s^{(v)}} (\phi_1(m_{k,1}) - \phi_1(q_{k,1}) - (d_{i,1} - q_{k,1})^\top \nabla\phi_1(q_{k,1}))\\
&\quad + \sum_{i\in \mathbf{C}_k} (\phi_2(m_{k,2}) - \phi_2(q_{k,2}) - (d_{i,2} - q_{k,2})^\top \nabla\phi_2(q_{k,2}))\Bigg)\\
&= \sum_{k=1}^{K}\Big(|\mathbf{C}_k \cap X_s^{(v)}| f_1(m_{k,1}, q_{k,1}) + |\mathbf{C}_k| f_2(m_{k,2}, q_{k,2})\Big).
\end{aligned} \tag{28}$$

Since f_1 and f_2 are two distance functions, we have $\Delta \geq 0$ and the updating rule in Equation (24) is optimal. Due to the limited solution space, the modified k-means converges in finite iterations and we finish the proof. □

The incomplete multi-view problem is a more general scenario in essence. As aforementioned, we can also employ k-means for incomplete multi-view clustering, although the computation of

Table 5. Experimental Datasets

Datasets	#instance	#cluster	#view	#features
Digit	2,000	10	2	240, 74
3-Sources	169	9	3	3,560, 3,631, 3,068
Multilingual	600	6	3	9,749, 9,109, 7,774
4-Areas	4,236	4	2	20, 13,214
Caltech101	2,386	20	6	48, 40, 254, 1,984, 512, 928
BBCSport	544	5	2	3,138, 3,203

some variables is slightly different. Actually, the weights of the views containing missing value will naturally decrease according to the missing ratio when generating the consensus H^*.

7 EXPERIMENTAL RESULTS

In this section, we evaluate CMVC on six broadly used multi-view datasets, including two image datasets, three text datasets, and one academic data set. We first demonstrate the effectiveness of CMVC compared with several state-of-the-art methods, and then explore the major impact factors of CMVC; finally, the results on incomplete multi-view data validate the robustness of CMVC.

7.1 Experimental Setup

Datasets: We summarize six real-world datasets in the experiments in Table 5. UCI Handwritten Digit[1]: This image data set includes 0–9 handwritten digits, where gray level values and Fourier coefficients are extracted as two views. 3-Sources[2]: This is an online news text datasets coming from BBC, Guardian and Reuter from February to April 2009. Three media are represented as three views. Multilingual[3]: Three different languages in the text data are regarded as three views. Here, we follow Liu et al. (2013) and use a subset with 600 instances. 4-Areas[4]: This academic dataset has two views, the conference view and abstract term view, which consists of 20 conferences in four areas with 28,702 authors and 13,214 terms in the abstract. Each author is labeled to one or multiple areas. After removing cross-domain authors, we have 4,236 authors with the conference view and term view. Caltech101 (Li et al. 2007) provides an image set of 101 categories for the object recognition task. By following Li et al. (2015), we use a 20-class subset, which consists of 2,836 images and encodes each image with six different features. BBCSports[5]: The news article dataset contains 737 news articles from the BBC Sport website corresponding to five sport topics from 2004 to 2005, such as athletics, cricket, football, rugby, and tennis. In the experiment, we use a subset of this dataset provided by Xia et al. (2014).

Baseline algorithms: Here, we compare our CMVC method with a number of baseline algorithms including two baseline methods with concatenated features, four multi-view clustering methods, and one ensemble clustering method. ConKM runs standard k-means on the concatenated representation from different views. ConNMF works on the concatenated data to obtain the new latent representation via NMF. ColNMF treats the multi-view data with a shared coefficient matrix with different basis matrices across views (Singh and Gordon 2008). CRSC integrates eigenvectors learnt from different views via co-regularization with in the spectral clustering framework (Kumar et al.

[1] http://archive.ics.uci.edu/ml/datasets.html.
[2] http://mlg.ucd.ie/datasets.
[3] http://www.webis.de/research/corpora.
[4] http://www.ccs.neu.edu/home/yzsun/data/four_area.zip.
[5] http://mlg.ucd.ie/datasets.

2011). MultiNMF achieves the consistency between individual matrix factorizations and the consensus one (Liu et al. 2013). Partial Multi-View Clustering (PVC) deals with incomplete two-view clustering based on NMF (Li et al. 2014), which learns the common space based on the condition of missing data in two views. KCC is a kind of late fusion method (Wu et al. 2015), which first generates basic clustering solution from each view and then fuses them into a consensus one via k-means. Different utility functions correspond to different k-means distances.

Tools: CMVC is completely developed in MATLAB by the authors. CMVC_{sqE} denotes that we run k-means with squared Euclidean distance to generate basic partitions for each view and merge these basic partitions with categorical utility function, and CMVC_{KL} applies KL-divergence to cluster the data from each view and leverages mutual information as the utility function to optimize the ensemble process, and CMVC_{cos} uses cosine similarity as k-means distance and utility function. KCC_{sqE}, KCC_{KL}, and KCC_{cos} have the similar meanings with different utility functions. All the comparative methods are provided by the corresponding authors. To achieve best performance of KCC, here, we make use of sub-view setting. That means for each view, 10 sub-views are generated with $r_s = 50\%$ feature sample rate in order to enrich the diversity. We apply the sub-view setting for KCC and CMVC, and set $\lambda = 0.01$ the same as (Liu et al. 2013) and Kumar et al. (2011). Each algorithm is repeated 10 times and the average results and standard deviations are reported. We set K for each algorithm to be the true cluster number for fair comparison.

Metric: Due to the availability of label information, we employ the widely used external measure NMI and R_n for cluster validity (Wu et al. 2009), which have the following formulations:

$$NMI = \frac{\sum_{i,j} n_{ij} \log \frac{n \cdot n_{ij}}{n_{i+} \cdot n_{+j}}}{\sqrt{(\sum_i n_{i+} \log \frac{n_{i+}}{n})(\sum_j n_{j+} \log \frac{n_{+j}}{n})}}, \tag{29}$$

$$R_n = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \sum_i \binom{n_{i+}}{2} \sum_j \binom{n_{+j}}{2} / \binom{n}{2}}{\sum_i \binom{n_{i+}}{2}/2 + \sum_j \binom{n_{+j}}{2}/2 - \sum_i \binom{n_{i+}}{2} \sum_j \binom{n_{+j}}{2} / \binom{n}{2}}, \tag{30}$$

where Table 2 contains all the variables of the above equation and helps to better understand the meaning of NMI and R_n. Note that both of these two metrics are positive measurements with large values indicating better performance. However, sometimes R_n might take negative values, which means the cluster result is worse than random guess.

7.2 Clustering Results Comparisons

Table 6 shows the clustering performance on six real-world datasets in terms of NMI and Rn, where the best results are highlighted in bold and "–" denotes that PCV cannot handle more than two-view data. Three observations are very clear. First, multi-view clustering methods are generally better than the methods with concatenated features. However, on 4-Areas and BBCSport datasets, CRSC and PCV have extremely worse performance, which indicates that these two methods struggle to handle high-dimensional datasets. ColNMF and MultiNMF also suffer from high deviations on this data set as well. Second, CMVC and KCC have substantial improvements than other methods, which demonstrates the benefits of fusing high-level information. Compared to KCC, which belongs to a single direction fusion, CMVC employs joint fusion to update basic partitions and the consensus one and gets slightly higher performance than KCC (see Figure 3), which validates the effectiveness of our proposed method. That means when a high quality clustering result is obtained, we further make full use of the high-quality partition to guide the basic ones and improve the performance. Third, CMVC achieves additional merits by providing a flexible framework that can incorporate various distances and utility functions for different

Table 6. Clustering Performance on Six Multi-view Datesets via *NMI* and *Rn* (%)

Datasets	Digit	3-Sources	Multilingual	4-Areas	Caltech101	BBCSport
	NMI					
ConKM	71.92 ± 2.96	41.28 ± 6.19	32.38 ± 6.06	8.46 ± 0.98	37.28 ± 1.17	48.59 ± 21.97
ConNMFConNMF	62.15 ± 2.26	45.36 ± 3.07	28.27 ± 1.47	17.87 ± 14.17	37.76 ± 1.00	42.56 ± 1.81
ColNMF	67.77 ± 4.49	51.95 ± 0.00	34.10 ± 2.79	17.86 ± 13.54	39.10 ± 0.09	24.77 ± 0.00
CRSC	73.02 ± 0.62	50.95 ± 0.79	33.67 ± 0.97	0.39 ±0.02	56.51 ± 0.29	21.60 ± 2.55
MultiNMF	76.63 ± 1.84	43.97 ± 4.13	31.46 ± 2.16	21.45 ± 17.60	58.89 ± 1.39	20.53 ± 1.41
PCV	66.79 ± 0.00	–	–	0.73 ± 0.00	–	5.68 ± 0.00
KCC_{sqE}	82.22 ± 3.26	34.18 ± 2.68	32.89 ± 3.02	41.24 ± 9.42	63.46 ± 0.47	85.23 ± 0.88
KCC_{KL}	73.94 ± 1.86	50.39 ± 3.02	27.46 ± 3.86	43.89 ± 5.12	55.56 ± 1.29	81.91 ± 4.92
KCC_{cos}	81.41 ± 1.71	69.12 ± 5.59	40.28 ± 2.26	50.25 ± 9.68	61.50 ± 1.21	88.00 ± 0.22
$CMVC_{sqE}$	**84.53 ± 1.10**	36.81 ± 4.35	34.03 ± 3.03	36.57 ± 5.54	**64.32 ± 1.65**	85.51 ± 0.52
$CMVC_{KL}$	75.01 ± 2.33	53.10 ± 1.75	27.79 ± 3.99	58.03 ± 5.63	56.32 ± 0.16	85.84 ± 3.05
$CMVC_{cos}$	82.34 ± 2.28	**72.75 ± 4.36**	**41.08 ± 2.42**	**64.08 ± 7.40**	62.20 ± 0.83	**88.11 ± 0.22**
	Rn					
ConKM	58.54 ± 6.16	15.56 ± 8.30	12.31 ± 4.12	0.01 ± 0.11	19.80 ± 1.09	34.04 ± 25.89
ConNMF	49.15 ± 6.47	27.60 ± 9.40	22.07 ± 1.58	1.80 ± 5.94	21.61 ± 2.24	29.40 ± 1.41
ColNMF	39.14 ± 2.63	20.23 ± 5.23	21.91 ± 0.79	11.41 ± 13.96	21.90 ± 2.01	19.26 ± 1.33
CRSC	64.44 ± 3.15	29.66 ± 4.31	24.16 ± 1.23	-0.05 ± 0.02	28.26 ± 2.33	11.02 ± 2.34
MultiNMF	65.22 ± 2.88	21.54 ± 5.79	22.31 ± 2.39	0.10 ± 0.10	31.61 ± 3.80	12.69 ± 0.22
PCV	55.50 ± 0.00	–	–	0.60 ± 0.00	–	0.50 ± 0.00
KCC_{sqE}	77.02 ± 5.15	60.90 ± 7.74	27.28 ± 1.53	52.57 ± 1.14	33.16 ± 5.57	86.90 ± 0.98
KCC_{KL}	67.24 ± 4.75	57.52 ± 5.80	27.62 ± 2.32	74.09 ± 6.42	33.28 ± 4.67	82.29 ± 7.92
KCC_{cos}	82.56 ± 5.29	63.18 ± 8.63	35.24 ± 2.84	75.63 ± 0.07	31.56 ± 2.58	89.51 ± 0.19
$CMVC_{sqE}$	73.34 ± 5.96	65.72 ± 6.98	28.76 ± 1.01	53.64 ± 1.62	**35.63 ± 5.57**	87.17 ± 0.95
$CMVC_{KL}$	70.46 ± 2.14	61.13 ± 3.13	29.77 ± 1.42	**78.59 ± 3.05**	35.16 ± 4.24	86.83 ± 3.37
$CMVC_{cos}$	**85.43 ± 3.91**	**69.28 ± 2.44**	**37.22 ± 0.51**	75.65 ± 0.06	33.76 ± 2.01	**89.60 ± 0.23**

Note: KCC and CMVC are used in sub-view setting.

applications. For instance, $CMVC_{cos}$ gets an excellent result on 3-Source with $NMI = 0.73$; however, the clustering results drop sharply to $NMI < 0.55$ via $CMVC_{sqE}$ and $CMVC_{KL}$. Although it is difficult to choose the best objective function of CMVC for a given data set, here, we recommend the cosine similarity due to its high quality and stability. In the following experiments, we use $CMVC_{cos}$ as default to conduct parameter analysis.

7.3 Inside Factors of CMVC

In this subsection, we systematically explore the important impact factors of CMVC for practical use in terms of the number of sub-view setting, continuous iteration, parameter analysis, and random feature selection rate.

Performance without sub-views: So far, we demonstrate the performance of CMVC with the sub-view setting. Someone might argue that the high performance of CMVC results from the random feature selection and diverse sub-views. Here, we show the performance of CMVC and KCC without sub-views in Figure 2. We can see that CMVC still performs the best among these multi-view clustering algorithms. As for KCC, it drops dramatically on Digit. Although CMVC and KCC both contain the same consensus part, the interaction mechanism between basic partitions and consensus one boosts the performance of CMVC, which shows the superiority of CMVC over other multi-view clustering algorithms.

Performance during iterations: Figure 3 shows the increasing performance of CMVC within iterations on Digits with all features, which also verifies CMVC can further improve the consensus

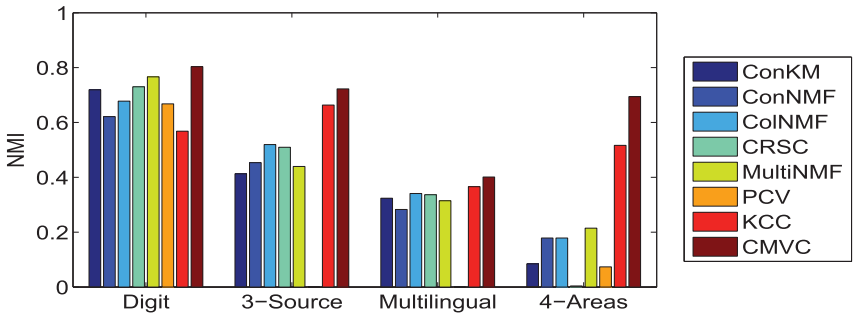


Fig. 2. Performance without sub-views setting.

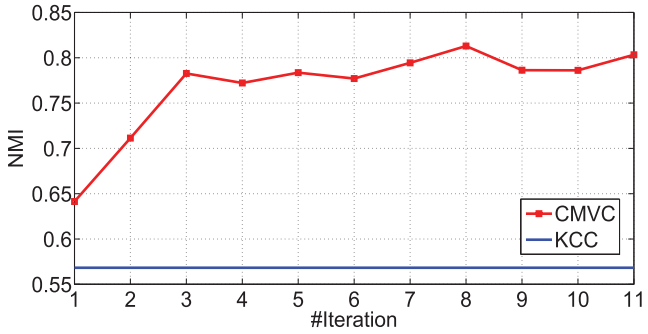


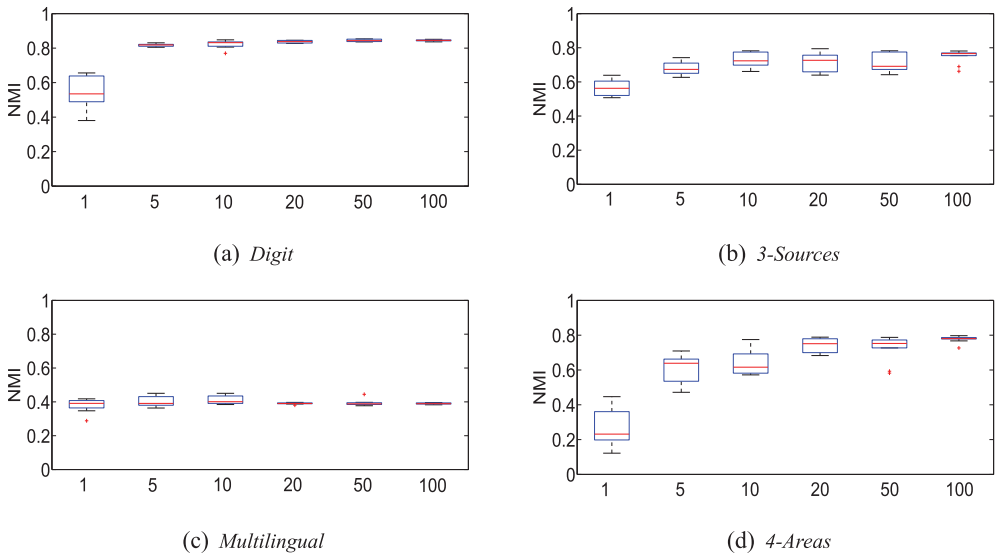
Fig. 3. Performance of CMVC during iterations on Digit.

result by mutual interaction (Figure 6). The high-quality consensus partition derived from basic partitions further guides the update of basic partitions, which not only uncover the cluster structure within each view, but also make them consistent with the consensus one as much as possible. These updated basic partitions are fused into a new consensus one. That is the reason why CMVC can get better performance than KCC with the iterative fashion.

Impact of random feature selection rate: Next, we explore the percent of random feature selection. To generate sub-view data, we conduct random feature selection with certain percent varying from 10% to 90% with 20% as interval. Figure 6 shows the results of CMVC with different feature selection rates. We can see that by increasing the random feature selection rate, the performance of CMVC gets subtle improvement on 4-Areas, and on the other datasets the results keep stable. Besides, on Digit, 3-Sources and 4-Areas, CMVC even achieves satisfactory performance with only 10% features. If we further take the efficiency and space issues into consideration, the 50% strategy would be more appealing due to enough features to generate good partitions and high diversity for fusing process.

Impact of the number of sub-views: Finally, the study of the impact of the number of sub-views is illustrated in Figure 4. If we only generate one partition for each view, CMVC not only produces bad results but also suffers from huge volatility. With the increasing of the number of sub-views, the performance of CMVC goes up and the interval of volatility narrows down as well. Therefore, the number of sub-views is a key factor to control the stability of CMVC. Here, we set the number to be 10 as default and get satisfactory results.

Impact of λ: Here, we study the impact of λ on CMVC. In the sub-view setting, we tune λ from 10^{-5}, 10^{-4}, ..., to 10^{5} to see the trend of performance. From Figure 5, we can see that CMVC achieves stably good performance on all datasets. This indicates that sub-view setting enhances the robustness of CMVC and makes it insensitive to λ.



(a) *Digit* (b) *3-Sources*

(c) *Multilingual* (d) *4-Areas*

Fig. 4. Impact of the number of sub-views.

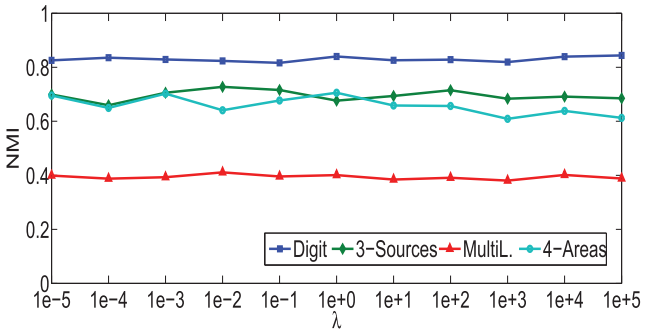


Fig. 5. Parameter analysis on λ.

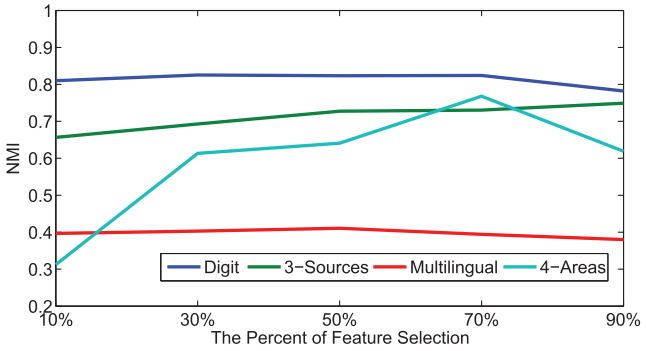


Fig. 6. Performance with different sampling ratios.

7.4 Incomplete Multi-View Clustering

Here, we validate the performance of CMVC on incomplete multi-view data, which mean there exist missing data in some views. To simulate the incomplete view setting, we randomly select a fraction of instances from each view as missing data from 5% to 50% with 5% as interval. Among

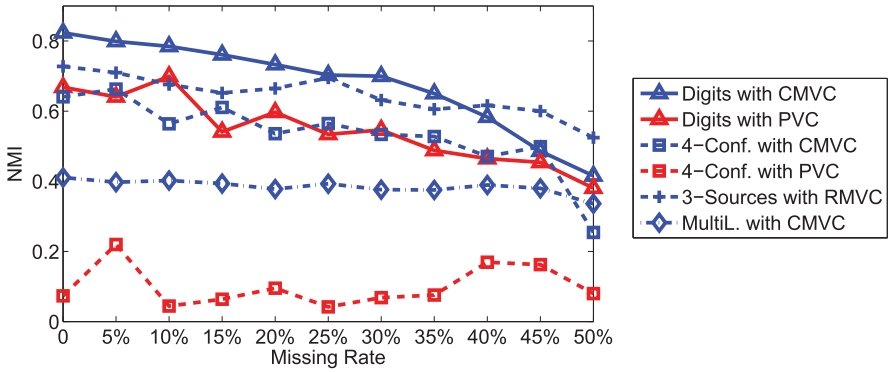


Fig. 7. Performance on incomplete multi-view data.

the competitive methods, only PVC can handle incomplete multi-view clustering, therefore, we only report the results of PVC and CMVC in Figure 7. Note that PVC cannot handle more than two-view data clustering; therefore, only results on 2-view datasets are reported. From Figure 7, it can be seen that the performance of CMVC and PVC goes down as the missing rate increases. Compared with 2-view datasets, CMVC achieves more stable results on 3-view datasets. This is because 3-view datasets provide more information with the same missing rate on each view. Moreover, CMVC outperforms PVC on all scenarios with different missing rates on Digit and 4-Areas by a large margin. Besides, it is obvious that CMVC keeps high stability with a slow decreasing rate on these four datasets. In summary, CMVC shows its robustness in handling incomplete multi-view clustering, which validates its effectiveness for real-world applications.

8 CONCLUSIONS

In this article, we proposed the CMVC framework, which incorporated the generation of basic partitions and fusion of consensus clustering into one integrated framework. Different from the existing work, the multi-view clustering was achieved in the partition space derived from each individual view. Moreover, the consensus multi-view partition further guided the updating of basic ones. By this means, basic partitions and consensus clustering were iteratively updated in a mutually promotional way. Based on this, approximate calculation was employed to solve CMVC by two iterative k-means optimization problems. Besides, it leaded to the generalization of CMVC and gave the rich diversity to different applications. Further, we extended CMVC to handle incomplete multi-view data. Experiments on real-world multi-view datasets demonstrated the effectiveness of CMVC compared with several state-of-the-art multi-view clustering algorithms. Some important impact factors of CMVC were thoroughly explored as well. In the future, we will employ the proposed method for real-world large-scale multi-view clustering.

REFERENCES

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. 2005. Clustering with Bregman divergences. *Journal of Machine Learning Research* 6 (2005), 1705–1749.

S. Bhadra, S. Kaski, and J. Rousu. 2017. Multi-view kernel completion. *Machine Learning* 106, 5 (2017), 713–739.

S. Bickel and T. Scheffer. 2004. Multi-view clustering. In *Proceedings of the International Conference on Data Mining.*

M. B. Blaschko and C. H. Lampert. 2008. Correlational spectral clustering. In *Proceedings of Computer Vision and Pattern Recoginition.*

E. Bruno and S. Marchand-Maillet. 2009. Multiview clustering: A late fusion approach using latent models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.*

X. Cai, F. Nie, and H. Huang. 2013. Multi-view k-means clustering on big data. In *Proceedings of International Joint Conference on Artificial Intelligence.*

K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. 2009. Multi-view clustering via canonical correlation analysis. In *Proceedings of International Conference on Machine Learning*.

N. Chen, J. Zhu, and E. P. Xing. 2010. Predictive subspace learning for multi-view data: A large margin approach. In *Proceedings of Advances in Neural Information Processing Systems*.

E. Eaton, M. Desjardins, and S. Jacob. 2010. Multi-view clustering with constraint propagation for learning with an incomplete mapping between views. In *Proceedings of the ACM International Conference on Information and Knowledge Management*.

X. Z. Fern and C. E. Brodley. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the International Conference on Machine Learning*.

A. L. Fred and A. K. Jain. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 6 (2005), 835–850.

Y. Guo. 2013. Convex subspace representation learning from multi-view data. In *Proceedings of AAAI Conference on Artificial Intelligence*.

Y. M. Kim, M. R. Amini, C. Goutte, and P. Gallinari. 2010. Multi-view clustering of multilingual documents. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

A. Kumar and H. Daume. 2011. A co-training approach for multi-view spectral clustering. In *Proceedings of International Conference on Machine Learning*.

A. Kumar, P. Rai, and H. Daume. 2011. Co-regularized multi-view spectral clustering. In *Proceedings of Advances in Neural Information Processing Systems*.

F. Li, R. Fergus, and P. Perona. 2007. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision Image Understanding* 106, 1 (2007), 59–70.

S. Li, Y. Jiang, and Z. Zhou. 2014. Partial multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Y. Li, F. Nie, H. Huang, and J. Huang. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2750–2756.

H. Liu and Y. Fu. 2015. Clustering with partition level side information. In *Proceedings of the International Conference on Data Mining*.

H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu. 2015. Spectral ensemble clustering. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

H. Liu, M. Shao, S. Li, and Y. Fu. 2016. Infinite ensemble for image clustering. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

H. Liu, M. Shao, S. Li, and Y. Fu. 2018. Infinite ensemble clustering. *Data Mining and Knowledge Discovery* 32, 2 (2018), 385–416.

H. Liu, Z. Tao, and Y. Fu. 2017a. Partition level constrained clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu. 2017b. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. *IEEE Transactions on Knowledge and Data Engineering* 29, 5 (2017), 1129–1143.

H. Liu, J. Wu, D. Tao, Y. Zhang, and Y. Fu. 2015. DIAS: A disassemble-assemble framework for highly sparse text clustering. In *Proceedings of the SIAM International Conference on Data Mining*.

J. Liu, C. Wang, J. Gao, and J. Han. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the SIAM International Conference on Data Mining*.

B. Mirkin. 2001. Reinterpreting the category utility function. *Machine Learning* 45, 2 (2001), 219–228.

W. Shao, L. He, and Y. Philip. 2015. Multiple incomplete views clustering via weighted nonnegative matrix factorization with L2,1 regularization. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

A. P. Singh and G. J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

A. Strehl and J. Ghosh. 2003. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3 (2003), 587–617.

S. Sun. 2013. A survey of multi-view machine learning. *Neural Computing and Applications* 23, 7 (2013), 2031–2038.

Z. Tao, H. Liu, and Y. Fu. 2017. Simultaneous clustering and ensemble. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu. 2017. From ensemble clustering to multi-view clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Z. Tao, H. Liu, S. Li, and Y. Fu. 2016. Robust spectral ensemble clustering. In *Proceedings of the ACM International Conference on Information and Knowledge Management*.

H. Wang, F. Nie, and H. Huang. 2013. Multi-view clustering and feature learning via structured sparsity. In *Proceedings of the International Conference on Machine Learning*.

X. Wang, B. Qian, J. Ye, and I. Davidson. 2008. Multi-objective multi-view spectral clustering via pareto optimization. In *Proceedings of the SIAM International Conference on Data Mining*.

D. Williams and L. Carin. 2015. Analytical kernel matrix completion with incomplete multi-view data. In *Proceedings of ICML Workshop on Learning with Multiple Views*.

J. Wu, H. Liu, H. Xiong, and J. Cao. 2013. A theoretic framework of K-means-based consensus clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen. 2015. K-means-based consensus clustering: A unified view. *IEEE Transactions on Knowledge and Data Engineering* 27, 1 (2015), 155–169.

J. Wu, H. Xiong, and J. Chen. 2009. Adapting the right measures for k-means clustering. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

J. Wu, H. Xiong, C. Liu, and J. Chen. 2012. A generalization of distance functions for fuzzy-means clustering with centroids of arithmetic means. *IEEE Transactions on Fuzzy Systems* 20, 3 (2012), 557–571.

R. Xia, Y. Pan, L. Du, and J. Yin. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2149–2155.

T. Xia, D. Tao, T. Mei, and Y. Zhang. 2010. Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 40, 6 (2010), 1438–1446.

C. Ying, X. Z. Fern, and G. D. Jennifer. 2007. Non-redundant multi-view clustering via orthogonalization. In *Proceedings of the International Conference on Data Mining*.

D. Zhang, F. Wang, C. Zhang, and T. Li. 2008. Multi-view local learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

H. Zhao, H. Liu, and Y. Fu. 2016. Incomplete multi-modal visual data grouping. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

H. Zhou and Y. Liu. 2008. Accurate integration of multi-view range images using k-means clustering. *Pattern Recognition* 41, 1 (2008), 152–175.

Received August 2016; revised January 2018; accepted January 2018