Incomplete Multi-view Clustering via Subspace Learning

Qiyue Yin, Shu Wu, Liang Wang
Center for Research on Intelligent Perception and Computing (CRIPAC)
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{qyyin, shu.wu, wangliang}@nlpr.ia.ac.cn

ABSTRACT

Multi-view clustering, which explores complementary information between multiple distinct feature sets for better clustering, has a wide range of applications, e.g., knowledge management and information retrieval. Traditional multi-view clustering methods usually assume that all examples have complete feature sets. However, in real applications, it is often the case that some examples lose some feature sets, **which results in incomplete multi-view data** and notable performance degeneration. In this paper, a novel incomplete multi-view clustering method is therefore developed, which learns unified latent representations and projection matrices for the incomplete multi-view data. To approximate the high level scaled indicator matrix defined to represent class label matrix, the latent representations are expected to be non-negative and column orthogonal. Besides, since data are often with high dimensional and noisy features, the projection matrices are enforced to be sparse so as to select relevant features when learning the latent space. Furthermore, the inter-view and intra-view data structure is preserved to further enhance the clustering performance. To these ends, an objective is developed with efficient optimization strategy and convergence analysis. Extensive experiments demonstrate that our model performs better than the state-of-the-art multi-view clustering methods in various settings.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

Keywords

Multi-view clustering; Incomplete multi-view data; Feature selection; Subspace learning; Graph regularization

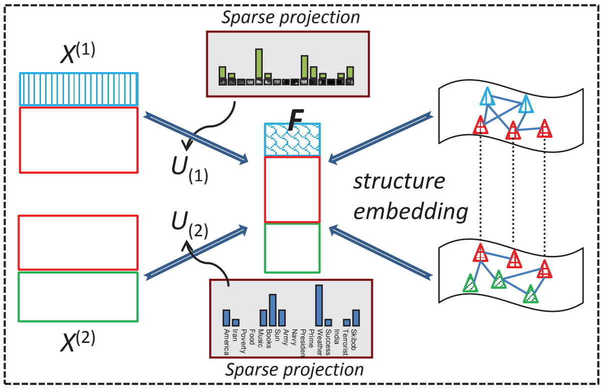

Figure 1: The overview of the proposed method. X_1 and X_2 are incomplete multi-view data with the red rectangular indicating examples with complete feature sets. We learn projection matrices $U_{(1)}$ and $U_{(2)}$ and the unified latent representations F (non-negative and column orthogonal) for multi-view data. Here we constrain the projection matrix to be sparse so as to select relevant features from the possibly high dimensional and noisy feature sets. Besides, the inter-view and intra-view data structure is preserved when learning the latent space. Finally, the clustering results are obtained by performing the k-means algorithm on F.

1. INTRODUCTION

Various kinds of real-world data appear in multiple modalities or come from multiple channels. For example, a web page can be described by both images and texts, and an image can be encoded by different visual features such as SIFT and GIST. We call such data multi-view data with each view representing a type of feature set. Usually, multiple views provide complementary information for the semantically same data, which leads to the development of multi-view learning. By exploiting the complementary characteristics between multi-view data, multi-view learning can obtain better performance of learning tasks than relying on just one single view [27]. Till now, multi-view learning has been widely studied in a variety of areas, such as knowledge management, data mining, multimedia and information retrieval [18, 27].

Multi-view clustering, as one of basic tasks of multi-view learning, provides a natural way to cluster multi-view datasets. Generally, the main challenge lies in the mining of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
CIKM'15, October 19–23, 2015, Melbourne, Australia.
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806526.

complementary information among multiple sources of information. Fortunately, a number of promising approaches have been proposed, which can be roughly classified into four categories [27, 31, 8]. Methods in the first category are subspace based ones [5, 15, 16, 29, 28], which learn a latent space so that different views are comparable in that space. On the other hand, some methods are co-training based algorithms [1, 14, 33], which obtain the clustering results in an iterative clustering manner. The third category is called late fusion [3, 10, 13], which combines the clustering results of different views by voting or other fusion strategies. The last category learns a unified similarity matrix among multi-view data [32, 19, 30], which serves as an affinity matrix for final clustering. For more details on multi-view clustering, please refer to Section 2.

It should be noted that previous multi-view clustering methods usually assume that all the examples have complete information of all views, i.e., each example in the database has complete feature sets. However, in real applications, it is often the case that some views suffer from missing information. For example, in image clustering based on visual and textual features, some images have only visual or textual information with only a part of the data sharing both feature sets. We call such dataset incomplete multi-view data. When traditional multi-view clustering methods confront the above scenarios, a naive approach may remove the data examples that are incomplete. However, this strategy is contradicting with our goal that groups all data examples into their corresponding clusters.

Recently, a few attempts have been made concentrating on multi-view clustering with incomplete views, which may be classified into two major categories. The first strategy preprocesses incomplete views by filling missing information. Piyush et al. [23] and Shao et al. [24] provided to complete the kernel matrices of the incomplete views and then used kernel-based clustering methods for final clustering. However, these two methods can only deal with the kernel-based multi-view clustering algorithms, which greatly limit their extension to more widely used subspace-based multi-view clustering methods. Recently, Li et al. [34] claimed that the methods in the first category are not a good choice for incomplete multi-view clustering and accordingly proposed a non-negative matrix factorization based method (PVC), which proved to be effective for document clustering. But, there are also some limitations about the PVC method. Firstly, since data are now often with high dimensional and noisy features, it becomes urgent to select relevant and discriminative features when performing clustering. Secondly, PVC utilizes nonnegative matrix factorization to learn latent representations of data, which cannot well deal with data with negative feature representations.

In this paper, we propose a novel incomplete multi-view clustering method based on joint feature selection and subspace learning (as shown in Figure 1). Firstly, we utilize a regression-like objective to learn a subspace, in which data examples from different views can be compared irrespective of the heterogeneity between feature sets. To directly explore the complementary characteristics among different views, the latent representations of data examples with complete views are expected to be the same. Besides, since the features for different views may be high dimensional and even noisy, feature selection is performed to select relevant features for latent space learning. At last, a graph regularization is utilized to further explore the inter-view and intra-view relationship of the data examples. To these ends, we develop an objective to achieve all the above goals, and accordingly propose an alternating minimization algorithm to find an efficient solution. Extensive experiments demonstrate that our method outperforms the state-of-the-art multi-view clustering methods.

Main contributions: 1) We propose a novel incomplete multi-view clustering method, which incorporates feature selection, subspace learning and inter-view and intra-view similarity preserving into a unified objective. 2) We develop an iterative optimization algorithm to efficiently solve the proposed objective, and theoretical analysis is provided to guarantee its convergence. 3) We validate our proposed method with extensive experiments under two settings on four databases, achieving better performance than the state-of-the-art methods.

The rest of the paper is organized as follows. In Section 2, we briefly review multi-view clustering methods. Then our incomplete multi-view clustering algorithm is elaborated in Section 3. Section 4 shows experimental results and analysis. Finally, Section 5 concludes the paper.

2. RELATED WORK

Multi-view learning deals with data represented by multiple distinct feature sets and aims at boosting the learning performance [27]. Till now, plenty of methods have been developed with sound theories and multi-view learning has become a hot topic with widespread applicability [20, 26]. For example, the co-training method [2], one of the most famous multi-view learning frameworks, has been widely applied for webpage classification. When multi-view learning meets the unsupervised clustering task, multi-view clustering is accordingly developed to extend traditional single view clustering to the multi-view case.

Generally, multi-view clustering can be roughly classified into four categories. Algorithms in the first category find a unified low-dimensional space, in which the learned embedding of data can well explore the complementary information among different views [5, 15, 16, 29, 11]. These methods obtain final clustering results through a single view clustering method performed on the learned embedding. Kamalika et al. [5] obtained the low-dimensional subspace of multi-view data through the widely used canonical correlation analysis technique. Kumar et al. [15] proposed two objectives to regularize the Laplacian embeddings between different views to be similar and spectral analysis is employed for parameter learning. Liu et al. [16] developed a multi-view non-negative matrix factorization based method to gain a consensus embedding of the original data, which is further developed by He et al. [12] using various co-regularization forms. Recently, Wang et al. [29] proposed a regression-like objective, which conducts multi-view clustering and feature selection at the same time. Tang et al. [28] utilized unsupervised feature selection to cluster multi-view social media, and Qian and Zhai [22] also resorted to the above technique to obtain a low dimensional embedding of multi-view web news data.

Methods in the second category integrate multiple sources of information in the clustering process. Typical examples are the co-training and co-EM based multi-view clustering methods [1, 14, 33]. Kumar et al. [14] resorted to co-training, a popular semi-supervised tool, to develop the first co-training based multi-view clustering algorithm. Furthermore, Zhan et al. [33] proposed a more sophisticated multi-view clustering algorithm by combining LDA, k-means and the co-training framework. The third category is late fusion, which integrates the clustering results obtained from each view by voting or other fusion strategies [3, 10, 13]. Long et al. [17] proposed to learn the best clusters by fusing the clusters from each view through mapping functions. Greene et al. [10] utilized the matrix factorization based method to obtain optimal clusters. The last category aims to learn a unified similarity matrix among multi-view data, which serves as affinity matrix for final clustering [32, 19, 30, 4]. Muthukrishnan et al. [19] combined multiple similarity matrices by using a regularization framework to obtain a better similarity graph. Furthermore, Yin et al. [32] resorted to subspace clustering to obtain comparable similarity matrices through pairwise co-regularization.

The existing multi-view clustering methods mainly focus on the data with complete views, i.e., each data example has complete feature sets. However, in real applications, some data examples possibly lose some views. To handle this scenario, a few works have been developed [23, 24, 34]. Piyush et al. [23] proposed a spectral-based multi-view clustering method, which can deal with the scenario that at least one view is complete. They use the similarity matrix of the complete view to fill the kernel matrices of incomplete views through Laplacian regularization. Furthermore, Shao et al. [24] improved [23] by dealing with situations where no views are complete. They collectively fill all the kernel matrices by optimizing the alignment of shared data examples in the database. To sum up, both methods are based on the kernel matrices and can only adapt to kernel-based multi-view clustering. Recently, Li et al. [34] proved that the above methods are not a good choice for incomplete multi-view clustering and proposed a subspace based method using nonnegative matrix factorization (PVC). However, PVC has some limitations restricting its applications. Firstly, multi-view data are often high dimensional and noisy, and it may be necessary to select discriminative features when learning the latent subspace. Secondly, PVC utilizes nonnegative matrix factorization to learn latent representations of the data, which limits it applications to data with negative feature sets.

3. METHOD

3.1 Notations

For the sake of introducing our model, we discuss a dataset with two views and it is straight-forward to extend our model to the dataset with more views. Assume the two views of data are represented as $X^{(1)}$ and $X^{(2)}$ respectively. In the traditional multi-view clustering setting, a complete database $X = \{X^{(1)}, X^{(2)}\} = \{(X_i^{(1)}, X_i^{(2)}), i = 1, ..., N\}$ is given, where N is the number of data examples. However, in the incomplete view setting, we are given data $\hat{X} = \{\hat{X}^{(1,2)}, \hat{X}^{(1)}, \hat{X}^{(2)}\}$, where $\hat{X}^{(1,2)} = \{(X_i^{(1)}, X_i^{(2)}), i = 1, ..., c\}$, $\hat{X}^{(1)} = \{(X_i^{(1)}), i = c+1, ..., c+m\}$ and $\hat{X}^{(2)} = \{(X_i^{(2)}), i = c+m+1, ..., c+m+n\}$ represent data examples having complete views, only the first view and only the second view with the number of examples being c, m and n respectively. In total, we have $c+m+n$ examples in the database.

We denote $X_c^{(1)} \in R^{d_1 \times c}$ and $X_c^{(2)} \in R^{d_2 \times c}$ the examples having both views with d_1 and d_2 being the dimensionality of the two feature sets. Then $X_c^{(1)}$ and $\hat{X}^{(1)}$ consist of the examples in the first view, as denoted as $\bar{X}^{(1)} = [X_c^{(1)}, \hat{X}^{(1)}] \in R^{d_1 \times (c+m)}$. Similarly, we have the examples of the second view represented as $\bar{X}^{(2)} = [X_c^{(2)}, \hat{X}^{(2)}] \in R^{d_2 \times (c+n)}$. Our task is to group the incomplete multi-view data into their corresponding groups.

3.2 Formulation

Generally, multi-view data consist of heterogeneous feature sets representing the same object, and therefore they share the same class labels. We denote $Y = [Y_1, ..., Y_{c+m+n}]^T \in \{0,1\}^{(c+m+n) \times k}$ the class index of the incomplete multi-view database, where $Y_i \in \{0,1\}^{k \times 1}$ is the class indicator vector for the i-th example and k is the number of clusters. Then the scaled indicator matrix F is defined as $F = [F_1, ..., F_{c+m+n}]^T = Y(Y^TY)^{-1/2}$ [28] with the property $F^TF = I_k$, where I_k is an identity matrix with a size of k.

In our objective, we aim to find a F satisfying the above properties for multi-view clustering and the advantages are listed as follows. Firstly, F reflects the class indicator of the multi-view data, which is a higher level semantic representation of data. Even though data consist of multiple heterogeneous features, they should share the same semantic information. By introducing this semantic space, we construct a bridge for different heterogeneous feature sets. Furthermore, using such an indictor matrix, we can learn the projection matrix for each view and perform feature selection in a supervised manner, which will be described later.

To learn the indictor matrix, we learn a projection matrix for each view to project their original space to such a semantic space. The objective is then formulated as:

$$\begin{aligned} \min ||[X_c^{(1)}, \hat{X}^{(1)}]^T U_{(1)} - [F^c; \hat{F}^{(1)}]||^2 + \beta \|U_{(1)}\|_{21} \\ + ||[X_c^{(2)}, \hat{X}^{(2)}]^T U_{(2)} - [F^c; \hat{F}^{(2)}]||^2 + \beta \|U_{(2)}\|_{21} \\ s.t. \quad F^T F = I_k, F \geq 0 \end{aligned} \tag{1}$$

where $U_{(1)} \in R^{d_1 \times k}$ and $U_{(2)} \in R^{d_2 \times k}$ are projection matrices for the two views. $F^c \in R^{c \times k}$, $\hat{F}^{(1)} \in R^{m \times k}$ and $\hat{F}^{(2)} \in R^{n \times k}$ are the learned latent representations for data examples with complete views, only the first view and only the second view, respectively. It can be seen that we explore the relationship between the two views by enforcing the data examples with complete feature sets to have the same latent representation. $||U_{(1)}||_{21} = \sum_i ||U_{(1)}(i,:)||$, where $U_{(1)}(i,:)$ is the i-th row of $U_{(1)}$. The $\ell 21$-norms imposed on the projection matrices result in relevant features being selected for each view as always done by supervised feature selection [21]. β is a regularization parameter controlling the degree of the sparsity of projection matrices. When β is big, only a small subset of features will be selected, otherwise a large subset of features will be selected. $F = [F^c; \hat{F}^{(1)}; \hat{F}^{(2)}] \in R^{(c+m+n) \times k}$ is the learned latent representations for all the data examples, and $F^TF = I_k$ and $F \geq 0$ are used to constrain the latent representation to be consistent with the indicator matrix of the database.

In Equation 1, we project different feature sets into the same latent space and the relationship between different views is explored on such space in a direct manner. In the following part, we add extra regularization constraints on the projection matrices to further dig the relationship between data examples in each view and between the two views to model the structure of the multi-view data. More specifically, we hope to preserve the intra-view similarity and the inter-view similarity relationships in the dataset. Their details are listed as follows.

1) Intra-view similarity relationship: to preserve the local structure of data examples in each view, we constrain the neighborhood relationship between data points under each view also hold in the learned latent space. Generally, the neighborhood structure can be obtained by using a Gaussian based kernel matrix. we denote the matrices as $W^{(1)}$ and $W^{(2)}$ for the two views respectively and the entities in the matrix indicate the similarity between two data examples under a specific view. The detailed formulation is:

$$W_{ij}^{(t)} = \begin{cases} \exp(-z_{ij}^{(t)}/2\sigma^2), \bar{X}_i^{(t)} \in N_k(\bar{X}_j^{(t)}) \text{or} \bar{X}_j^{(t)} \in N_k(\bar{X}_i^{(t)}) \\ 0, \qquad \text{otherwise} \end{cases} \tag{2}$$

where $z_{ij}^{(t)}$ is the Euclidean distance between data examples $X_i^{(t)}$ and $X_j^{(t)}$ and $N_k(X_i^{(t)})$ indicates the examples of k nearest neighbors of $X_i^{(t)}$.

2) Inter-view similarity relationship: although different views of data have different feature sets, they share the same semantics if they represent the same content or topic. To preserve such inter-view similarity when learning the projection matrices, we construct similarity matrices $W^{(12)}$ and $W^{(21)}$ for view 1 to view 2 and view 2 to view 1 respectively. Under the incomplete view setting $W^{(12)} = (W^{(21)})^T$ and they are defined as:

$$W_{ij}^{(12)} = \begin{cases} 1, \text{ if } \bar{X}_i^{(1)} \text{and } \bar{X}_j^{(2)} \text{have same semantics} \\ 0, \qquad \text{otherwise} \end{cases} \tag{3}$$

Using the inter-view and intra-view similarities, we define the overall similarity matrix W as:

$$W = \begin{bmatrix} W^{(1)} & W^{(12)} \\ W^{(21)} & W^{(2)} \end{bmatrix} \tag{4}$$

Based on this similarity, we define the regularization on the projection matrices as :

$$\begin{aligned} \Omega(U_{(1)}, U_{(2)}) = & \sum_{ij} W_{ij}^{(1)} ||U_{(1)}^T \bar{X}_i^{(1)} - U_{(1)}^T \bar{X}_j^{(1)}||^2 \\ & + \sum_{ij} W_{ij}^{(2)} ||U_{(2)}^T \bar{X}_i^{(2)} - U_{(2)}^T \bar{X}_j^{(2)}||^2 \\ & + \sum_{ij} W_{ij}^{(12)} ||U_{(1)}^T \bar{X}_i^{(1)} - U_{(2)}^T \bar{X}_j^{(2)}||^2 \\ & + \sum_{ij} W_{ij}^{(21)} ||U_{(2)}^T \bar{X}_i^{(2)} - U_{(1)}^T \bar{X}_j^{(1)}||^2 \end{aligned} \tag{5}$$

and it is further rewritten as:

$$\Omega(U_{(1)}, U_{(2)}) = \sum_{s=1}^{2} \sum_{t=1}^{2} Tr(U_{(s)}^T \bar{X}^{(s)} L_{st} (\bar{X}^{(t)})^T U_{(t)}) \tag{6}$$

where $L = D - W$ is the Laplacian matrix and D is a diagonal matrix with its i-th diagonal element defined as the sum of the i-th row of W. Tr is the trace of a matrx.

Adding this regularization constraint to Equation 1, we obtain the final objective as:

$$\begin{aligned} \min_{U,F} & ||[X_c^{(1)}, \hat{X}^{(1)}]^T U_{(1)} - [F^c; \hat{F}^{(1)}]||^2 + \beta ||U_{(1)}||_{21} \\ & + ||[X_c^{(2)}, \hat{X}^{(2)}]^T U_{(2)} - [F^c; \hat{F}^{(2)}]||^2 + \beta ||U_{(2)}||_{21} \\ & + \gamma \sum_{s=1}^{2} \sum_{t=1}^{2} Tr(U_{(s)}^T \bar{X}^{(s)} L_{st} (\bar{X}^{(t)})^T U_{(t)}) \\ s.t. \quad & F^T F = I_k, \ F \geq 0 \end{aligned} \tag{7}$$

In our objective, we have three terms: using the projection matrix to project each incomplete view to the latent space defined by F; feature selection for each view using the $\ell 21$-norm based constraint and the inter-view and intra-view similarity preserving term defined by the Laplacian matrix. Besides, the constraints imposed on F guarantee that each example only belongs to one group.

3.3 Optimization

In this section, we propose to optimize the objective as described in Equation 7. Since the variables, such as the projection matrix and the latent representation, are coupled together, it may be difficult to optimize them at the same time. Hence, we propose to alternatively optimize the variables to obtain a local solution.

1) Optimize F with fixed U: the constraints on F in Equation 7 make the optimization not an easy problem, especially different views only have part of all the latent representations, i.e., $[F^c; \hat{F}^{(1)}]$ and $[F^c; \hat{F}^{(2)}]$ are only part of F. To handle this, we optimize F^c, $\hat{F}^{(1)}$ and $\hat{F}^{(2)}$ separately and relax the constraints to the following form:

$$(F^c)^T F^c = I_k, F \geq 0 \tag{8}$$

Even though the orthogonal constraint on F^c may not be rigorous when data examples with complete feature sets do not have all kinds of class labels. We ignore this slight influence. In turn, it makes our optimization very compact. As for $\hat{F}^{(k)}, (k = 1, 2)$, since examples in the same view share the same projection matrix and these examples follow the same data distribution, $\hat{F}^{(k)}$ will have similar characteristic with F^c. In summary, the relaxed constraints will have almost the same effect with that of the original ones and can make the optimization more succinct.

We denote the objective in Equation 7 as O. Then minimizing O over F^c, $\hat{F}^{(1)}$ and $\hat{F}^{(2)}$ are simplified as:

$$\begin{aligned} \min_{F^c} & ||(X_c^{(1)})^T U_{(1)} - F^c||^2 + ||(X_c^{(2)})^T U_{(2)} - F^c||^2 \\ s.t. \quad & (F^c)^T F^c = I_k, F^c \geq 0 \end{aligned} \tag{9}$$

$$\begin{aligned} & \min_{\hat{F}^{(1)} \geq 0} ||(\hat{X}^{(1)})^T U_{(1)} - \hat{F}^{(1)}||^2 \\ & \min_{\hat{F}^{(2)} \geq 0} ||(\hat{X}^{(2)})^T U_{(2)} - \hat{F}^{(2)}||^2 \end{aligned} \tag{10}$$

To optimize F^c, we bring in Lagrangian function as:

$$\begin{aligned} L(F^c) = & Tr(\Gamma((F^c)^T F^c - I_k)) \\ & - Tr(\Lambda F^c) + \sum_{i=1,2} Tr(-2A_i^T F^c + (F^c)^T F) \end{aligned} \tag{11}$$

where Γ and Λ are Lagrangian multipliers of the above function and $A_i = (X_c^{(i)})^T U_{(i)}$. Applying KKT condition, i.e., $\Lambda(s,t) F^c(s,t) = 0$, we obtain:

$$(\sum_{i=1,2} (-A_i + F^c) + F^c \Gamma)(s,t) \, F^c(s,t) = 0 \tag{12}$$

and we can obtain the following updating rule for F^c:

$$F^c(s,t) = F^c(s,t) \sqrt{\frac{(\sum_{i=1,2} A_i^+ + F^c \Gamma^-)(s,t)}{(\sum_{i=1,2} (A_i^- + F^c) + F^c \Gamma^+)(s,t)}} \tag{13}$$

where for a matrix C, $C^+(s,t) = (|C(s,t)| + C(s,t))/2$, $C^-(s,t) = (|C(s,t)| - C(s,t))/2$ and $C = C^+ - C^-$. As for Γ, its diagonal elements are obtained by summing s: $\Gamma(s,s) = \sum_{i=1,2}((F^c)^T A_i - I_k)(s,s)$. And the off-diagonal elements of Γ are approximated by ignoring the non-negative values of F^c: $\Gamma(s,t) = \sum_{i=1,2}((F^c)^T A_i - I_k)(s,t)$. In summary, Γ is calculated by $\Gamma = \sum_{i=1,2}((F^c)^T A_i - I_k)$.

To optimize $\hat{F}^{(1)}$ and $\hat{F}^{(2)}$, we directly obtain their gradients and the updating rule is:

$$\hat{F}^{(i)} = \max((X_c^{(i)})^T U_{(i)}, 0), \quad i = 1,2 \tag{14}$$

2) Optimize U with fixed F: Minimizing the objective O in Equation 7 with respect to $U_{(1)}$ and $U_{(2)}$ are rewritten as:

$$\begin{aligned}\min_{U_{(s)}} \sum_{s=1,2} ||(\bar{X}^{(s)})^T U_{(s)} - \bar{F}^{(s)}||^2 + \beta \sum_{s=1,2} ||U_{(s)}||_{21} \\ +\gamma \sum_{s=1}^{2}\sum_{t=1}^{2} Tr(U_{(s)}^T \bar{X}^{(s)} L_{st} (\bar{X}^{(t)})^T U_{(t)})\end{aligned} \tag{15}$$

where $\bar{X}^{(s)}, (s=1,2)$ and $\bar{F}^{(s)}, (s=1,2)$ are the feature matrix and the latent representation for one view as described before. They consist of the data examples with both feature sets and only with the s-th feature set.

Differentiating the objective function in Equation 15 with respect to $U_{(s)}$ and setting it to zero, we have the following equation:

$$\begin{aligned}\bar{X}^{(s)}((\bar{X}^{(s)})^T U_{(s)} - \bar{F}^{(s)}) + \beta D_{(s)} U_{(s)} \\ +\gamma \bar{X}^{(s)} L_{ss} (\bar{X}^{(s)})^T U_{(s)} + \gamma \sum_{t \neq s} \bar{X}^{(s)} L_{st} (\bar{X}^{(t)})^T U_{(t)} = 0\end{aligned} \tag{16}$$

where $D_{(s)}$ is a diagonal matrix with its i-th diagonal element calculated as $D_{(s)}(i,i) = 1/(2||U_{(s)}(i,:)||)$, and $U_{(s)}(i,:)$ is the i-th row of $U_{(s)}$. Practically, $D_{(s)}(i,i)$ is calculated by[1]:

$$D_{(s)}(i,i) = \frac{1}{2\sqrt{||U_{(s)}(i,:)||^2 + \varepsilon}} \tag{17}$$

where ε is a smoothing term, which is usually set to be a small positive value.

Then Equation 16 is further written as:

$$\begin{aligned}(\bar{X}^{(s)}(\bar{X}^{(s)})^T + \beta D_{(s)} + \gamma \bar{X}^{(s)} L_{ss} (\bar{X}^{(s)})^T) U_{(s)} \\ = \bar{X}^{(s)} \bar{F}^{(s)} - \gamma \sum_{t \neq s} \bar{X}^{(s)} L_{st} (\bar{X}^{(t)})^T U_{(t)}\end{aligned} \tag{18}$$

The objective can be optimized using the following equation:

$$\begin{aligned}U_{(s)} = (\bar{X}^{(s)}(\bar{X}^{(s)})^T + \beta D_{(s)} + \gamma \bar{X}^{(s)} L_{ss} (\bar{X}^{(s)})^T)^{-1} \\ (\bar{X}^{(s)} \bar{F}^{(s)} - \gamma \sum_{t \neq s} \bar{X}^{(s)} L_{st} (\bar{X}^{(t)})^T U_{(t)})\end{aligned} \tag{19}$$

Algorithm 1 gives the overall optimization for equation 7. In Step 3, we calculate the latent representation for the incomplete multi-view dataset. In Steps 4 and 5, we optimize the projection matrices $U_{(s)}, (s=1,2)$. Finally Steps 3, 4 and 5 are repeated until convergence. Based on the latent representation, the final clustering results can be obtained by using regular clustering algorithms, e.g., k-means. The overall clustering algorithm is summarized in Algorithm 2.

Algorithm 1 Solving Equation 7 to obtain the latent representation of the incomplete multi-view dataset

```
Input:
    Incomplete multi-view dataset $\{\bar{X}^{(1)}, \bar{X}^{(2)}\}$ , parameter
    $\beta$ and $\gamma$, the number of classes.
1: $t = 1$. Initialize $U_{(s)}, (s = 1,2)$ and $F$ randomly;
2: while not converge do
3:     Calculate $F^c$, $\hat{F}^{(1)}$ and $\hat{F}^{(2)}$ using Equation 13 and 14
       respectively;
4:     Solve $D_{(s)}, (s = 1,2)$ using Equation 17;
5:     Calculate $U_{(s)}, (s = 1,2)$ using Equation 19 respec-
       tively;
6: end while
Output:
    The latent representation for the incomplete multi-view
    dataset $F = [F^c; \hat{F}^{(1)}; \hat{F}^{(2)}]$.
```

Algorithm 2 Clustering procedure for the incomplete multi-view dataset

```
Input:
    Incomplete multi-view dataset $\{\bar{X}^{(1)}, \bar{X}^{(2)}\}$ , parameter
    $\beta$ and $\gamma$, the number of classes.
1: Obtain the latent representation $F$ of all the data by
   using Algorithm 1.
2: Perform $k$-means clustering on $F$ to obtain the clustering
   results.
Output:
    Groups of the incomplete multi-view dataset
```

3.4 Convergence and complexity analysis

In this section, we prove that Algorithm 1 converges to a local minima.

Theorem 1. *The proposed iterative optimization strategy in Algorithm 1 will monotonically decrease the objective function in Equation 7 in each iteration until convergence.*

a) In Step 3 of Algorithm 1, we will resort to auxiliary function approach [7] to validate that the updating rule for F^c will monotonically decrease the objective value. As for the updating rule for $\hat{F}^{(1)}$ and $\hat{F}^{(2)}$, it is easy to verify that their objectives are convex and their optimization methods can decrease the objective function monotonically.

Let

$$\begin{aligned}H(F^c) = Tr(\sum_{i=1,2}(-2A_i^T F^c + (F^c)^T F^c) \\ +\Gamma((F^c)^T F^c - I_k))\end{aligned} \tag{20}$$

and it is further rewritten as:

$$\begin{aligned}H(F^c) = Tr(\sum_{i=1,2}(2(A_i^-)^T F^c + (F^c)^T F^c) + \Gamma^+(F^c)^T F^c \\ -Tr(\sum_{i=1,2}(2(A_i^+)^T F^c + \Gamma^-(F^c)^T F^c)\end{aligned} \tag{21}$$

Then the following function

$$\begin{aligned}&h(F^c, \tilde{F}^c) = \\ &\sum_{i,s,t}(A_i^-(s,t)\tfrac{F^c(s,t)^2 + \tilde{F}^c(s,t)^2}{\tilde{F}^c(s,t)} + \tfrac{\tilde{F}^c(s,t)F^c(s,t)^2}{\tilde{F}^c(s,t)}) \\ &-\sum_{st}(\sum_i 2A_i(s,t))\tilde{F}^c(s,t)(1 + \log\tfrac{F^c(s,t)}{\tilde{F}^c(s,t)}) \\ &+\sum_{st}\tfrac{(\tilde{F}^c\Gamma^+)(s,t)F^c(s,t)^2}{\tilde{F}^c(s,t)} \\ &+\sum_{ist}\Gamma^-(s,t)\tilde{F}^c(i,s)\tilde{F}^c(i,t)(1 + \log\tfrac{F^c(i,s)F^c(i,t)}{\tilde{F}^c(i,s)\tilde{F}^c(i,t)})\end{aligned} \tag{22}$$

[1] $||U_{(s)}(i,:)||$ can be zero, which cannot guarantee the convergence of the algorithm. Similar to [9], we add a smoothing term as in Equation 17.

ia an auxiliary function of $H(F^c)$. Besides, it is easy to verify that the Hessian matrix of $h(F^c, \tilde{F}^c)$ is a positive definite matrix, thus, $h(F^c, \tilde{F}^c)$ is convex and its global minimum is obtained as in Equation 13.

Through the definition of the auxiliary function and the above derivation, we can obtain the following inequality:

$$H(F_0^c) = h(F_0^c, F_0^c) \geq h(F_0^c, F_1^c) \geq H(F_1^c)... \tag{23}$$

Thus, the updating rule for F^c will monotonically decrease the objective value.

b) In Step 5 of Algorithm 1, we will prove that the updating rule in Equation 19 for $U_{(s)}, (s = 1, 2)$ will decrease the objective monotonically.

Taking $U_{(1)}$ as an example, we can derive that:

$$\begin{aligned} U_{(1)}^{t+1} = \min_{U_{(1)}} &||(\bar{X}^{(1)})^T U_{(1)} - \bar{F}^{(1)}||^2 + \beta tr(U_{(1)}^T D_{(1)}^{t+1} U_{(1)}) \\ &+\gamma \sum_{s=1}^{2} Tr(U_{(1)}^T \bar{X}^{(1)} L_{1s} (\bar{X}^{(s)})^T U_{(s)}) \end{aligned} \tag{24}$$

and Equation 19 is the analytic solution of the above function. Then we have:

$$L_{t+1} + \beta tr((U_{(1)}^T)^{t+1} D_{(1)}^{t+1} U_{(1)}^{t+1}) \leq L_t + \beta tr((U_{(1)}^T)^t D_{(1)}^{t+1} U_{(1)}^t) \tag{25}$$

where

$$\begin{aligned} L_{t+1} = &|(\bar{X}^{(1)})^T U_{(1)}^{t+1} - \bar{F}^{(1)}||^2 \\ &+\gamma Tr((U_{(1)}^T)^{t+1} \bar{X}^{(1)} L_{11} (\bar{X}^{(1)})^T U_{(1)}^{t+1}) \\ &+\gamma Tr((U_{(1)}^T)^{t+1} \bar{X}^{(1)} L_{12} (\bar{X}^{(2)})^T U_{(2)}^t) \end{aligned} \tag{26}$$

Substituting $D_{(1)}^{t+1}$ into the above inequality, we have:

$$\begin{aligned} L_{t+1} + \sum_i \sum_j \frac{U_{(1)}^{t+1}(i,j) U_{(1)}^{t+1}(i,j)}{2||U_{(1)}^t(i,:)||} \\ \leq L_t + \sum_i \sum_j \frac{U_{(1)}^t(i,j) U_{(1)}^t(i,j)}{2||U_{(1)}^t(i,:)||} \end{aligned} \tag{27}$$

Here we introduce a function $f(x) = x - x^2/(2a)$, which satisfies $\{\forall x \in R, f(x) \leq f(a) | a > 0\}$. Then we make x and a be $||U_{(1)}^{t+1}(i,:)||$ and $||U_{(1)}^t(i,:)||$ respectively, we have the following inequality:

$$\begin{aligned} ||U_{(1)}^{t+1}(i,:)|| - \sum_j \frac{U_{(1)}^{t+1}(i,j) U_{(1)}^{t+1}(i,j)}{2||U_{(1)}^t(i,:)||} \\ \leq \sum_j ||U_{(1)}^t(i,:)|| - \frac{U_{(1)}^t(i,j) U_{(1)}^t(i,j)}{2||U_{(1)}^t(i,:)||} \end{aligned} \tag{28}$$

Add both sides of the above inequality to Equation 27, we obtain the following inequality:

$$L_{t+1} + \beta ||U_{(1)}^{t+1}||_{21} \leq L_t + \beta ||U_{(1)}^t||_{21} \tag{29}$$

Thus the updating rule for U will decrease the objective function monotonically.

Combining the above derivation, we prove that Algorithm 1 converges to a local minimum.

Complexity analysis: We briefly discuss the computational complexity of our algorithm. As for the optimization of F, the main computation lies in the updating for F^c as in Equation 13, which mainly consists of some matrix multiplication operations. When optimizing U, we need to compute the overall multi-view similarity matrix, whose complexity is about $O(d_m N_m^2)$, where $d_m N_m^2$ being the product of the dimensionality and the square of the number of examples for the m-th view is the largest one among all views. However, it is a constant matrix and can be computed before the optimization of the variables. Besides, we need to use Equation 19 to calculate U, which solves an inverse problem. Instead, we can update the projection matrices by solving a linear system for $O(\hat{d}^2)(\hat{d} = \max(d_1, d_2))$.

4. EXPERIMENTS

4.1 Datasets

We report experiments on four widely used multi-view datasets and their descriptions are summarized in Table 1.

Dataset	# size	# view	# cluster	# feature size
USPS	2,000	2	10	76+216
Cora	2,708	2	7	2,708+1,433
BBC	2,012	2	5	822+840
WebKB	1,051	2	2	1,840+3,000

Table 1: Information of the multi-view datasets. # feature size means the dimensionality of the two feature sets of the database.

UCI Handwritten Digit Dataset[2] It consists of feature sets of handwritten numerals (0-9) extracted from Dutch utility maps. The database has 2,000 examples even-distributed in ten categories and is represented in terms of six visual features. Being same in [15], we use the 76 Fourier coefficients of the character shapes and the 216 profile correlations as two views.

Cora Dataset[3] It contains 2,708 scientific publications divided into 7 classes (Neural_Networks, Rule_Learning, Reinforcement_Learning, Probabilistic_Methods, Theory, Genetic_Algorithms, Case_Based). Two heterogeneous feature sets, i.e., citations and content are utilized here for experiments, where the content feature is represented by 0/1-valued word vector indicating the absence/presence of the corresponding word from the 1,433 words constructed dictionary.

BBC Dataset[4] It is a synthetic multi-view text database, which is constructed using single view BBC and BBCSport corpora. In total, it consists of 2,012 data examples categorized into 5 classes. The two views used here are the segments representations of the same document with the dimensions being 6,838 and 6,790 respectively. We use principal component analysis (PCA) to preprocess the data and the dimension is selected based on the eigenvalues of the covariance matrix obtained from the data.

WebKB Datasets[5] It is a webpage dataset from the computer science departments of four universities. The dataset consists of two categories, i.e., course and non-course with two heterogeneous feature sets, namely the textual content of the webpage and the link representation. Here the link representation is the anchortext on links in the other webpages linking to the current webpage.

4.2 Settings

To simulate the incomplete multi-view datasets, we randomly select part of examples to have only one single feature

[2] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[3] http://lig-membres.imag.fr/grimal/data.html

[4] http://mlg.ucd.ie/datasets/segment.html

[5] http://vikas.sindhwani.org/manifoldregularization.html

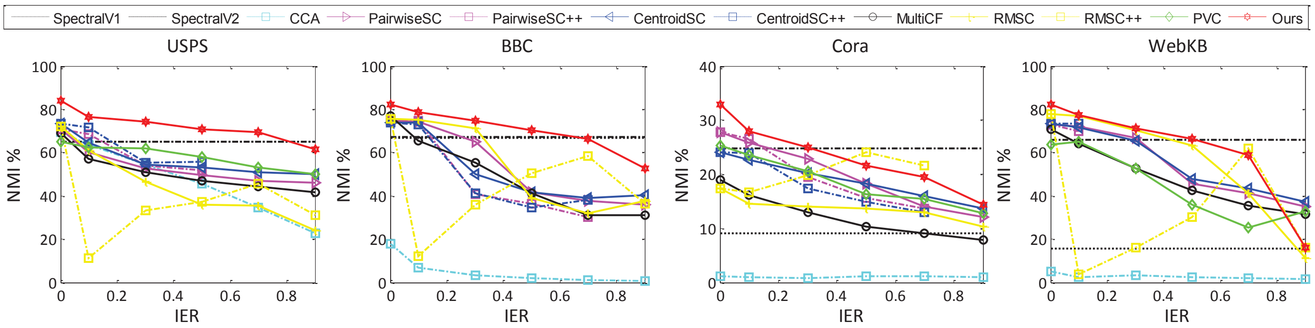

Figure 2: The NMI results on the four databases when both views suffer from the loss of examples. IER (incomplete example ratio) is the ratio of examples with only one feature set.

set. Similar to [34], two different settings are considered and listed as follows.

As described in Section 3, we denote m and n the numbers of examples appearing only in the first view and only in the second view respectively.

1) the first setting: $m > 0, n > 0$, namely both views do not contain all the examples in the database.

2) the second setting: either $m = 0$ or $n = 0$, namely at least one view is complete.

For the above two settings, we randomly select 10% to 90% of the total examples, with 20% as interval, to have only one feature set. And this process is repeated 10 times with the average to be reported. Besides, as for the first setting, we evenly distribute the number of examples for the two views for simplicity.

4.3 Compared methods

We compare our algorithm with several representative multi-view clustering methods, which consist of three subspace learning based methods and three kernel matrix based methods and their modifications.

SingleV1, SingleV2: We run spectral clustering [25] on the two views under the condition that all views have complete data examples.

CCA: We use canonical correlation analysis to obtain the latent representation of multi-view data and then apply k-means on the obtained representation.

PairwiseSC, CentroidSC: The multi-view spectral clustering methods based on two regularization frameworks developed by Kumar et al. [15].

MultiCF: Wang et al. [29] proposed a structure sparsity based unsupervised feature selection method for the task of multi-view clustering.

RMSC: Xia et al. [30] developed a multi-view spectral clustering method, which is based on low rank and sparse decomposition of the transition matrix.

PVC: Li et al. [34] proposed probably the only incomplete multi-view clustering method without filling the missing information.

PairwiseSC++, CentroidSC++, RMSC++: For the kernel based multi-view clustering algorithms, Piyush et al. [23] proposed to fill the kernel matrix of the view with incomplete examples using the kernel matrix of the view with complete examples. So in our second setting that one view is complete, we can use this method to fill the incomplete kernel matrix. Then the modified PairwiseSC, CentroidSC and RMSC methods may obtain better clustering results. Moreover, Shao et al. [24] proposed to fill the kernel matrices even there are no views with complete examples. Then in our first setting, we may promote PairwiseSC, CentroidSC and RMSC methods using this method. We denote the PairwiseSC, CentroidSC and RMSC methods with the preprocessing of the kernel matrix under the two settings as PairwiseSC++, CentroidSC++, RMSC++ respectively.

For the compared methods that are not designed for incomplete multi-view clustering, i.e., CCA, PairwiseSC, CentroidSC, MultiCF and RMSC, we just use zeros to replace incomplete feature sets. This may be a little arbitrary, but we find possibly no methods can well fill various types of features at the same time, e.g., visual features and textual features. Besides, it may be fair enough since our method do not preprocess the data at all. For PairwiseSC, CentroidSC, RMSC and PVC methods, we use the codes the authors have released to achieve their best performance and the method CCA is achieved using the LSCCA package[6]. As for MultiCF, we implement the method and follow the authors' suggestions to achieve the clustering results. For our method, we use KNN based Gaussian kernel to construct the intra-view similarity matrix and the number of the KNN neighbors and the width parameter for Gaussian kernel are empirically selected as ten percent of the total examples of the database and one respectively in all the experiments. As for the trade off parameters β and γ, they are empirically selected to achieve the best clustering results. We will test their effects in the parameter study part. Since k-means is used in all the experiments, it is run 20 times with random initialization and the mean value is reported.

Finally, by following [34], the normalized mutual information (NMI), as one of the most famous clustering evaluation measures, is utilized. Users can refer to [6] for more details on its definition.

4.4 Experimental results

4.4.1 Experimental results under the first setting

Figure 2 shows the clustering results of all the methods under the first setting, i.e., both views suffer from information loss. *IER* (incomplete example ratio) indicates the percentage of examples having only one feature set. Besides, the results of all methods with *IER* being zero are also reported as the upper bound of each method. Overall, it can be seen that our method performs better than all the com-

[6] http://www.public.asu.edu/jye02/Software/CCA/ index.html

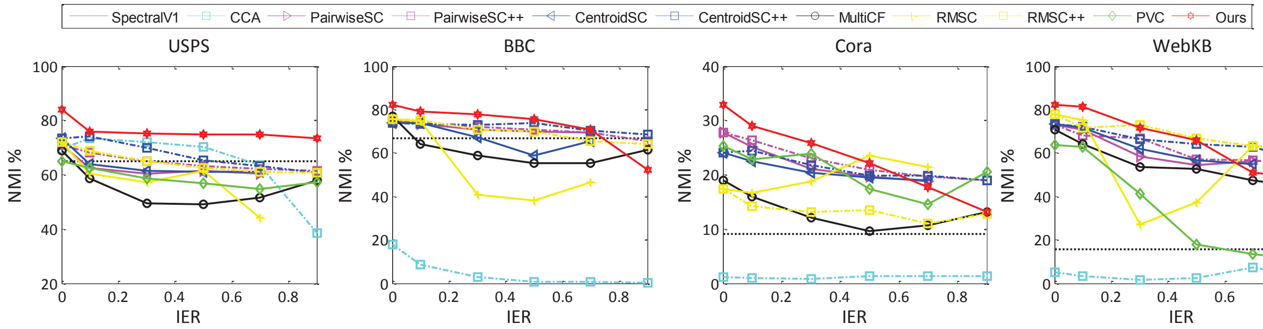

Figure 3: The NMI results on the four databases when the first view suffer from the loss of examples. IER (incomplete example ratio) is the ratio of examples with only one feature set.

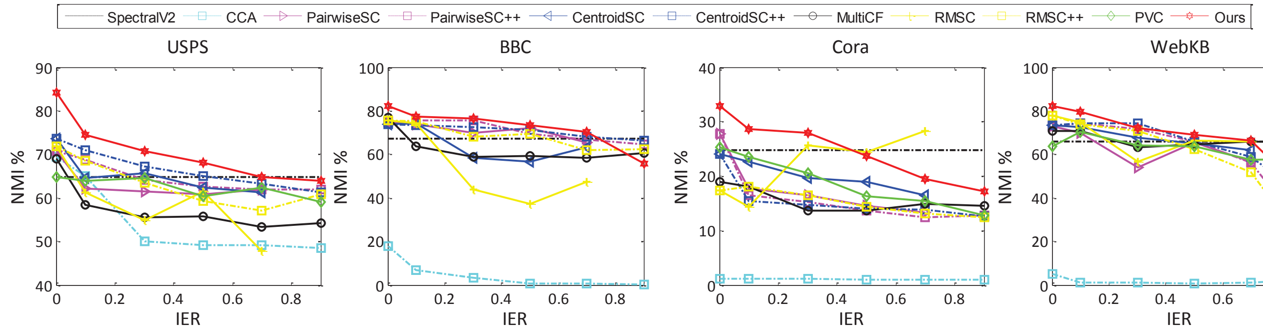

Figure 4: The NMI results on the four databases when the second view suffer from the loss of examples. IER (incomplete example ratio) is the ratio of examples with only one feature set.

peting methods under different incomplete example ratios on the four databases.

Compared with the results of SingleV1 and SingleV2 methods, our method performs better even with up to 50% of examples only appearing in one feature set on the USPS, Cora and WebKB datasets. This is an inspiring result, which indicates that our method can well explore the multi-view complementary information even in relatively large incomplete example ratios.

As for PairwiseSC, CentroidSC and RMSC, we utilize the method proposed in [24] to fill the kernel matrices of the incomplete views and accordingly PairwiseSC++, CentroidSC++, RMSC++ are developed. From Figure 2, they perform better in some databases and the performance gain seems not very considerable especially when *IER* being large. In summary, our method performs better although these kernel based multi-view clustering methods preprocess to fill the lost information.

As for PVC, it uses non-negative matrix factorization to find a unified low dimensional space and constrains the examples with complete views sharing the same representations to deal with the incomplete multi-view data. Compared with it, we also apply feature selection to select relevant features when learning the low dimensional subspace, which works confronting the high dimensional and noisy features. Besides, the multi-view data structure is also explored in the proposed method. Thus our method performs better than PVC.

One of the major differences between our method and the MultiCF method under complete views is the constraint imposed on the learned latent representation. We add the non-negative constraint, which is more reasonable to approach the normalized indictor matrix and this may be the reason that our method performs better when the incomplete example ratio is zero. Since MultiCF is not designed for incomplete multi-view data, our method also outperforms it when *IER* is greater than zero.

4.4.2 *Experimental results under the second setting*

Figures 3 and 4 display the clustering performance under the second setting with the first and second view suffering from incomplete examples respectively. It should be noted that we apply the method in [23] to fill the kernel matrix of the incomplete view using that of the complete view for PairwiseSC, CentroidSC and RMSC to obtain the PairwiseSC++, CentroidSC++, RMSC++ methods.

It can be seen that similar results are obtained as in Figure 2 except that all the methods obtain relatively better performance compared with that in the first setting under the same incomplete example ratio. This may be because there exists one complete view to aid the multi-view clustering and it may be more useful compared with the scenario of no complete views. Overall, our method still obtains the best clustering performance almost on all the datasets under this setting.

4.5 Parameter study

In our proposed model as in Equation 7, there are two parameters β and γ balancing the effect of feature projection term, $\ell21$-norm based feature selection term and graph regularization based structure preserving term. In this section, we investigate how the performance varies with the changes of the above two parameters. Due to space limitation, we conduct experiments on the four databases under the first setting and the incomplete example ratios are selected as 0 and 0.3 respectively. It should be noted that similar results can be obtained under the second setting. The results are shown in Figure 5.

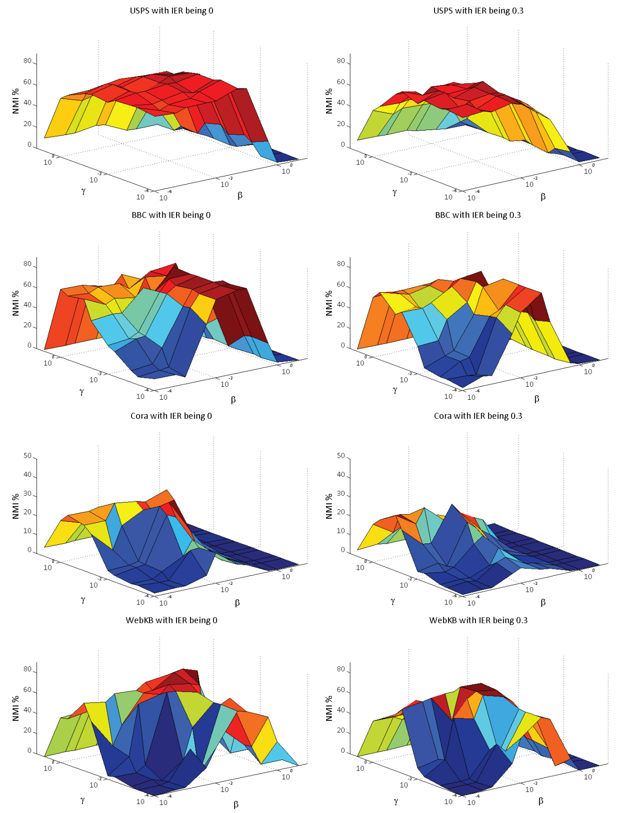

Figure 5: The NMI results on the four databases under the first setting with the incomplete example ratio being 0 and 0.3 respectively.

β controls the sparsity of the projection matrices. When it is small, the constraint will lose the effect of feature selection. In the case when β is too big, the sparse characteristic will lead to the loss of useful features and harm the learned latent representations. γ is the weight for the graph regularization term, which keeps the inter-view and intra-view data structure of the original spaces in the learned space. When it is too big, it may rely on too much of the neighborhood relationship obtained using the similarity metric and this may harm the intrinsical data structure because of the possible inaccuracy of the calculated similarity matrix. In summary, β and γ should be carefully selected and [0.001,0.01] is an optimal interval when the multi-view data is normalized.

4.6 Convergence study

As discussed in Section 3.4, the optimization strategy converges to a local minima. In this section, we give the convergence and the corresponding NMI curves with the varying updating iterations. Due to space limitation, we only give the results under the first setting with incomplete example ratio being 30% and similar results can be achieved under the second setting. From Figure 6, it can be seen that the objective function converges fast, and the clustering performance needs about 100 iterations to reach the best results. This may because the initial values of the variables in Algorithm 1 are randomly set. In the future, we may consider a nice initialization method to reduce the number of iterations.

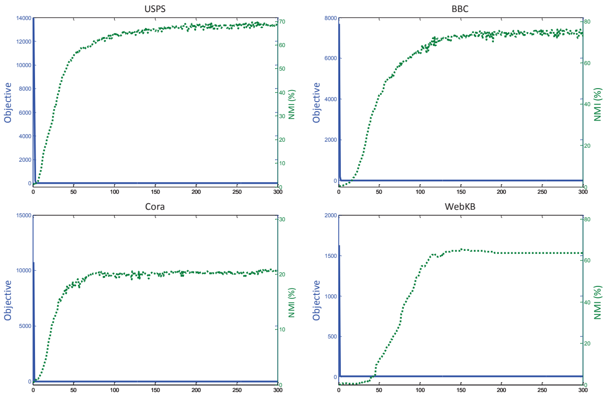

Figure 6: Convergence and the corresponding NMI curves for the four databases under the first setting with incomplete example ratio being 0.3.

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel incomplete multi-view clustering algorithm to cluster incomplete multi-view data. In our model, we learn a latent representation of the data examples, which serves as an approximation of the normalized indictor matrix. Besides, the complementary information between different views is explored by enforcing examples with complete views sharing the same representations. Through the $\ell21$-norm based constraint, relevant features are selected for the projection to the latent space. Furthermore, we add a graph regularization term to preserve the inter-view and intra-view data structure, which further promotes the clustering performance. Extensive experiments have validated the effectiveness of the proposed method compared with the state-of-the-art methods. Since it is practical to obtain partial label or must-link and cannot-link information between data examples, we may consider adding such information to promote clustering in the future.

Acknowledgments

This work is jointly supported by National Basic Research Program of China (2012CB316300), and National Natural Science Foundation of China (61175003, 61420106015, U1435221, 61403390).

6. REFERENCES

[1] S. Bickel and T. Scheffer. Multi-view clustering. *International Conference on Data Mining*, 4:19–26, 2004.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[3] E. Bruno and S. Marchand-Maillet. Multiview clustering: a late fusion approach using latent models. *ACM SIGIR Conference on Research and*

Development in Information Retrieval, pages 736–737, 2009.

[4] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang. Diversity-induced multi-view subspace clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[5] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. *International Conference on Machine Learning*, pages 129–136, 2009.

[6] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

[7] C. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, 2010.

[8] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on Signal Processing*, 62(4):905–918, 2014.

[9] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, 1997.

[10] D. Greene and P. Cunningham. A matrix factorization approach for integrating multiple data views. *Machine Learning and Knowledge Discovery in Databases*, pages 423–438, 2009.

[11] Y. Guo. Convex subspace representation learning from multi-view data. *AAAI Conference on Artificial Intelligence*, 1:2, 2013.

[12] X. He, M.-Y. Kan, P. Xie, and X. Chen. Comment-based multi-view clustering of web 2.0 items. *International Conference on World Wide Web*, pages 771–782, 2014.

[13] S. F. Hussain, M. Mushtaq, and Z. Halim. Multi-view document clustering via ensemble method. *Journal of Intelligent Information Systems*, 43(1):81–99, 2014.

[14] A. Kumar and H. Daumé. A co-training approach for multi-view spectral clustering. *International Conference on Machine Learning*, pages 393–400, 2011.

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

[16] J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. *SIAM International Conference on Data Mining*, 13:252–260, 2013.

[17] B. Long, S. Y. Philip, and Z. M. Zhang. A general model for multiple view unsupervised learning. *SIAM International Conference on Data Mining*, pages 822–833, 2008.

[18] E. Muller, S. Gunnemann, I. Farber, and T. Seidl. Discovering multiple clustering solutions: grouping objects in different views of the data. *International Conference on Data Engineering*, pages 1207–1210, 2012.

[19] P. Muthukrishnan, D. Radev, and Q. Mei. Edge weight regularization over multiple graphs for similarity learning. *International Conference on Data Mining*, pages 374–383, 2010.

[20] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. *International Conference on Machine Learning*, pages 689–696, 2011.

[21] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint l21-norms minimization. *Advances in neural information processing systems*, pages 1813–1821, 2010.

[22] M. Qian and C. Zhai. Unsupervised feature selection for multi-view clustering on text-image web news data. *ACM International Conference on Information and Knowledge Management*, pages 1963–1966, 2014.

[23] P. Rai, A. Trivedi, H. Daumé III, and S. L. DuVall. Multiview clustering with incomplete views. *NIPS Workshop on Machine Learning for Social Computing*, 2010.

[24] W. Shao, X. Shi, and P. S. Yu. Clustering on multiple incomplete datasets via collective kernel learning. *International Conference on Data Mining*, pages 1181–1186, 2013.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[26] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. *Advances in Neural Information Processing Systems*, pages 2222–2230, 2012.

[27] S. Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

[28] J. Tang, X. Hu, H. Gao, and H. Liu. Unsupervised feature selection for multi-view data in social media. *SIAM International Conference on Data Mining*, pages 270–278, 2013.

[29] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. *International Conference on Machine Learning*, pages 352–360, 2013.

[30] R. Xia, Y. Pan, L. Du, and J. Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. *AAAI Conference on Artificial Intelligence*, 2014.

[31] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *arXiv:1304.5634*, 2013.

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

[33] X. Zhao, N. Evans, and J.-L. Dugelay. A subspace co-training framework for multi-view clustering. *Pattern Recognition Letters*, 41:73–82, 2014.

[34] S.-Y. L. Y. J. Zhi and H. Zhou. Partial multi-view clustering. *AAAI Conference on Artificial Intelligence*, 2014.