Incomplete Multi-view Clustering via Structured Graph Learning

Jie Wu[1], Wenzhang Zhuge[1], Hong Tao[1], Chenping Hou[1(✉)], and Zhao Zhang[2]

[1] National University of Defense Technology,
No. 47, Yanwachi Street, Changsha 410073, China
wujienudt@yahoo.com, zgwznudt@yeah.net, taohong.nudt@hotmail.com,
hcpnudt@hotmail.com

[2] School of Computer Science and Technology, Soochow University,
No. 1, Shi-zi Street, Suzhou 215006, People's Republic of China
cszzhang@gmail.com

Abstract. In real applications, multi-view clustering with incomplete data has played an important role in the data mining field. How to design an algorithm to promote the clustering performance is a challenging problem. In this paper, we propose an approach with learned graph to handle the case that each view suffers from some missing information. It combines incomplete multi-view data and clusters it simultaneously by learning the ideal structures. For each view, with an initial input graph, it excavates a clustering structure with the consideration of consistency with the other views. The learned structured graphs have exactly c (the predefined number of clusters) connected components so that the clustering results can be obtained without requiring any post-clustering. An efficient optimization strategy is provided, which can simultaneously handle both the whole and the partial regularization problems. The proposed method exhibits impressive performance in experiments.

Keywords: Incomplete multi-view data · Clustering
Structured graph learning

1 Introduction

In many real applications, data are often coming from multiple sources or with multiple modalities becoming multi-view data, which have attracted extensive attention in the data mining field [1,7,14]. Observing that multiple views usually provide each other with complementary and compatible information, integrating them together to get better performance becomes natural [6,14]. However, in real applications, it is often the case that some or even all of the views suffer from some missing information [12,19]. For example, in speaker grouping, the audio and visual appearances represent two views and some speakers may miss audio or

Supported by the National Natural Science Foundation of China (No. 61473302, 61503396).

© Springer Nature Switzerland AG 2018
X. Geng and B.-H. Kang (Eds.): PRICAI 2018, LNAI 11012, pp. 98–112, 2018.
https://doi.org/10.1007/978-3-319-97304-3_8

visual information. Another example is document clustering, different language versions of a document can be regarded as multiple views, but many documents may not be translated into each language. Therefore, it is necessary to explore how to integrate such incomplete multi-view data. In this paper, we focus on multi-view clustering with incomplete data. Exiting approaches for this task can be divided into two categories: completion methods [2,13] and subspace methods [12,20].

Completion methods are based on matrix completion. For example, singular value decomposition imputation [2] first uses the eigenvalues to apply a regression to the complete attributes of the instance, to obtain an estimation of the missing value itself, and then applies conventional multi-view clustering methods [7,10] to derive the clustering results. The difference among different completion methods [7,9,10] is that they complete the incomplete data according to different principles. Their performances are usually unsatisfactory when the data are missing block-wise [12].

In recent years, some subspace methods to handle this case have been proposed. Partial multi-view clustering (PVC) [12] first divides the partial examples into two blocks and then executes non-negative matrix factorization (NMF) [11] to learn a low-dimensional representation for each multi-view data. Incomplete multi-modal visual data grouping (IMG) [20] can be regarded as a version of PVC, which requires that the low-dimensional representations conform to a self-learning manifold structure. Other methods such as [17,18] can also be classified into this category. Although these methods have achieved good performance, there is still room to improve. Most of them are based on NMF, which is aimed at learning the latent low-dimensional representations of data rather than clustering data. As a result, they must utilize a post-clustering such as K-means and spectral clustering to obtain the clustering results. Besides, they assume that the shared data in different views have exactly the same representation in the latent space, which may lead to a negative effect on the intrinsic inner structure of each individual view.

In this paper, with the goal to learn cluster structures directly, we propose the Structured Graph Learning (SGL) method to manipulate incomplete multi-view data and group them simultaneously. For each view, our SGL excavates a structured graph to combine the partial and the complete examples. To establish interaction between the different views, we naturally constrain the subgraphs corresponding to the shared data (in different views) to be close. Thus to some extent, we maintain the intrinsic structure of each individual view, as well as the consistency between different views. By improving the mechanism of the graph learning, the graph matrixes learned by our method have ideal structures-exactly c connected components, so that the clustering results can be derived from these graphs without requiring any post-clustering. Besides, we propose an efficient optimization strategy to solve our formulated problem. Experimental results on real benchmark data sets validate the advantages of our method.

2 The Proposed SGL

For the convenience of presentation, we take two-view data for illustration. As we can see from following formulations, extension to any number of views is direct. For example, most simply, we can separate the multiple views into pairs and then solve all the two-view problems. For input data, each column is a data and each row is an attribute. The feature dimensions of view 1 and view 2 data are d_1 and d_2, respectively. Input data $[X^{(1)^T}, X^{(2)^T}]^T \in \mathbb{R}^{(d_1+d_2)\times n_3}$, $\hat{X}^{(1)} \in \mathbb{R}^{d_1\times n_1}$ and $\hat{X}^{(2)} \in \mathbb{R}^{d_2\times n_2}$ denote the examples appearing and only appearing in both views, view 1 and view 2, respectively. $X^{(1)} \in \mathbb{R}^{d_1\times n_3}$ denotes the shared data in view 1, and $X^{(2)} \in \mathbb{R}^{d_2\times n_3}$ denotes the shared data in view 2. We assume that $X_1 \in \mathbb{R}^{d_1\times(n_3+n_1)} = [X^{(1)}, \hat{X}^{(1)}]$, $X_2 \in \mathbb{R}^{d_2\times(n_3+n_2)} = [X^{(2)}, \hat{X}^{(2)}]$, so that the n-th ($\forall n \leq n_3$) columns of X_1, X_2 belong to the same example, while the rest columns of them contain no common example. Figure 1 illustrates the notations.

We denote the initial graphs constructed from X_1 and X_2 as $A_1 \in \mathbb{R}^{(n_3+n_1)\times(n_3+n_1)}$ and $A_2 \in \mathbb{R}^{(n_3+n_2)\times(n_3+n_2)}$ respectively. And we denote the learned graphs that best approximate A_1 and A_2 as $S_1 \in \mathbb{R}^{(n_3+n_1)\times(n_3+n_1)}$ and $S_2 \in \mathbb{R}^{(n_3+n_2)\times(n_3+n_2)}$ respectively. We denote the initial graphs that correspond to $X^{(1)}$ and $X^{(2)}$ as $\bar{A}_1 \in \mathbb{R}^{n_3\times n_3}$ and $\bar{A}_2 \in \mathbb{R}^{n_3\times n_3}$ respectively. Thus $\bar{A}_1$ and $\bar{A}_2$ are the subgraphs of A_1 and A_2 respectively. And we denote the learned graphs that correspond to $X^{(1)}$ and $X^{(2)}$ as $\bar{S}_1 \in \mathbb{R}^{n_3\times n_3}$ and $\bar{S}_2 \in \mathbb{R}^{n_3\times n_3}$ respectively. Thus $\bar{S}_1$ and $\bar{S}_2$ are the subgraphs of S_1 and S_2 respectively. Figure 2 illustrates the learned graphs S_1 and S_2.

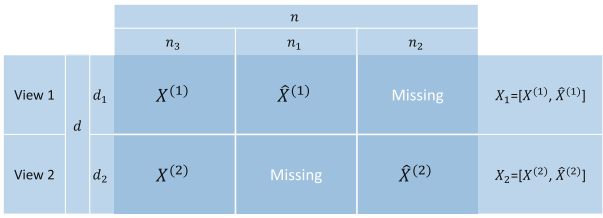
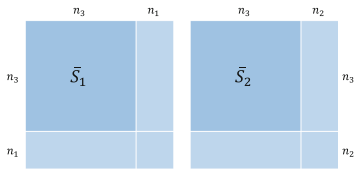


Fig. 1. Notations of data.

For each individual view, we aim to excavate its intrinsic clustering structure. Motivated by [16], which is effective in learning clustering structure with complete single-view data, we intend to learn the ideal structured graph from the initial graph. Given initial affinity matrixes A_1 and A_2 constructed from X_1 and X_2 respectively, we can learn the graph matrixes S_1 and S_2 that best approximate A_1 and A_2 respectively. Following are the elementary objectives:

$$\min_{\sum_j s_{1ij}=1, s_{ij}\geq 0} \| S_1 - A_1 \|_F^2, \tag{1}$$

$$\min_{\sum_j s_{2ij}=1, s_{ij}\geq 0} \| S_2 - A_2 \|_F^2. \tag{2}$$



(a) S_1 (b) S_2

Fig. 2. Learned graphs S_1 and S_2. S_1 and S_2 are corresponding to X_1 and X_2 respectively. $\bar{S}_1$ and $\bar{S}_2$ are corresponding to $X^{(1)}$ and $X^{(2)}$ respectively.

Here we append the constraints $\sum_j s_{1ij} = 1$ and $\sum_j s_{2ij} = 1$ $(\forall i)$ to avoid the case that some rows of S_1 or S_2 are all zeros, and to make S_1 and S_2 to be comparable at the same scale.

For each view, according to Eq. (1) (or Eq. (2)), we can learn a graph to explore and maintain the inner structure of each view. Further, we aim to bridge different views, for which it is natural to take into account the consistency between the shared data in different views. For $\bar{S}_1$, we can not only learn from $\bar{A}_1$, but also $\bar{S}_2$. For $\bar{S}_2$, we can not only learn from $\bar{A}_2$, but also $\bar{S}_1$. Thus we have the following objective function:

$$\begin{aligned} \min_{S_1,S_2} \quad & \| S_1 - A_1 \|_F^2 + \| S_2 - A_2 \|_F^2 + \mu \| \bar{S}_1 - \bar{S}_2 \|_F^2 \\ s.t. \quad & \sum_j s_{1ij} = 1, s_{1ij} \geq 0, \sum_j s_{2ij} = 1, s_{2ij} \geq 0, \end{aligned} \tag{3}$$

where $\mu > 0$ is a parameter that balances the first two terms and the last term.

To make the learned graphs have ideal structures directly for clustering tasks, we introduce the following property [4,15]:

Property 1. The number of connected components in the graph with the similarity matrix S is equal to the multiplicity c of the eigenvalue zero of L_S.

$L_S \in \mathbb{R}^{n \times n}$ in Property 1 is the Laplacian matrix of the nonnegative similarity matrix S, i.e. $L_S = D_S - (S^T + S)/2$ (where D_S is the degree matrix defined as a diagonal matrix whose i-th diagonal element is $\sum_j (s_{ij} + s_{ji})/2$).

Given a graph with graph matrix S, Property 1 indicates that if $rank(L_S) = n - c$, then this graph contains c connected components and each component corresponds to a cluster. We can obtain clustering results from the learned graph. Thus the problem (3) can be improved to the following problem:

$$\begin{aligned} \min_{S_1,S_2} \quad & \| S_1 - A_1 \|_F^2 + \| S_2 - A_2 \|_F^2 + \mu \| \bar{S}_1 - \bar{S}_2 \|_F^2 \\ s.t. \quad & \sum_j s_{1ij} = 1, s_{1ij} \geq 0, rank(L_{S_1}) = n_1 + n_3 - c, \\ & \sum_j s_{2ij} = 1, s_{2ij} \geq 0, rank(L_{S_2}) = n_2 + n_3 - c, \end{aligned} \tag{4}$$

which is equivalent to the following optimization problem for large enough values of both λ_1 and λ_2:

$$\begin{aligned}\min_{S_1,S_2} \quad & \parallel S_1 - A_1 \parallel_F^2 + 2\lambda_1 \sum_{i=1}^{c} \sigma_{1i}(L_{S_1}) + \parallel S_2 - A_2 \parallel_F^2 \\ & + 2\lambda_2 \sum_{i=1}^{c} \sigma_{2i}(L_{S_2}) + \mu \parallel \bar{S}_1 - \bar{S}_2 \parallel_F^2 \\ s.t. \quad & \sum_j s_{1ij} = 1, s_{1ij} \geq 0, \sum_j s_{2ij} = 1, s_{2ij} \geq 0.\end{aligned} \tag{5}$$

Here, $\sigma_{1i}(L_{S_1})$ and $\sigma_{2i}(L_{S_2})$ denote the i-th smallest eigenvalue of L_{S_1} and L_{S_2} respectively. Noting that $\sigma_{1i}(L_{S_1}) \geq 0(\forall i)$ and $\sigma_{2i}(L_{S_2}) \geq 0(\forall i)$, therefore when λ_1 and λ_2 are large enough, $\sum_{i=1}^{c} \sigma_{1i}(L_{S_1}) = 0$ and $\sum_{i=1}^{c} \sigma_{2i}(L_{S_2}) = 0$, so that the constraint $rank(L_{S_1}) = n_1 + n_3 - c$ and $rank(L_{S_2}) = n_2 + n_3 - c$ in the problem (4) will be satisfied. Thus, the problem (5) is equivalent to the problem (4). According to Ky Fan's Theorem [5], we have

$$\begin{aligned}\sum_{i=1}^{c} \sigma_{1i}(L_{S_1}) &= \min_{F_1 \in \mathbb{R}^{(n_1+n_3)\times c}, F_1^T F_1 = I} Tr(F_1^T L_{S_1} F_1), \\ \sum_{i=1}^{c} \sigma_{2i}(L_{S_2}) &= \min_{F_2 \in \mathbb{R}^{(n_2+n_3)\times c}, F_2^T F_2 = I} Tr(F_2^T L_{S_2} F_2),\end{aligned} \tag{6}$$

where F_1 and F_2 are the intermediate variables. Thus the problem (5) is further equivalent to the following problem:

$$\begin{aligned}\min_{S_1,F_1,S_2,F_2} \quad & \parallel S_1 - A_1 \parallel_F^2 + 2\lambda_1 Tr(F_1^T L_{S_1} F_1) + \parallel S_2 - A_2 \parallel_F^2 \\ & + 2\lambda_2 Tr(F_2^T L_{S_2} F_2) + \mu \parallel \bar{S}_1 - \bar{S}_2 \parallel_F^2 \\ s.t. \quad & \sum_j s_{1ij} = 1, s_{1ij} \geq 0, \sum_j s_{2ij} = 1, s_{2ij} \geq 0, \\ & F_1 \in \mathbb{R}^{(n_1+n_3)\times c}, F_1^T F_1 = I, F_2 \in \mathbb{R}^{(n_2+n_3)\times c}, F_2^T F_2 = I,\end{aligned} \tag{7}$$

where both parameters λ_1 and λ_1 are large enough to guarantee that the sums of the c smallest eigenvalues of both L_{S_1} and L_{S_1} are equal to zero.

According to Eq. (7), note that different from previous graph-based methods, our graphs are learned, which are learned from both the initial graphs and the consistency between different views. Besides, Eq. (7) is designed to cluster data points directly by learning graphs with structures which are ideal for clustering. From the learned structured graph, we can obtain clustering results of each view directly. Then by making a simple best one-to-one map between the clustering results of $X^{(1)}$ and $X^{(2)}$, the final clustering results can be derived.

Since the problem (7) is not convex and the regularization is added on the partial elements of the graph matrixes, it seems difficult to solve. We propose an efficient algorithm to solve this problem in the next section.

3 Optimization Algorithms

When S_1 and S_2 are fixed, optimizing F_1, F_2, the problem (7) becomes

$$\min_{F_1 \in \mathbb{R}^{(n_1+n_3)\times c}, F_1^T F_1 = I} Tr(F_1^T L_{S_1} F_1), \tag{8}$$

$$\min_{F_2 \in \mathbb{R}^{(n_2+n_3)\times c} F_2^T F_2 = I} Tr(F_2^T L_{S_2} F_2). \tag{9}$$

The optimal solution of F_1 and F_2 is formed by the c eigenvectors of L_{S_1} and L_{S_2} respectively corresponding to their c smallest eigenvalues.

When F_1, F_2 and S_2 are fixed, optimizing S_1, the problem (7) becomes

$$\begin{aligned} &\min_{S_1} \ \| S_1 - A_1 \|_F^2 + 2\lambda_1 Tr(F_1^T L_{S_1} F_1) + \mu \| \bar{S}_1 - \bar{S}_2 \|_F^2 \\ &s.t. \sum_j s_{1ij} = 1, s_{1ij} \geq 0, \end{aligned} \tag{10}$$

where $S_1, A_1 \in \mathbb{R}^{(n_3+n_1)\times(n_3+n_1)}$ and $\bar{S}_1, \bar{S}_2 \in \mathbb{R}^{n_3 \times n_3}$.

The problem (10) is equal to

$$\begin{aligned} \min_{\sum_j s_{1ij}=1, s_{1ij}\geq 0} &\sum_{1\leq i,j\leq n_3+n_1} (s_{1ij} - a_{1ij})^2 + \lambda_1 \sum_{1\leq i,j\leq n_3+n_1} s_{1ij}(f_{1i} - f_{1j}) \\ &+ \mu \sum_{1\leq i,j\leq n_3+n_1} (\bar{s}_{1ij} - \bar{s}_{2ij})^2. \end{aligned} \tag{11}$$

We can solve the following problems separately for each i since the problem (11) is independent for different i.

Update the first n_3 rows of $S_1 (1 \leq i \leq n_3)$. For each i, we denote $v_{1ij} = f_{1i} - f_{1j}$. Then the problem (11) becomes

$$\begin{aligned} \min_{\sum_j s_{1ij}=1, s_{1ij}\geq 0} &\sum_{j=1}^{n_3} (s_{1ij} - a_{1ij})^2 + \sum_{j=n_3+1}^{n_3+n_1} (s_{1ij} - a_{1ij})^2 + \lambda_1 \sum_{j=1}^{n_3} s_{1ij} v_{1ij} \\ &+ \lambda_1 \sum_{j=n_3+1}^{n_3+n_1} s_{1ij} v_{1ij} + \mu \sum_{j=1}^{n_3} (s_{1ij} - \bar{s}_{2ij})^2. \end{aligned} \tag{12}$$

For each i, we denote $s_{1i} = [\bar{s}_{1i}, \hat{s}_{1i}]$, $v_{1i} = [\bar{v}_{1i}, \hat{v}_{1i}]$ and $a_{1i} = [\bar{a}_{1i}, \hat{a}_{1i}]$. $\bar{s}_{1i}$, $\bar{v}_{1i}$ and $\bar{a}_{1i}$ contains and only contains the first n_3 elements of the vector s_{1i}, v_{1i} and a_{1i} respectively. $\hat{s}_{1i}$, $\hat{v}_{1i}$ and $\hat{a}_{1i}$ contains and only contains the last n_1 elements of the vector s_{1i}, v_{1i} and a_{1i} respectively.

The problem (12) can be written in vector form as

$$\min_{s_{1i}^T \mathbf{1}=1, s_{1i}\geq 0} \| \sqrt{1+\mu}\bar{s}_{1i} - \frac{(\bar{a}_{1i} - \frac{1}{2}\lambda_1 \bar{v}_{1i} + \mu \bar{s}_{2i})}{\sqrt{1+\mu}} \|_2^2 + \| \hat{s}_{1i} - (\hat{a}_{1i} - \frac{1}{2}\lambda_1 \hat{v}_{1i}) \|_2^2 . \tag{13}$$

We denote $c1 = \bar{a}_{1i} - 1/2\lambda_1 \bar{v}_{1i} + \mu \bar{s}_{2i}$, $c_2 = \hat{a}_{1i} - 1/2\lambda_1 \hat{v}_{1i}, b_1 = [c_1, c_2]$. U_1 is a diagonal matrix whose first n_3 diagonal elements are all $1 + \mu$, and the others are all 1. Then the problem (13) becomes

$$\min_{s_{1i}^T \mathbf{1}=1, s_{1i} \geq 0} \quad s_1 U_1 s_1^T - 2 s_1 b_1. \tag{14}$$

This problem can be solved by algorithm in [16].

Update the last n_1 rows of $S_1 (n_3 < i \leq n_3 + n_1)$. When $n_3 < i \leq n_3 + n_1$, denoting $v_{1ij} = f_{1i} - f_{1j}$, the problem (11) becomes

$$\min_{\sum_j s_{1ij}=1, s_{1ij} \geq 0} \quad \sum_j (s_{1ij} - a_{1ij})^2 + \sum_j s_{1ij} v_{1ij}. \tag{15}$$

The problem (15) can be written in vector form as

$$\min_{s_{1i}^T \mathbf{1}=1, s_{1i} \geq 0} \quad \| \, s_{1i} - (a_{1i} - \frac{1}{2} \lambda_1 v_{1i}) \, \|_2^2 \, . \tag{16}$$

This problem can be solved with an effective iterative algorithm [8], or solved by the solution with a similar form as Eq. (30) in [16]

When F_1, F_2 and S_1 are fixed, optimizing S_2 is similar to optimizing S_1. We omit the detailed process.

Update the first n_3 rows of $S_2 (1 \leq i \leq n_3)$. When $1 \leq i \leq n_3$, the problem (7) becomes

$$\min_{s_{2i}^T \mathbf{1}=1, s_{2i} \geq 0} \quad s_2 U_2 s_2^T - 2 s_2 b_2, \tag{17}$$

where $b_2 = [c_3, c_4]$, U_2 is a diagonal matrix whose first n_3 diagonal elements are all $1 + \mu$, and the others are all 1, $c_3 = (\bar{a}_{2i} - 1/2\lambda_2 \bar{v}_{2i} + \mu \bar{s}_{1i}), c_4 = \hat{a}_{2i} - 1/2\lambda_2 \hat{v}_{2i}$.

Update the last n_2 rows of $S_2 (n_3 < i \leq n_3 + n_2)$. When $n_3 < i \leq n_3 + n_2$, the problem (7) becomes

$$\min_{s_{2i}^T \mathbf{1}=1, s_{2i} \geq 0} \quad \| \, s_{2i} - (a_{2i} - \frac{1}{2} \lambda_2 v_{2i}) \, \|_2^2 \, . \tag{18}$$

The algorithm is provided in Algorithm 1, in which for each data point, we only update the nearest k similarities in S in order to reduce the complexity of updating S, F significantly. This technique makes our method applied on data sets with very large scale.

Algorithm 1. SGL

Input: A_1, A_2, μ, large enough λ_1, large enough λ_2.
Initialize: F_1 formed by the c eigenvectors of L_{A_1} corresponding to the c smallest eigenvalues, F_2 formed by the c eigenvectors of L_{A_2} corresponding to the c smallest eigenvalues.
repeat
  For each i, update the i-th row of S_1 by solving the problems (14)(16).
  Update F_1 by solving the problems (8).
  For each i, update the i-th row of S_2 by solving the problems (17)(18).
  Update F_2 by solving the problems (9).
until convergence
Output: S_1 with c connected components, S_2 with c connected components.

4 Discussion

4.1 Convergence Analysis

Property 2. The Algorithm 1 will monotonically decrease the objective of the problem in each iteration, and converge to a local optimum of the problem.

The brief idea in proving this theorem is summarized as follows. The Algorithm 1 converges because the objective function value of Eq. (7) decreases as iteration round increases. In detail, with fixed S_1 and S_2, optimal F_1 and F_2 can be obtained by solving the problems (8) and (9) respectively, which will reduce the objective function value, and with fixed F_1 and F_2, we can get optimal S_1 and S_2 by solving the problems (14) (16) and (17) (18) respectively, which will also reduce the objective function value. In summary, the Algorithm 1 will converge to a local optimum of the problem (7).

4.2 Computational Time

Since SGL is solved in an alternative way, we calculate their total computational complexity by analyzing the computational complexity in solving corresponding alternative optimization problems. The Algorithm 1 of SGL can be divided into three alternative optimization problems. The problems in Eqs. (8) and (9) updating matrix F_1 and F_2 respectively can be solved by eigen-decomposition, and the computational complexity is $O((n_1 + n_3)^3)$ and $O((n_2 + n_3)^3)$ respectively. Therefore, the total computational complexity of this procedure is $O(\max\{(n_1 + n_3)^3, (n_2 + n_3)^3\})$.

The problems in Eqs. (14) and (16) to update S_1 row by row are the subproblems of (10). The problem in Eq. (14) can be solved by Lagrange Multiplier and Newton's method, of which the computational complexity is $O((n_1 + n_3) \times n_3)$. The problem (16) can be solved by an efficient iterative algorithm [8], and the computational complexity is $O((n_1 + n_3) \times n_1)$. Therefore, the total computational complexity of this step is $O((n_1 + n_3)^2)$.

The problems in Eqs. (17) and (18) to update S_2 are similar to Eqs. (14) and (16) respectively, and the total computational complexity is $O((n_2 + n_3)^2)$.

As a result, the total computational complexity of SGL is $O(T \times \max\{(n_1 + n_3)^3, (n_2 + n_3)^3\})$, where T is the number of iterations. Obviously, cubic time complexity is caused by spectral decompositions. But in our algorithm, the Laplacian matrixes on which we implement spectral decompositions is sparse. Besides, nowadays there are many other novel alternative efficient methods handling spectral decomposition.

5 Experiment

5.1 Data Sets

MSRCv1 is comprised of 240 images in 8 class in total. Two pairs of visual features are extracted: 256 Local Binary Pattern(LBP) and 512 GIST noted as **MSRCv1GC**, 1302 CENTRIST and 200 SIFT noted as **MSRCv1CS**. **Ionosphere** is composed of 351 free electrons in the ionosphere observed by a system in Goose Bay, Labrador. The data can be divided into two classes: "Good" are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not. **Caltech101-7** contains 441 objective images in 7 categories as whole. We extract two features from each image data, including 200 SIFT and 32 Gabor texture. **Handwritten numerals (HW)** contains 2000 images data points for 0 to 9 digit classes and each class has 200 data points. We select 216 profile correlations (FAC) and 76 Fourier coefficients of the character shapes (FOU) for our clustering. **WebKB** is comprised of 1051 pages collected from four universities. Each page has 2 views: 334 citation view, and 2949 content view. Statistics of the data sets are summarized in Table 1.

Table 1. Data sets descriptions

Data sets	Size	View	Cluster Num	Feat1	Feat2
MSRCv1GC	210	2	7	256	512
MSRCv1CS	210	2	7	1302	200
Ionosphere	351	2	2	34	25
Caltech101-7	441	2	7	200	32
HW	2000	2	10	216	76
WebKB	1051	2	2	334	2949

5.2 Comparing Methods

Single-view methods **V1CLR** and **V2CLR**: With the partial example ratio being zero, CLR [16] algorithm is executed separately on view 1 and view 2 noted as V1CLR and V2CLR respectively.

Completion methods **CentroidSC** and **PairwiseSC**: Firstly, the partial examples are completed with the Robust Rank-k Matrix Completion [9] method. Two co-regularization schemes [10] are proposed to accomplish this work: Centroid-Based Co-regularization noted as CentroidSC and Pairwise-Based Co-regularization noted as PairwiseSC.

Subspace methods **PVC** [12] and **IMG** [20].

Other method: **CGC** [3] is aimed to deal with many-to-many instance relationship, which supports the situation of incomplete views.

We construct the sparse affinity matrixes A_1 and A_2 by Eq. (35) in [16]. Following [12], we randomly select a ratio of examples to be partial to simulate the partial view setting, i.e. they only appear in either of the two views while the remaining ones are described by both views. we evenly assign them to the two views to simplify the experiment. Each time we randomly select 10% to 90% examples, with 10% as interval, as partial examples. We repeat such process 10 times and record the average and standard deviation results. For PVC and IMG, the k-means algorithm is performed to get the final clustering result as in the original paper. The other clustering methods may be also feasible. We utilize two standard clustering evaluation metric to measure the multi-view clustering performance, that is, Clustering Accuracy (ACC) and Normalized Mutual Information (NMI). Same as [12], we test all the methods under different Partial Example Ratio (PER) varying from 0.1 to 0.9 with an interval of 0.1.

5.3 Clustering Result Comparison

Table 2 and Fig. 3 report the ACC and NMI values respectively on various data sets with different PER ratio settings. From these figures and the table, we make the following observations and discussions.

In almost all the settings, our method usually outperform both methods executed on single complete view (V1CLR and V2CLR) even when the PER ratio equals 30% , which confirm that our approach synthesizes the information from both views validly and proposed constraint between the subgraphs corresponding to the same samples (from different views) is effective. As the partial example ratio PER varies from 10% to 90%, the proposed method usually performs much better than other multi-view clustering baselines. Particularly, the performance of our approach improves much compared with the baselines when Per is less than 40%. And with more missing examples, the performance of all the methods drops basically.

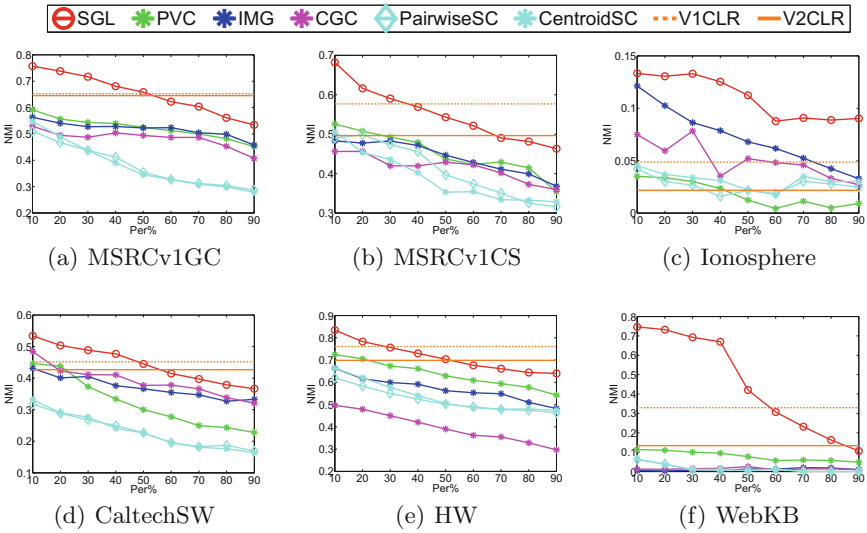
CentroidSC and PairwiseSC usually perform worst on almost all the data sets, which may be caused by that matrix completion requires the randomness of the missing locations, while the data are missing block-wise for the multi-view incomplete data setting. One may be curious why our method performs much better than other subspace learning methods. This may be caused by that their assumption that shared data in different views have exactly the same representations in the latent space may damage the inner structure of each view and increase the risk of over-fitting. Besides, our graphs are learned from both

Table 2. Experimental ACC (the higher the better) results (mean(std)) on six data sets. The best result is highlighted in boldface. T-Test (statistical significance of T-Test is 5%) results between our and other algorithms, Win (•) means our performs better. Lose (⊗) means other algorithm performs better. Tie (⊙) means that our and other algorithms cannot outperform each other.

Data sets	V1CLR	V2CLR	PER	CentroidSC	PairwiseSC	PVC	IMG	CGC	SGL
MSRCv1GC	.7524	.6619	0.1	.6499(.0399)•	.6160(.0598)•	.6879(.0191)•	.6952(.0182)•	.6271(.0487)•	**.8300(.0430)**
			0.2	.6030(.0370)•	.5752(.0382)•	.6510(.0414)•	.6749(.0294)•	.5924(.0349)•	**.8238(.0365)**
			0.3	.5480(.0329)•	.5494(.0223)•	.6482(.0429)•	.6645(.0332)•	.5952(.0435)•	**.8090(.0282)**
			0.4	.4979(.0354)•	.5165(.0296)•	.6403(.0246)•	.6748(.0445)•	.5876(.0664)•	**.7852(.0313)**
			0.5	.4568(.0246)•	.4619(.0297)•	.6302(.0442)•	.6564(.0381)•	.5986(.0583)•	**.7657(.0279)**
			0.6	.4329(.0371)•	.4339(.0361)•	.5848(.0313)•	.6605(.0438)•	.5848(.0743)•	**.7371(.0413)**
			0.7	.4134(.0296)•	.4199(.0318)•	.5926(.0407)•	.6415(.0470)•	.5771(.0436)•	**.7224(.0440)**
			0.8	.4014(.0347)•	.4043(.0298)•	.5758(.0233)•	.6250(.0593)•	.5652(.0552)•	**.6810(.0341)**
			0.9	.3733(.0243)•	.3830(.0246)•	.4948(.0297)•	.5505(.0711)•	.5152(.0995)•	**.6368(.0310)**
MSRCv1CS	.6762	.5667	0.1	.6231(.0269)•	.6193(.0232)•	.6305(.0331)•	.5983(.0270)•	.5286(.0490)•	**.7895(.0071)**
			0.2	.5604(.0328)•	.6125(.0423)•	.6228(.0367)•	.5600(.0244)•	.5000(.0231)•	**.7067(.0415)**
			0.3	.5521(.0209)•	.6036(.0364)•	.5823(.0682)•	.5628(.0468)•	.5667(.0554)•	**.7210(.0568)**
			0.4	.5129(.0423)•	.5757(.0240)•	.5552(.0550)•	.5611(.0341)•	.5562(.0600)⊙	**.6762(.0516)**
			0.5	.4629(.0352)•	.5175(.0395)•	.5386(.0735)⊙	.5541(.0731)⊙	.5714(.0288)•	**.6381(.0297)**
			0.6	.4525(.0376)•	.4859(.0390)•	.5550(.0460)•	.5570(.0445)⊙	.6067(.0338)⊙	**.6390(.0701)**
			0.7	.4268(.0380)•	.4433(.0277)•	.5294(.0764)⊙	.5171(.0546)•	.5467(.0407)•	**.6143(.0370)**
			0.8	.4149(.0321)•	.4090(.0208)•	.5496(.0269)⊙	.5518(.0550)⊙	.5267(.0605)⊙	**.5981(.0300)**
			0.9	.3972(.0165)•	.4017(.0332)•	.4605(.0218)•	.4890(.0643)•	.4990(.0652)•	**.5848(.0637)**
Ionosphere	.6097	.5726	0.1	.6407(.0192)•	.6397(.0172)•	.5832(.0044)•	.5983(.0270)•	.5286(.0490)•	**.7179(.0083)**
			0.2	.6270(.0251)•	.6172(.0287)•	.5770(.0125)•	.5600(.0244)•	.5000(.0231)•	**.7179(.0110)**
			0.3	.6227(.0351)•	.6076(.0379)•	.5683(.0075)•	.5628(.0468)•	.5667(.0554)•	**.7179(.0252)**
			0.4	.6101(.0234)•	.6180(.0332)•	.5693(.0085)•	.5611(.0341)•	.5562(.0600)•	**.7174(.0174)**
			0.5	.5907(.0354)•	.5960(.0367)•	.5648(.0053)•	.5541(.0731)•	.5714(.0288)•	**.7020(.0162)**
			0.6	.5870(.0331)•	.5887(.0260)•	.5730(.0088)•	.5570(.0445)•	.6067(.0338)•	**.6923(.0225)**
			0.7	.5744(.0174)•	.5855(.0206)•	.5794(.0056)•	.5171(.0546)•	.5467(.0407)•	**.6917(.0219)**
			0.8	.5532(.0188)•	.5670(.0345)•	.5652(.0140)•	.5518(.0550)•	.5267(.0605)•	**.6883(.0235)**
			0.9	.5558(.0351)•	.5554(.0382)•	.5656(.0049)•	.4890(.0643)•	.4990(.0652)•	**.6724(.0309)**
CaltechSW	.5578	.4671	0.1	.4221(.0339)•	.3976(.0197)•	.5189(.0258)•	.5472(.0118)•	.5757(.0280)⊙	**.5912(.0137)**
			0.2	.4011(.0231)•	.3926(.0189)•	.5128(.0209)•	.5243(.0596)•	.5508(.0769)⊙	**.5821(.0143)**
			0.3	.3891(.0225)•	.3709(.0212)•	.4903(.0366)•	.5365(.0243)•	.5054(.0606)•	**.5653(.0245)**
			0.4	.3639(.0266)•	.3604(.0201)•	.4645(.0241)•	.5060(.0670)⊙	.5279(.0636)⊙	**.5440(.0364)**
			0.5	.3506(.0225)•	.3457(.0224)•	.4325(.0175)•	.5033(.0508)⊙	**.5086(.0467)**⊙	.5075(.0291)
			0.6	.3307(.0296)•	.3344(.0303)•	.4226(.0345)•	.4901(.0414)⊙	**.5159(.0324)**⊙	.4912(.0270)
			0.7	.3230(.0262)•	.3194(.0265)•	.3988(.0274)•	.4657(.0530)⊙	**.4968(.0644)**⊙	.4776(.0196)
			0.8	.3205(.0248)•	.3251(.0222)•	.4053(.0405)•	.4546(.0361)•	.4649(.0340)⊙	**.4794(.0186)**
			0.9	.3098(.0235)•	.3100(.0256)•	.3890(.0297)•	.4482(.0359)⊙	.4422(.0250)⊙	**.4583(.0339)**
HW	.7745	.7020	0.1	.7041(.0060)•	.6380(.0169)•	.5629(.0204)•	.5893(.0403)•	.6718(.0527)•	**.8231(.0049)**
			0.2	.6631(.0131)•	.5924(.0108)•	.5649(.0279)•	.5341(.0387)•	.6318(.0678)•	**.7862(.0230)**
			0.3	.6145(.0075)•	.5547(.0122)•	.5201(.0224)•	.5215(.0148)•	.6058(.0473)•	**.7763(.0144)**
			0.4	.5734(.0067)•	.5254(.0080)•	.4937(.0123)•	.5011(.0513)•	.5937(.0708)•	**.7655(.0101)**
			0.5	.5325(.0114)•	.4964(.0086)•	.4601(.0163)•	.4835(.0199)•	.5674(.0552)•	**.7613(.0186)**
			0.6	.5048(.0098)•	.4802(.0077)•	.4526(.0144)•	.4563(.0273)•	.5408(.0486)•	**.7441(.0150)**
			0.7	.4941(.0105)•	.4584(.0088)•	.4314(.0159)•	.4213(.0088)•	.5248(.0304)•	**.7351(.0165)**
			0.8	.4892(.0133)•	.4429(.0060)•	.4184(.0098)•	.4041(.0226)•	.5096(.0300)•	**.7267(.0099)**
			0.9	.4846(.0179)•	.4354(.0101)•	.4145(.0171)•	.3743(.0299)•	.4809(.0336)•	**.7272(.0271)**
WebKB	.8563	.8249	0.1	.7925(.0081)•	.8006(.0126)•	.7336(.0063)•	.2952(.3154)•	.7159(.0426)•	**.9701(.0030)**
			0.2	.7824(.0088)•	.7812(.0124)•	.7291(.0062)•	.2846(.3006)•	.6830(.0574)•	**.9678(.0051)**
			0.3	.7565(.0267)•	.7617(.0253)•	.7219(.0041)•	.3124(.3299)•	.6852(.0565)•	**.9615(.0074)**
			0.4	.7782(.0112)•	.7796(.0098)•	.7175(.0148)•	.3166(.3343)•	.6834(.0568)•	**.9574(.0097)**
			0.5	.7723(.0272)•	.7692(.0420)•	.6993(.0120)•	.3364(.3548)•	.6728(.0538)•	**.9006(.0302)**
			0.6	.7047(.0748)•	.7174(.0763)•	.6946(.0234)•	.3461(.3650)•	.6626(.0404)•	**.8680(.0260)**
			0.7	.5834(.0134)•	.5834(.0134)•	.6916(.0170)•	.3580(.3779)•	.6463(.0527)•	**.8418(.0231)**
			0.8	.5568(.0107)•	.5568(.0107)•	.6737(.0286)•	.3532(.3728)•	.6422(.0488)•	**.8092(.0313)**
			0.9	.5264(.0102)•	.5264(.0102)•	.6564(.0235)•	.3541(.3737)•	.6197(.0566)•	**.7857(.0250)**
win\tie\lose				54 \ 0 \ 0	54 \ 0 \ 0	51\ 3 \ 0	46 \ 8 \ 0	43 \ 11 \ 0	

the initial graphs and the consistency between views. And the learned graphs contain ideal clustering structures.

One may be also interested in the reason why our method has a considerable improvement when PER is high. When PER is high, for most of the subspace learning methods, it is hard to accurately estimate the common representation P_c simply from the little common complete data. IMG utilizes a general global structure to remedy this deviation. However, the global structure overlooks the specific intrinsic structure of each individual view. Our structures are more detailed and specific.



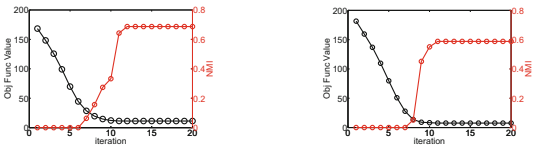
(a) MSRCv1GC (b) MSRCv1CS (c) Ionosphere

(d) CaltechSW (e) HW (f) WebKB

Fig. 3. The NMI (the higher the better) results for the six data sets. PER (partial example ratio) is the ratio of partial examples. Partial examples are evenly distributed to the two views.

5.4 Convergence Study

We experiment on data set MSRCv1 to show the convergence property. The convergence curves and corresponding NMI performances with $PER = 30\%$ and $PER = 70\%$ setting are plotted in Fig. 4. For $PER = 30\%$ setting, we set $\{\lambda 1, \lambda 2, \mu\}$ as $\{50, 50, 2.0\}$. For $PER = 70\%$ setting, we set $\{\lambda 1, \lambda 2, \mu\}$ as $\{50, 50, 9.0\}$. The objective function during each iteration is drawn in black. We can see that the value of objective function decreases rapidly with the increasing of iteration round. Inspiringly, it only takes around 12 rounds to converge. The NMI value during each iteration is drawn in red, from which we can see 12 round is enough to get good clustering performance.



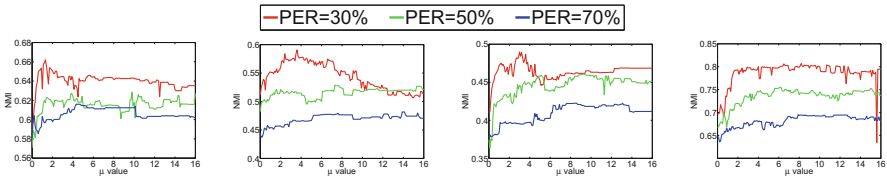
(a) $PER = 30\%$ (b) $PER = 70\%$

Fig. 4. Convergence curve of Objective function value and corresponding NMI performance curve *vs* number of iterations of our with $PER = 30\%$ and $PER = 70\%$ on MSRCv1GC data set. (Color figure online)

5.5 Parameter Study

We study parameters on four data sets: MSRCv1GC, MSRCv1SW CaltechSW and HW. There are three parameters to explore: λ_1, λ_2 and μ. Following [16], we determined both λ_1 and λ_2 in a heuristic way : in each iteration, we computed the numbers of zero eigenvalues of L_{S1} and L_{S2}, if one is larger (smaller) than k, we divide (multiply) it by two (respectively). Following [12], We tune μ for three different PER 30%, 50% and 70%. As above experiment, we randomly select a ratio of examples to be partial and repeat such process 10 times to record the average results. The effect of the parameter μ is showed in Fig. 5.

From Fig. 5, it is easy to see that on all data sets, our method achieves steadily good performance for NMI with a very large range of μ in all settings, which validates the robustness of our method. Our method usually has a relatively good performance when μ is in the range of $[2.0, 4.0]$ for the $PER = 30\%$ and $PER = 50\%$ settings, while the best range becomes $[7.0, 9.0]$ for the $PER = 70\%$ setting.



(a) MSRCv1GC (b) MSRCv1SW (c) CaltechSW (d) HW

Fig. 5. Effect of the parameter μ on four data set with three different PER.

6 Conclusion

In this paper, we propose a method to handle multi-view clustering problem in the case that all the views suffer from the missing of some data. Different from existing approaches, we simultaneously manipulate and cluster incomplete multi-view data. We excavate and maintain the intrinsic structure of each individual view, and establish interaction between the different views through the shared data. For each view, a graph with exactly c connected components can be learned so that the clustering results can be derived from graphs without any post-clustering. To optimize our proposed objective, we provide the solution which can simultaneously handle both the whole and partial regularization problem. Experimental results on six real-world multi-view data sets compared with several baselines validate the effectiveness of our method. In the future, we will study how to reduce the computational cost of our method.

References

1. Bickel, S., Scheffer, T.: Multi-view clustering. In: Proceedings of the 4th IEEE International Conference on Data Mining (ICDM), pp. 19–26 (2004)
2. Brand, M.: Incremental singular value decomposition of uncertain data with missing values. In: Computer Vision - ECCV 7th European Conference on Computer Vision, pp. 707–720 (2002)
3. Cheng, W., Zhang, X., Guo, Z., Wu, Y., Sullivan, P.F., Wang, W.: Flexible and robust co-regularized multi-domain graph clustering. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 320–328 (2013)
4. Chung, F.R.: Spectral Graph Theory, vol. 92. American Mathematical Society, New York (1997)
5. Fan, K.: On a theorem of Weyl concerning eigenvalues of linear transformations i. Proc. Natl. Acad. Sci. **35**(11), 652–655 (1949)
6. Greene, D., Cunningham, P.: A matrix factorization approach for integrating multiple data views. In: Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, pp. 423–438 (2009)
7. Guo, Y.: Convex subspace representation learning from multi-view data. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence (2013)
8. Huang, J., Nie, F., Huang, H.: A new simplex sparse learning model to measure data similarity for clustering. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence IJCAI, pp. 3569–3575 (2015)
9. Huang, J., Nie, F., Huang, H., Lei, Y., Ding, C.H.Q.: Social trust prediction using rank-k matrix recovery. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence IJCAI, pp. 2647–2653 (2013)
10. Kumar, A., Rai, P., Daume, H.: Co-regularized multi-view spectral clustering. In: Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems, pp. 1413–1421 (2011)
11. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
12. Li, S., Jiang, Y., Zhou, Z.: Partial multi-view clustering. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 1968–1974 (2014)
13. Lin, Z., Chen, M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055 (2010)
14. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization. In: Proceedings of the 2013 SIAM International Conference on Data Mining, pp. 252–260. SIAM (2013)
15. Mohar, B., Alavi, Y., Chartrand, G., Oellermann, O.: The laplacian spectrum of graphs. Graph theory, combinatorics, and applications **2**(871–898), 12 (1991)
16. Nie, F., Wang, X., Jordan, M.I., Huang, H.: The constrained laplacian rank algorithm for graph-based clustering. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 1969–1976 (2016)
17. Shao, W., He, L., Yu, P.S.: Multiple incomplete views clustering via weighted nonnegative matrix factorization with $L_{2,1}$ regularization. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Soares, C., Gama, J., Jorge, A. (eds.) ECML PKDD 2015. LNCS (LNAI), vol. 9284, pp. 318–334. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23528-8_20
18. Shao, W., Shi, X., Philip, S.Y.: Clustering on multiple incomplete datasets via collective kernel learning. In: 2013 IEEE 13rd International Conference on Data Mining (ICDM), pp. 1181–1186. IEEE (2013)

19. Wang, Q., Si, L., Shen, B.: Learning to hash on partial multi-modal data. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI, pp. 3904–3910 (2015)
20. Zhao, H., Liu, H., Fu, Y.: Incomplete multi-modal visual data grouping. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI, pp. 2392–2398 (2016)