Neurocomputing 366 (2019) 1–11

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Auto-weighted multi-view constrained spectral clustering

Chuan Chen[a], Hui Qian[a], Wuhui Chen[a], Zibin Zheng[a,*], Hong Zhu[b]

[a] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China
[b] Faculty of Science, Jiangsu University, Zhenjiang, China

ARTICLE INFO

Article history:
Received 24 September 2018
Revised 13 May 2019
Accepted 30 June 2019
Available online 23 July 2019

Communicated by Dr Zenglin Xu

Keywords:
Constrained clustering
Spectral clustering
Multi-view learning
Auto-weight learning

ABSTRACT

Constrained clustering is a new fashion of semi-supervised learning which focused on enhancing the quality of the partition by utilizing pairwise constraints. Though many constrained clustering methods have an excellent performance in single-view clustering, they can't be directly applied to multi-view scenario. In this paper, we propose a novel constrained spectral clustering approach for multi-view data, which explicitly imposes pairwise constraints as a series of linear constraints on the unified indicator matrix. To our best knowledge, this is the first work on multi-view constrained spectral clustering. Our approach can differ the importance of different views via the auto-weight learning strategy. Simultaneously, the views which contain much noisy or irrelevant information are also automatically eliminated, thereby improving the prediction performance. Extensive experiments conducted on various multi-view datasets demonstrate that the proposed approach can efficiently utilize pairwise constraints and outperforms the state-of-the-art approaches.

1. Introduction

In recent years, constrained clustering, a new fashion of semi-supervised clustering algorithm, has been successfully studied in plenty of real-world applications, such as GPS-based map refinement [1], person identification [2], community detection [3] and so on. Benefiting from pairwise constraints, a kind of supervised information, constrained clustering algorithms can achieve a big improvement over unsupervised clustering algorithms. A pairwise constraint including "must-links (ML)" and "cannot-links" (CL) between two instances indicates whether they belong to the same cluster or not. These constraints occur in a variety of applications and domains. For example, social networks contain not just "trust" link (ML) but also "distrust" link (CL) between two users [4]; in co-citation network, if we find the keywords of two papers are exactly similar, we can denote the relationship of these two papers as ML. On top of that, pairwise constraints are obtained with less human effort [5] than other supervised information, such as class labels. In other words, these constraints can be collected more easily and conveniently thus the cost of constrained clustering will be much less than other semi-supervised clustering algorithms. In many real-world applications, data are dramatically collected from multiple modalities or presented by multiple representations with the rapid development of the Internet and technology. For example, the web page includes the content and the hyperlink [6]; pictures can either be described by their color or texture feature. Nevertheless, many constrained clustering methods can only be applied to data with a single modality [7–9]. Thus, how to effectively and efficiently utilize pairwise constraints for multi-view clustering remains a crucial problem to explore.

In essence, pairwise constraints don't provide explicit class information and are not possible to infer class label directly. This property of pairwise constraints increases the difficulty of utilizing them for clustering. In multi-view scenario, besides the instances share the latent consistent clustering in all views, pairwise constraints also ought to be coherent. From these perspectives, it's arduous to efficiently incorporate pairwise constraints to multi-view clustering. So far, only a handful of studies have provided solutions. In [10], Eaton et al. presented a Co-EM algorithm. It iterated the E-step in one view to utilize the constraints through constrained k-means (a single view constrained clustering method) followed by the M-step in the other view to transfer these constraints and update the clustering. Inspired by label propagation, Fu et al. [11] enforced pairwise constraints via horizontal and vertical constraint propagation. In [12], Zhao et al. proposed multi-view matrix completion (MVMC) for multi-view clustering with pairwise constraints, which naturally encoded the ML and CL constraints to an observed similarity matrix. In addition, it's worthy to mention that the available pairwise constraints are still extremely scarce in reality. It becomes the second puzzle

* Corresponding author.
E-mail addresses: chenchuan@mail.sysu.edu.cn (C. Chen), qianh6@mail2.sysu.edu.cn (H. Qian), chenwuh@mail.sysu.edu.cn (W. Chen), zhzibin@mail.sysu.edu.cn, zibinzheng@yeah.net (Z. Zheng), zhuhongmath@126.com (H. Zhu).

https://doi.org/10.1016/j.neucom.2019.06.098
0925-2312/

to multi-view constrained clustering. Existing MVMC method [12] suffered a failure and performed poorly due to the limitation of matrix completion with few pairwise constraints. Furthermore, since different views provide partial information on the underlying structure, their effect on clustering varies from each other. Obviously, we cannot ignore the discrepancy across views. It's a pity that most multi-view constrained clustering methods can't differ the importance of different views and may tend to perform poorly.

In this paper, we propose an auto-weighted multi-view semi-supervised clustering approach with pairwise constraints called Multi-View Constrained Spectral Clustering (*MVCSC*). Firstly, MVCSC explicitly imposes ML and CL constraints as a set of linear constraints on the unified indicator matrix. This process effectively incorporates pairwise constraints to clustering. Concurrently, it guarantees the consistency of pairwise constraints in multi-view scenario. To trade off between the number of satisfied constraints and the quality of partition, MVCSC uses L_1 regularizer on penalty vector of linear constraints in the objective function. MVCSC also introduces an auto-weight learning strategy to differ the importance of different views, thereby improving the prediction performance. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we propose the first multi-view constrained spectral clustering method in which efficiently utilizes pairwise constraints.
- We present an auto-weighted learning strategy, such that the view weights are calculated automatically and the irrelevant views will be eliminated.
- We propose an effective alternating optimization method to solve the objective function.
- We conduct extensive experiments on various multi-view datasests, which demonstrate the advantages of our technique over the state-of-the-arts approaches.

The rest of the paper is organized as follows. We give a brief review of related work in Section 2. In Section 3, the basic notations and background are introduced firstly, then the multi-view constrained spectral clustering is presented. In Section 4, we give an iterative algorithm to optimize the objective function. All the experimental results are shown in Section 5. Finally, we conclude our work in Section 6.

2. Related work

During the past few years, graph-based learning [13–15] has attracted much attention. Clustering is one of the fundamental unsupervised learning techniques in machine learning and data mining [16,17]. *Spectral clustering* is a well-known graph-based clustering algorithm that considers each data point as a node and each relationship as an edge in the constructed graph. In recent years, spectral clustering has become one of the most popular clustering algorithms due to its adaptation in arbitrary data distribution and well-defined mathematical framework [18]. For relaxing the NP-hard problem of spectral clustering, it has two classical transformation based on graph cut theory: RatioCut [19] and the normalized cut (Ncut) [20]. When accessing to multi-view data collected from multiple sources or represented by multiple representation, multi-view spectral clustering is able to learn a consistent clustering from multiple graphs (i.e., views). Kumar et al. [21] proposed a spectral clustering framework based on co-regularization that make the clustering hypotheses on different graphs agree with each other. Another well-know multi-view spectral clustering approach based on co-training was proposed in [22]. However, this kind of methods failed to differ the importance of different graphs and may behave poorly when an unreliable view is added. To address this issue, some approaches adaptively learn a weight for each graph during the optimization including the methods in [23–25]. Besides multi-view spectral clustering, the group of multi-view clustering methods also includes multi-view subspace clustering [26], multi-view nonnegative matrix factorization clustering [27], multi-kernel based multi-view clustering [28] and so on. More advanced multi-view clustering approaches can refer to [29].

Single view constrained clustering aims at partitioning with pairwise constraints in single view scenario. Current approaches, mostly based on classical clustering methods, have been successfully studied. Wagstaff et al. [1] firstly incorporated pairwise constraints to the K-means algorithm. Subsequently, other constrained K-means approaches were proposed [7–9]. Wang et al. [30] developed a nonnegative matrix factorization method with pairwise constraints. Significantly, constrained spectral clustering is an area of active research within the broad domain of constrained clustering. In this field, a great many methods have been proposed. One of the ways is spectral learning [31] which sets the similarity between a pair of constrained points to 1 for ML or to 0 for CL. This method only adjust similarity between constrained points without constraint propagation. The opposite is that, Lu et al. [32] proposed a constraint propagation framework that the unconstrained data can be constrained by learning the pairwise similarity. Similarly, Wang et al. [33] proposed a self-teaching framework for constraint propagation. In [9], a flexible constrained spectral clustering was presented, which explicitly encoded pairwise constraints as a new Laplacian matrix and could be solved in polynomial time. However, this method is valid only in two-class partition case. Another work in [34] which imposed pairwise constraints as a series of linear constraints suffered the same limitation.

Multi-view constrained clustering is a class of constrained clustering on multi-view data. Due to the difficulty of incorporating pairwise constraints to multi-view clustering, pairwise constraints are rarely studied in multi-view scenario. In [10], Eaton et al. proposed a Co-EM algorithm which iteratively propagated the constraints on the one view via constrained K-means at the E-step which followed by the M-step in the other view to transfer those constraints and update the clustering. Nevertheless, this method obtained a clustering result for each view rather a consistent partition from multiple views. Inspired by label propagation, Fu et al. [11] proposed the multi-modal constraint propagation for image clustering. However, this method needed an extra step to obtain the final results by clustering on the similarity matrix recomputed by constraint propagation result, and the view weights were given manually while inaccessible in reality. Considering the tight relationship between pairwise constraints and similarity, Zhao et al. [12] firstly proposed multi-view matrix completion that encoded the ML and CL constraints as an observed similarity matrix, i.e., transferring the constrained problem to matrix completion problem. However, this method performed poorly when the amount of constraints is extremely scarce. The reason is that matrix completion cannot obtain a satisfying performance with a small number of pairwise constraints (observed information). Furthermore, the aforementioned multi-view constrained clustering methods can't distinguish the significance of different views and may be inclined to perform poorly if an unreliable view is involved.

To the best of our knowledge, we are presenting the first multi-view constrained spectral clustering approach, which is able to handle the issues in the existing work.

3. Methodology

In this section, we will briefly introduce the notations and get started from the spectral clustering framework. Then, the multi-view spectral clustering assisted with pairwise constraints is presented.

3.1. Background and notations

Let $\boldsymbol{X} = [x_1, x_2, \ldots, x_N]^\top \in \mathbb{R}^{N\times d}$ denote the data matrix, where N is the number of instances and d is dimension of feature. Given the whole data matrix $\boldsymbol{X}$, an undirected weighted graph $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W}\}$ can be constructed, where each instance is presented as a vertex $v \in \boldsymbol{V}$ and each edge $e \in \boldsymbol{E}$ represents the affinity relation of a pair of vertices. $\boldsymbol{W}$ is the weighted adjacency matrix of the graph. In practice, to model the local neighborhood relationship between instances, the k-nearest neighbor graph is usually adopted. Specifically, x_i and x_j are connected if x_j is among the k-nearest neighbors of x_i while they are unconnected otherwise. After connecting the appropriate vertices, we weight the edges by the similarity between their endpoints.

Spectral clustering is an extensively used graph partitioning algorithm. Obviously, it's necessary to find a partition that the edges between different clusters have a low weight and the edges within the same cluster have a high weight. To reach this purpose, the normalized cut (Ncut) [20], one of the most representative methods of spectral clustering, was proposed based on graph minimum cut. The objective function of Ncut for two-class partition can be formulated as

$$\min_{\boldsymbol{f}^\top \boldsymbol{f}=1, \boldsymbol{f}^\top \boldsymbol{1}=0} \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f},$$

where $\boldsymbol{f} = [f_1, f_2, \ldots, f_N]^\top \in \mathbb{R}^{N\times 1}$ is the class indicator vector of all data and $\boldsymbol{1}$ is a vector with all elements being 1. For multi-class partition, Ncut can be written as

$$\min_{\boldsymbol{F}^\top \boldsymbol{F}=\boldsymbol{I}} Tr(\boldsymbol{F}^\top \boldsymbol{L} \boldsymbol{F}),$$

where $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_c] \in \mathbb{R}^{N\times c}$ is the indicator matrix and each row of $\boldsymbol{F}$ is the indicator vector of one instance, c is the number of class. $\boldsymbol{L}$ denotes the normalized graph Laplacian matrix, defined as $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}$. $\boldsymbol{D}$ is the degree matrix whose ith diagonal element is $d_{ii} = \sum_{j=1}^{N} \boldsymbol{W}_{ij}$, $\boldsymbol{I}$ is the identity matrix and $Tr(\boldsymbol{A})$ denotes the trace of matrix $\boldsymbol{A}$.

For multi-view data, let K be the number of views and $\{\boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^K\}$ be the data matrix of all the views, where $\boldsymbol{X}^k \in \mathbb{R}^{N\times d^k}$ and d^k represent data features and dimension of the kth view respectively. We can construct the K k-nearest neighbor graphs $\{\boldsymbol{G}^1, \ldots, \boldsymbol{G}^K\}$ and corresponding normalized Laplacian matrix $\{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_K\}$. Two vital questions need to be answered by multi-view approaches are how to reach a consensus of clustering and how to express the relationship of all views. In this paper, we linearly combine different graphs with weight $\mu_k (k = 1, 2, \ldots, K)$. And we further constrain indicator matrix $\boldsymbol{F}$ to be a unified one across all the views and minimize the Ncut for multiple graphs. In view of that many multi-view approaches always give nonzero weights to graphs [23,24], we add L_2 regularizer of weighting coefficients in the objective function to make view weights sparse. That is, a part of views has nonzero weights and the weights of the rest are equal to zero. This important property provides two advantages. Firstly, the graphs contain much noisy or irrelevant information are eliminated when integrating multiple graphs by estimating their weights at zero. Which will improve the performance of clustering. Secondly, the importance of the rest views can be identified by the value of nonzero weighting coefficients. Where the larger weight indicates higher importance. Thus, the multi-view spectral framework can be modeled as

$$\min_{\boldsymbol{F}^\top \boldsymbol{F}=\boldsymbol{I}} \sum_{k=1}^{K} \frac{\mu_k}{2} Tr(\boldsymbol{F}^\top \boldsymbol{L}_k \boldsymbol{F}) + \frac{\beta}{2} \|\boldsymbol{\mu}\|_2^2$$

$$s.t. \quad \boldsymbol{\mu} \geq 0, \boldsymbol{\mu}^\top \boldsymbol{1} = 1,$$

where the weight coefficient vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_K]^\top$ is nonnegative and the sum of view weights is equal to 1.

3.2. MVCSC for two-class partition

To efficiently deal with pairwise constraints, we impose ML and CL constraints as a series of linear constraints with the form $\boldsymbol{C}\boldsymbol{f} = \boldsymbol{0}$, where $\boldsymbol{C} \in \mathbb{R}^{M\times N}$ is the constraint matrix and M is the number of pairwise constraints. ML and CL constraint can be encoded in rows of $\boldsymbol{C}$ of the form

$(0, \ldots, 0, -1, 0, \ldots, 0, +1, 0, \ldots, 0)$ (Must-Link)

$(0, \ldots, 0, +1, 0, \ldots, 0, +1, 0, \ldots, 0)$ (Cannot-Link).

$\boldsymbol{C}$ can be regarded as a relationship matrix "translated" from the constrained graph where each ML is a positive edge and each CL is a negative edge. We assume that the mth row of constraint matrix $\boldsymbol{C}_m$ represents the relation between x_i and x_j. If they belong to the set of ML constraints, $\boldsymbol{C}_m \boldsymbol{f} = f_i - f_j$ is equal to 0. It conveys that two different instances in the same cluster have the same indicator value. In contrast, if they belong to CL constraints, $\boldsymbol{C}_m \boldsymbol{f} = 0$ controls that x_i and x_j have opposite indicator value.

For multi-view constrained clustering, pairwise constraints are required to be coherent across all the views. In this paper, MVCSC constrains the unified indicator $\boldsymbol{f}$ to satisfy a series of ML and CL constraints through constraint function $\boldsymbol{C}\boldsymbol{f} = \boldsymbol{0}$. In other words, pairwise constraints are effectively used to guide multi-view clustering. Hence, the unified pairwise constraints are built. It can be also shown in the optimization that instance labels are predicted when iteratively updating indicator $\boldsymbol{f}$ under the constraint function. Therefore, the multi-view constrained spectral clustering can be represented by appending the constraint function,

$$\min_{\boldsymbol{f}, \boldsymbol{\mu}} \sum_{k=1}^{K} \frac{\mu_k}{2} \boldsymbol{f}^\top \boldsymbol{L}_k \boldsymbol{f} + \frac{\beta}{2} \|\boldsymbol{\mu}\|_2^2$$

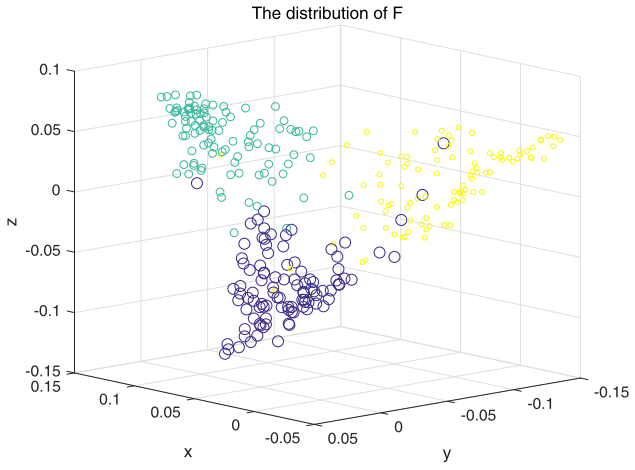
$$s.t. \quad \boldsymbol{f}^\top \boldsymbol{f} = 1, \ \boldsymbol{f}^\top \boldsymbol{1} = 0, \ \boldsymbol{C}\boldsymbol{f} = \boldsymbol{0}, \ \boldsymbol{\mu} \geq 0, \ \boldsymbol{\mu}^\top \boldsymbol{1} = 1.$$

In practice, utilizing all pairwise constraints with the form of $\boldsymbol{C}\boldsymbol{f} = \boldsymbol{0}$ has two weaknesses. Firstly, this equality constraint is excessively tight for clustering. In general, we just want the instances in the same group to have similar indicator values and instances in the different clusters to have distinct indicator values for clustering. The same or opposite constraints may overly distort the clustering. Secondly, it cannot handle the case where some constraints are uncertain or speculative due to noise or other factors. Thus, we wish for a problem setup which relaxes the equality constraint and minimizes the violation of the constraints while minimizing the consensus via Ncut for multiple graphs. This would lead to the following model, where we rewrite the linear constraint function as $\boldsymbol{C}\boldsymbol{f} = \boldsymbol{z}$ and add L_1 regularizer on the penalty vector $\boldsymbol{z}$ in the objective function. Using L_1 regularization, on the one hand, relaxes the tight constraints. On the other hand, it encourages $\boldsymbol{z}$ as sparse as possible to reduce the number of violated constraints and concurrently reduce the impact of noise. The new MVCSC can be modeled as

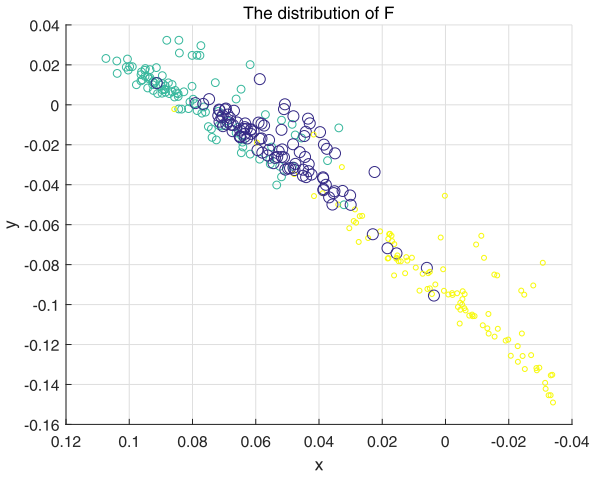
$$\min_{\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\mu}} \sum_{k=1}^{K} \frac{\mu_k}{2} \boldsymbol{f}^\top \boldsymbol{L}_k \boldsymbol{f} + \gamma \|\boldsymbol{z}\|_1 + \frac{\beta}{2} \|\boldsymbol{\mu}\|_2^2$$

$$s.t. \quad \boldsymbol{f}^\top \boldsymbol{f} = 1, \ \boldsymbol{f}^\top \boldsymbol{1} = 0, \ \boldsymbol{C}\boldsymbol{f} = \boldsymbol{z}, \ \boldsymbol{\mu} \geq 0, \ \boldsymbol{\mu}^\top \boldsymbol{1} = 1.$$

To integrate the multiple views, the first term of the objective function linearly combines K graphs by using weight coefficients $\boldsymbol{\mu}$. The second term is the L_1 regularization of penalty vector $\boldsymbol{z}$. Here γ is a user-selected parameter which controls the degree of sparsity of $\boldsymbol{z}$. The greater γ indicates that the penalty vector $\boldsymbol{z}$ is more sparse, i.e., more pairwise constraints are satisfied while the linear constraints are tighter. Thus, we need choose a appropriate



(a) 3D diagram of $\boldsymbol{F}$



(b) 2D mapping diagram of $\boldsymbol{F}$

Fig. 1. An example of indicator matrix $\boldsymbol{F}$ for three-class partition. We can see that some instances in purple group and green group have similar value of x and dissimilar y and z.

γ to balance the number of satisfied constraints and constraint intensity. The third term of the objective function is a L_2 regularizer that controls the view selection with sparse weight coefficients $\boldsymbol{\mu}$. When the kth graph is useful for clustering, μ_k is large. Otherwise, when the kth view contains much noise or irrelative information, μ_k tends to be zero (see Theorem 1). The parameter β controls the sparsity of $\boldsymbol{\mu}$. We can see from (5) that $\boldsymbol{\mu}$ tends to have only one nonzero entry with small β, while all entries in $\boldsymbol{\mu}$ tend to be $1/K$ with large β. We won't identify the importance of views under these two cases. Between two extremes, we can obtain sparse weighting coefficients $\boldsymbol{\mu}$ where only some entries of $\boldsymbol{\mu}$ are nonzero. In addition, we can see that the larger v_i will make the weight of ith views to be zero with greater probability. Where the graph with larger v_i contains much more noisy or irrelevant information according to $v_i = Tr(\boldsymbol{F}^\top \boldsymbol{L}_i \boldsymbol{F})/2 = \sum_{i,j=1}^{N} \boldsymbol{W}_{ij}(\boldsymbol{F}_i - \boldsymbol{F}_j)^2/2$ [18]. Therefore, these less useful views will be more likely to be eliminated.

3.3. Extension to multi-class partition

The MVCSC method can be naturally extended to multi-class partition where $c > 2$. Let $\boldsymbol{F} \in \mathbb{R}^{N\times c}$ be the indicator matrix. We still encode pairwise constraints to constraint matrix $\boldsymbol{C}$ and use L_1 regularizer on the penalty vector $\boldsymbol{z}$. For ML, similar with two-class partition, each relaxed linear constraint of two different instances in the same group constrains indicator vectors of them to be as possible as similar. When it comes to CL constraints, we also handle with them through L_1 regularization. Firstly, CL constraints are relaxed and the indicator vectors of two instances in the different clusters are constrained to be dissimilar. Furthermore, one thing worth paying attention to is that the indicator matrix $\boldsymbol{F}$ is represented in multi-dimension space in the case of multi-class partition. This phenomenon implies that two different instances can be also divided to different groups although the indicator vectors of them are dissimilar in not all dimensions. An example of three-class partition is shown in Fig. 1. We can see that a part of instances in purple group and green group can be also divided into two different groups when they have dissimilar values of y and z but similar x. Nevertheless, the number of dissimilar dimensions can't be less than a certain value. The partition that doesn't follow this rule will be fail because it can't recognition the whole groups accurately. Obviously, we can further and slightly relax the linear constraints to achieve it. Thus, the CL are also effectively incorporated to multi-view clustering. The objective function for multi-class partitioning can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{F},\boldsymbol{Z},\boldsymbol{\mu}} \sum_{k=1}^{K} \frac{\mu_k}{2} Tr\left(\boldsymbol{F}^\top \boldsymbol{L}_k \boldsymbol{F}\right) + \gamma \|\boldsymbol{Z}\|_1 + \frac{\beta}{2}\|\boldsymbol{\mu}\|_2^2 \\ &s.t. \quad \boldsymbol{F}^\top \boldsymbol{F} = \boldsymbol{I}, \boldsymbol{C}\boldsymbol{F} = \boldsymbol{Z}, \boldsymbol{\mu} \geq 0, \boldsymbol{\mu}^\top \boldsymbol{1} = 1. \end{aligned} \tag{1}$$

4. Optimization

To address the optimization in (1), we develop an iterative algorithm to find the relative optimal solution with high probability. In theory, the problem (1) can be broken down into two subproblems, i.e., one subproblem estimating indicator matrix $\boldsymbol{F}$ for predicting instance labels and one subproblem automatically weighting the importance of different views with $\boldsymbol{\mu}$. To solve the prediction subproblem, we use the ADMM method [35] to alternatively update $\boldsymbol{F}$, $\boldsymbol{Z}$ and the lagrangian multiplier $\boldsymbol{\lambda}$. For the latter, we use the Lagrange Multiplier method to obtain weighting coefficients $\boldsymbol{\mu}$ with fixed $\boldsymbol{F}$ and $\boldsymbol{Z}$. Finally, we repeat these two processes until converges. After obtain the final indicator matrix $\boldsymbol{F}$, we apply K-means algorithm on $\boldsymbol{F}$ to find the best data partition. The proposed MVCSC is summarized in Algorithm 1 .

Algorithm 1: Algorithm for multi-view constrained spectral clustering.

Input: graph Laplacian $\boldsymbol{L}_k (k = 1, 2, \ldots, K)$, constraint matrix $\boldsymbol{C}$, parameter ρ, γ, β, number of class c.
Output: indicator matrix $\boldsymbol{F}$, weighting coefficients $\vec{\mu}$, final clustering results.

```
Initialize F orthogonally, initialize λ, μ⃗ randomly;
for iter = 1 to T_1 do
    Optimize F according to Algorithm 2;
    Optimize μ⃗ according to Algorithm 3;
end
Performing K-means on F to obtain final clustering results.
```

4.1. Estimating indicator matrix

When the weight vector $\boldsymbol{\mu}$ is fixed, the subproblem for estimating indicator matrix $\boldsymbol{F}$ can be defined as

$$\min_{\boldsymbol{F},\boldsymbol{Z}} \sum_{k=1}^{K} \frac{\mu_k}{2} Tr\left(\boldsymbol{F}^\top \boldsymbol{L}_k \boldsymbol{F}\right) + \gamma \|\boldsymbol{Z}\|_1$$
$$s.t. \quad \boldsymbol{CF} = \boldsymbol{Z}, \boldsymbol{F}^\top \boldsymbol{F} = \boldsymbol{I}. \tag{2}$$

We use the Alternating Direction Method of Multipliers (ADMM) [35] to optimize the problem (2). The corresponding Augmented Lagrangian expression on the equality constraint function $\boldsymbol{CF} = \boldsymbol{Z}$ is given by

$$L_\rho(\boldsymbol{F}, \boldsymbol{Z}; \boldsymbol{\lambda}) = \sum_{k=1}^{K} \frac{\mu_k}{2} Tr(\boldsymbol{F}^\top \boldsymbol{L}_k \boldsymbol{F}) + \gamma \|\boldsymbol{Z}\|_1 + \langle \boldsymbol{\lambda}, \boldsymbol{CF} - \boldsymbol{Z} \rangle + \frac{\rho}{2} \|\boldsymbol{CF} - \boldsymbol{Z}\|_F^2,$$

thus, the objective function (2) can be transformed to a scaled form

$$\min_{\boldsymbol{F}^\top \boldsymbol{F} = \boldsymbol{I}} L_\rho(\boldsymbol{F}, \boldsymbol{Z}; \boldsymbol{\lambda}),$$

where $\rho > 0$ is the penalty parameter and $\boldsymbol{\lambda}$ is the lagrangian multiplier.

Fixing $\boldsymbol{Z}$, *updating* $\boldsymbol{F}$. To optimize indicator matrix $\boldsymbol{F}$, we apply a feasible method for orthogonality constraints in [36]. At first, the Crank-Nicolson-like update scheme is used to preserve the orthogonality constraints. We define a skew-symmetric matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = \boldsymbol{G}\boldsymbol{F}^\top - \boldsymbol{F}\boldsymbol{G}^\top,$$

where $\boldsymbol{G} = \partial L_\rho(\boldsymbol{F}, \boldsymbol{Z}; \boldsymbol{\lambda}) / \partial \boldsymbol{F}$. By the Crank–Nicolson-like scheme, the new trial point $\boldsymbol{Y}$ can be determined,

$$\boldsymbol{Y}(\tau) = \boldsymbol{F} - \frac{\tau}{2} \boldsymbol{A}(\boldsymbol{F} + \boldsymbol{Y}(\tau)),$$

where τ is step size and $\boldsymbol{Y}(\tau)$ is give in the closed form:

$$\boldsymbol{Y}(\tau) = \boldsymbol{Q}\boldsymbol{F}; \quad \boldsymbol{Q} = \left(\boldsymbol{I} + \frac{\tau}{2}\boldsymbol{A}\right)^{-1} \left(\boldsymbol{I} - \frac{\tau}{2}\boldsymbol{A}\right).$$

The most important property of the curve $\boldsymbol{Y}$ is $\boldsymbol{Y}(\tau)^\top \boldsymbol{Y}(\tau) = \boldsymbol{F}^\top \boldsymbol{F}$ for all $\tau \in \mathbb{R}$. We can see that the indicator matrix $\boldsymbol{F}$ can be obtained through updating $\boldsymbol{Y}(\tau)$. Thus, in the second step, we use the Curvilinear Search Method with Barzilai-Borwein (BB) steps [37] to find a suitable step size τ and update $\boldsymbol{Y}(\tau)$. More details about this algorithm can be found in [36].

Fixing $\boldsymbol{F}$, *updating* $\boldsymbol{Z}$. For nonsmooth $L1$ regularization, we employ the Soft Thresholding in [38] to optimize $\boldsymbol{Z}$. The closed-form solution can be obtained as following,

$$\boldsymbol{Z} = \frac{1}{\rho} Shrink(\boldsymbol{\lambda} + \rho \boldsymbol{CF}, \gamma), \tag{3}$$

where $Shrink(x, y) = sign(x) \odot \max|\, x \,| - y, 0$.

Updating $\boldsymbol{\lambda}$.

$$\boldsymbol{\lambda} = \boldsymbol{\lambda} + \rho(\boldsymbol{CF} - \boldsymbol{Z}). \tag{4}$$

4.2. Auto-weighting for views

With the indicator vector $\boldsymbol{F}$ fixed, we update weighting coefficients $\boldsymbol{\mu}$ for automatically weighting the importance of views. Obviously, we can rewrite the objective function as follows:

$$\min_{\boldsymbol{\mu}} \boldsymbol{v}^\top \boldsymbol{\mu} + \frac{\beta}{2} \|\boldsymbol{\mu}\|_2^2$$
$$s.t. \quad \boldsymbol{\mu} \geq 0, \ \boldsymbol{\mu}^\top \boldsymbol{1} = 1, \tag{5}$$

where $\boldsymbol{v} = [v_1, v_2, \ldots, v_K]^\top$ is a K-by-1 vector and v_i represents $Tr(\boldsymbol{F}^\top \boldsymbol{L}_i \boldsymbol{F})/2$. Without loss of generality, we assume that the entries in $\boldsymbol{v}$ are sorted in increasing order, i.e., $v_1 \leq v_2 \leq \cdots \leq v_K$.

Theorem 1. *The optimal solution of the problem in (5) is analytically given by*

$$\mu_k = \begin{cases} \frac{\theta - v_k}{\beta} & k \leq P \\ 0 & k > P \end{cases}$$

where

$$P = \underset{k}{argmax}\, (\theta - v_k > 0)$$
$$\theta = \frac{\sum_{k=1}^{P} v_k + \beta}{P}.$$

Proof. (5) is a quadratic optimization problem and can be optimized by Lagrange Multiplier method.

$$L(\boldsymbol{\mu}, \boldsymbol{\eta}, \theta) = \boldsymbol{v}^\top \boldsymbol{\mu} + \frac{\beta}{2} \|\boldsymbol{\mu}\|_2^2 - \boldsymbol{\eta}^\top \boldsymbol{\mu} - \theta(\boldsymbol{\mu}^\top \boldsymbol{1} - 1),$$

where $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_K]^\top \geq 0$ and $\theta \geq 0$ are the lagrangian multipliers. The optimal solution μ^* satisfies the KKT condition:

$$\partial_{\boldsymbol{\mu}} \boldsymbol{L}(\boldsymbol{\mu}^*, \boldsymbol{\eta}, \theta) = \boldsymbol{v} + \beta \boldsymbol{\mu}^* - \boldsymbol{\eta} - \theta \boldsymbol{1} = 0 \tag{6}$$

$$\boldsymbol{\mu}^* \geq 0, \boldsymbol{\mu}^{*\top} \boldsymbol{1} - 1 = 0 \tag{7}$$

$$\boldsymbol{\eta} \geq 0 \tag{8}$$

$$\mu_k^* \eta_k = 0 \tag{9}$$

from (6), we can obtain:

$$\mu_k = \frac{\eta_k + \theta - v_k}{\beta},$$

it can be discussed separately in three cases according to (7)–(9):

(1) When $\theta - v_k > 0$, since $\eta_k \geq 0$, we get $\mu_k > 0$. From the condition $\mu_k^* \eta_k = 0$, it can be obtained that $\eta_k = 0$. Then, $\mu_k = \frac{\theta - v_k}{\beta}$.
(2) When $\theta - v_k = 0$, $\mu_k = \eta_k / \beta$, from the condition $\mu_k^* \eta_k = 0$, we infer that $\eta_k = 0$, $\mu_k = 0$.
(3) When $\theta - v_k < 0$, if $\mu_k > 0$, then $\eta_k > 0$, that is inconsistent whit $\mu_k^* \eta_k = 0$. Thus, we get $\mu_k \leq 0$. And because $\mu_k \geq 0$, we get $\mu_k = 0$. Therefore, with $\boldsymbol{v}$ increasing, we can find the positive integer $P = argmax_k\, (\theta - v_k > 0)$. The optimality conditions are summarized as follows:

$$\mu_k = \begin{cases} \frac{\theta - v_k}{\beta} & k \leq P \\ 0 & k > P. \end{cases}$$

From $\boldsymbol{\mu}^\top \boldsymbol{1} = 1$, i.e., $\sum_{k=1}^{P} \mu_k = 1$, we can get:

$$\theta = \frac{\sum_{k=1}^{P} v_k + \beta}{P}. \qquad \square$$

4.3. Time complexity analysis

The time complexity of MVCSC mainly depends on the computation of Algorithms 2 and 3. In Algorithm 2, the complexity of $\boldsymbol{F}$ is determined by the computation of $\boldsymbol{G}$, $\boldsymbol{A}$ and $\boldsymbol{Y(\tau)}$. To compute $\boldsymbol{G}$ and $\boldsymbol{A}$, it needs to take $\mathcal{O}(N^2 c + N^2 M)$ and $\mathcal{O}(N^2 c)$. The computation of $Y(\tau)$ can be $\mathcal{O}(c^2 N + c^3)$ according to [39]. So the computation complexity of $\boldsymbol{F}$ is $\mathcal{O}(T_\tau \times (N^2 c + N^2 M + c^2 N + c^3))$, where T_τ is the iteration number of searching an appropriate τ with BB steps [37]. $\boldsymbol{Z}$ and $\boldsymbol{\lambda}$ have the same computation cost of $\mathcal{O}(MNc)$. As $c \ll N$, $M \ll N$, Algorithm 2 takes $\mathcal{O}(T_2 \times (T_\tau \times (N^2 \times$

Algorithm 2: ADMM for estimating indicator matrix $\boldsymbol{F}$.

Input: graph Laplacian $\boldsymbol{L}_k(k = 1, 2, \ldots, K)$,constraint matrix $\boldsymbol{C}$, weighting coefficients $\vec{\mu}$, parameter ρ, γ.
Output: indicator matrix $\boldsymbol{f}$.

for $iter = 1$ *to* T_2 **do**
  Update $\boldsymbol{f}$ by using the Curvilinear Search Method with BB steps [37];
  Update $\boldsymbol{z}$ according to Eq.(3);
  Update $\boldsymbol{\lambda}$ according to Eq.(4);
end

Algorithm 3: Lagrange Multiplier for Auto-Weighting.

Input: sorted vector $\vec{v}$, parameter β.
Output: weight vector $\vec{\mu}$.

for $P \leftarrow 1$ *to* K **do**
  $\theta \leftarrow (\sum_{k=1}^{P} v_k + \beta)/P$;
  if $P \leftarrow argmax_k\,(\theta - v_k > 0)$ **then**
    break;
  end
end
$\mu_k \leftarrow (\theta + v_k)/\beta, k \le P$;
$\mu_k \leftarrow 0, k > p$.

$\max(c, M))))$, where T_2 is the number of iterations in Algorithms 2. In Algorithms 3, each iteration for the loop takes $\mathcal{O}(K)$ to get the maximal number P. It is obvious that the computational complexity of the entire process is $\mathcal{O}(K^2)$. As $K \ll N$, the overall complexity of MVCSC is $\mathcal{O}(T_1 \times T_2 \times (T_\tau \times (N^2 \times \max(c, M))))$, where T_1 is the number of iterations of Algorithm 1. In our experiments, T_τ and T_2 are usually less than 5, and T_1 is usually less than 10.

5. Experiment

In this section, we compare the performance of our multi-view constrained spectral clustering with several state-of-the-art methods on various multi-view datasets.

5.1. Data set description

Synthetic data consists of six views. The first two views are generated by three component Gaussian mixture model. The features are correlated. We sample 100 points for view1 and view2. The cluster means in view1 are $\mu_1^{(1)} = (1\ 1)$, $\mu_2^{(1)} = (3\ 4)$, $\mu_3^{(1)} = (1\ 3)$, and in view2 are $\mu_1^{(2)} = (1\ 2)$, $\mu_2^{(2)} = (2\ 4)$, $\mu_3^{(2)} = (3\ 2)$. The covariances of two views are given below. For view3 and view4, we add a small amount of random noise based on view1 and view2. The last two views are generated to be irrelevant by reordering view1 and view2.

$$\Sigma_1^{(1)} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}, \Sigma_2^{(1)} = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.6 \end{pmatrix}, \Sigma_3^{(1)} = \begin{pmatrix} 1.2 & 0.2 \\ 0.2 & 1 \end{pmatrix}$$

$$\Sigma_1^{(2)} = \begin{pmatrix} 1 & 0.2 \\ 0.2 & 1 \end{pmatrix}, \Sigma_2^{(2)} = \begin{pmatrix} 0.6 & 0.1 \\ 0.1 & 0.5 \end{pmatrix}, \Sigma_3^{(2)} = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 0.7 \end{pmatrix}$$

20Newsgroups[1] data is a synthetic dataset and collects approximately 20000 newsgroup documents. We choose 12 newsgroups from three categories Comp, Rec, Talk, and each category has four newsgroups. In each category, we generate 10 relevant views and 5 irrelevant views. For relevant views, we randomly sample 400 documents from 4 news groups (100 documents from each group). For irrelevant views, we randomly choose 5 views from 10 relevant views and reorder them.

Reuters [24] consists of documents that are written in five different languages and their translations (English, French, German, Spanish, and Italian). All the documents are categorized into 6 classes. We randomly sample 1200 documents (200 documents for each class). After generating five different languages graphs, we reorder them to generate five irrelevant views.

Caltech-101 data [40] consists of 101 categories of images. We choose the widely used 7 classes which contain totally 1474 images, i.e., faces (435), motorbikes (798), dollar_bill (52), garfield (34), snoopy (35), stop_sign (64), windsor_chair (56). We call it *Caltech-07*. Six features are provided, i.e., 48 dimension Gabor feature, 40 dimension wavelet moments, 254 dimension CENHIST feature, 1984 dimension HOG feature, 512 dimension GIST feature, 928 dimension LBP feature.

Handwritten[2] is a handwritten digits $(0-9)$ data from the UCI repository. It consists of 2000 samples and each class has 200 patterns. We use six features including 76 Fourier coefficients of the character shapes (FOU), 216 profile correlations (FAC), 64 Karhunen–Love coefficients (KAR), 240 pixel averages in 2×3 windows (Pix), 47 Zernike moments (ZER), 6 morphological features (MOR).

5.2. Experimental settings

We will evaluate our proposed MVCSC with several baseline approaches. These baseline methods can be divided into three categories: single view constrained clustering, multi-view unconstrained clustering and multi-view constrained clustering.

- Self-taught spectral framework (**SF**[3]) [33]: SF is a single-view constrained clustering algorithm based on spectral clustering. It effectively improves the utilization of pairwise constraints by incorporating self-teaching. Thus, SF is compared as the single view baseline. Since our datasets consist of multiple views, we choose the best clustering performance **SF_best** and its fused performance **SF_fusion**, where the fused Laplacian matrix is defined as $\boldsymbol{L} = 1/K \sum_{k=1}^{K} \boldsymbol{L}_k$.
- Co-regularization multi-view spectral clustering (**Co − reg**[4]) [21]: Co-reg is a classical and well-known multi-view spectral clustering which has a favour of the co-regularization idea. This method is selected as one of two comparative methods for multi-view unconstrained clustering.
- Auto-weighted multiple graph learning (**AMGL**[5]) [23]: AMGL is a multi-view approach via reformulating the standard spectral learning model, which automatically learns an optimal weight without introducing an additive parameter. AMGL will be compared as the second multi-view unconstrained clustering.
- Multi-view constraint propagation (**MMCP**) [11]: Inspired by label propagation, MMCP propagates the pairwise relationship via both vertical and horizontal propagation, two independent multi-graph based propagation methods. The propagated constraints are further used to refine similarities of instances. Finally, the prediction performance of MMCP is obtained by applying spectral clustering on the new similarity matrix. MMCP is introduced as the first multi-view constrained baseline.
- Multi-view matrix completion with side information (**MVMC**[6]) [12]: Considering the tight relation between similarities and

[1] http://qwone.com/%7Ejason/20Newsgroups/.

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

[3] https://github.com/gnaixgnaw/CSP.

[4] https://github.com/areslp/matlab/tree/master/code_coregspectral.

[5] http://www.escience.cn/people/fpnie/index.html.

[6] http://lamda.nju.edu.cn/zhaop/.

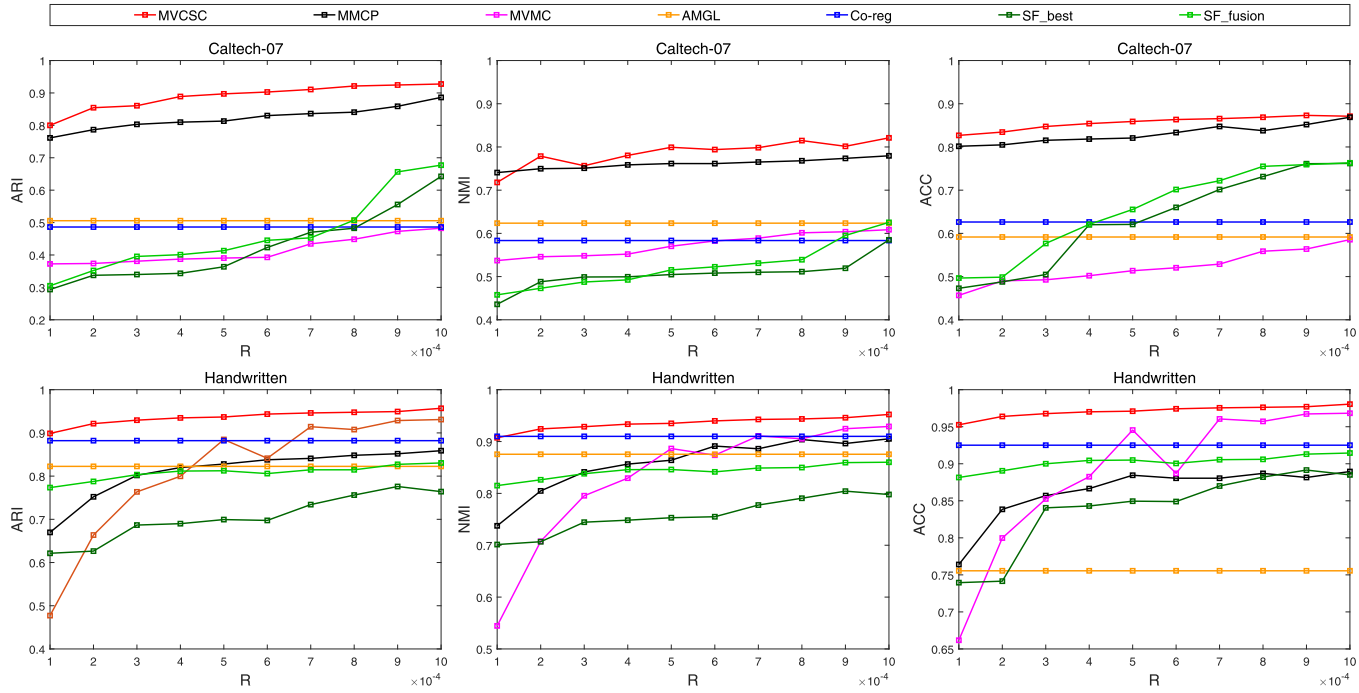


Fig. 2. Comparations of clustering performance between other state-of-the-art approaches and the proposed MVCSC on Caltech-07 and Handwritten datasets with respect to the averaged ARI, NMI, ACC (the higher, the better). R is used to measure the amount of pairwise constraints which varies 0.01% to 0.1%.

pairwise constraints, MVMC transfers constrained clustering problem to similarity matrix completion problem. The similarity matrix updates iteratively with the help of data form multiple modalities or representations. To obtain the clustering results, MVMC also performs the spectral clustering on the final similarity matrix. MVMC is chosen as the second comparative multi-view constrained clustering method.

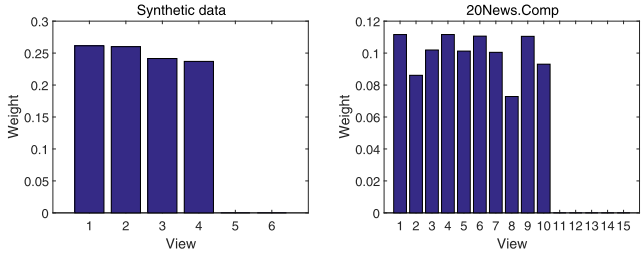
Pairwise Constraints: For the construction of the constraint matrix $\boldsymbol{C}$, we randomly select a pair of samples and repeat M times. If the pair instances have the same label, we form them as ML while as CL otherwise. R is used to measure the number of constraints, i.e., $R = M/N^2$. We vary R from $[0.01\%, 0.02\%, \ldots, 0.1\%]$.

Metrics: For the experiment results, we report three metrics: the Adjusted Rand Index (**ARI**) [41], the Normalized Mutual Information (**NMI**) and the Clustering Accuracy (**ACC**) [42]. We note that all the metrics lie in the interval [0,1], and the higher result demonstrates the better clustering performance. Each experiment randomly run 10 times and the averaged ARI, NMI, ACC are reported.

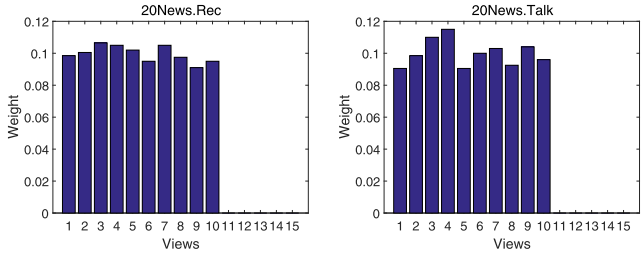
Settings: For Synthetic dataset, we compute the similarities using Gaussian kernel (width $\sigma = 1$). For 20Newsgroups dataset and Reuters dataset, the similarities are computed based on cosine similarity. For these three data sets, the corresponding 10 NN graphs are constructed. For Caltech-07 and Handwritten, we use the Gaussian kernel (width $\sigma = 1$) to compute the affinity matrix and construct 5 NN graphs.

5.3. Results

Due to the space limitation, we only present the Figs. 2 and 3 and Tables 2–4 in experiments section to demonstrate the proposed MVCSC approach. The Tables 2–4 respectively summarize the comparison with $R = [0.01\%, 0.04\%, 0.07\%, 0.1\%]$ for Synthetic dataset, 20Newsgroups, and Reuters. Fig. 2 summarizes the performance with respect to ARI, NMI and ACC on Caltech-07 and Handwritten datasets when R increases from 0.01% to 0.1%. The



(a) Synthetic data, β=9 (b) 20News.Comp, β=9



(c) 20News.Rec, β=9 (d) 20News.Talk, β=9

Fig. 3. The average weights over 10 run on Synthetic and 20Newsgroups datasets with $R = 0.01\%$.

Table 1
Statistics of evaluated datasets.

Dataset	size	view	cluster
Synthetic data	300	6	3
20News.Comp	400	15	4
20News.Rec	400	15	4
20News.Talk	400	15	4
Reuters	1200	10	6
Caltech-07	1474	6	7
Handwritten	2000	6	10

Table 2
The averaged ARI results on Synthetic data, 20Newsgroups and Reuters for different baselines and MVCSC.

Datasets	Ration	SF_best	SF_fusion	Co-reg	AMGL	MVMC	MMCP	MVCSC
Synthetic data	0.01%	0.2314	0.7540	0.7642	0.8002	0.3462	0.7726	**0.8313**
	0.04%	0.2663	0.7624	0.7642	0.8002	0.4589	0.7821	**0.8556**
	0.07%	0.2929	0.7633	0.7642	0.8002	0.6915	0.7993	**0.8634**
	0.1%	0.3311	0.7810	0.7642	0.8002	0.7106	0.8096	**0.8737**
20News.Comp	0.01%	0.0873	0.6812	0.7885	0.8801	0.0176	0.7460	**0.9021**
	0.04%	0.0949	0.6882	0.7885	0.8801	0.0238	0.7879	**0.9215**
	0.07%	0.1158	0.6976	0.7885	0.8801	0.0466	0.8112	**0.9411**
	0.1%	0.1157	0.7001	0.7885	0.8801	0.0531	0.8316	**0.9481**
20News.Rec	0.01%	0.0579	0.4678	0.8664	0.9320	0.0112	0.8134	**0.9578**
	0.04%	0.0927	0.4874	0.8664	0.9320	0.0289	0.8360	**0.9741**
	0.07%	0.0985	0.4885	0.8664	0.9320	0.0646	0.8541	**0.9794**
	0.1%	0.1017	0.5717	0.8664	0.9320	0.0841	0.8734	**0.9801**
20News.Talk	0.01%	0.1193	0.4961	0.8702	0.8519	0.0113	0.8207	**0.8893**
	0.04%	0.1284	0.5597	0.8702	0.8519	0.0233	0.8586	**0.8976**
	0.07%	0.1753	0.5788	0.8702	0.8519	0.0496	0.8706	**0.9010**
	0.1%	0.1966	0.6402	0.8702	0.8519	0.0525	0.8958	**0.9201**
Reuters	0.01%	0.3554	0.6542	0.7770	0.8708	0.1458	0.6217	**0.8981**
	0.04%	0.3639	0.6841	0.7770	0.8708	0.2275	0.6778	**0.9205**
	0.07%	0.3824	0.7015	0.7770	0.8708	0.2935	0.6856	**0.9376**
	0.1%	0.3888	0.7224	0.7770	0.8708	0.4322	0.7055	**0.9445**

Table 3
The averaged NMI results on Synthetic data, 20Newsgroups and Reuters for different baselines and MVCSC.

Datasets	Ration	SF_best	SF_fusion	Co-reg	AMGL	MVMC	MMCP	MVCSC
Synthetic data	0.01%	0.3748	0.6980	0.6935	0.7362	0.3486	0.7157	**0.7710**
	0.04%	0.3941	0.7087	0.6935	0.7362	0.4422	0.7231	**0.7970**
	0.07%	0.4059	0.7088	0.6935	0.7362	0.6365	0.7410	**0.8055**
	0.1%	0.4067	0.7249	0.6935	0.7362	0.6554	0.7497	**0.8167**
20News.Comp	0.01%	0.1841	0.7995	0.7899	0.8709	0.0262	0.7298	**0.8868**
	0.04%	0.1919	0.8017	0.7899	0.8709	0.0404	0.7516	**0.9051**
	0.07%	0.2202	0.8172	0.7899	0.8709	0.0618	0.7916	**0.9228**
	0.1%	0.2216	0.8278	0.7899	0.8709	0.7690	0.8050	**0.9292**
20News.Rec	0.01%	0.2367	0.6219	0.8531	0.9149	0.0297	0.7979	**0.9250**
	0.04%	0.2415	0.6636	0.8531	0.9149	0.0554	0.8103	**0.9351**
	0.07%	0.2487	0.6755	0.8531	0.9149	0.0724	0.8321	**0.9425**
	0.1%	0.2732	0.6871	0.8531	0.9149	0.1048	0.8582	**0.9501**
20News.Talk	0.01%	0.3008	0.5481	0.8329	0.8454	0.0261	0.7815	**0.8645**
	0.04%	0.3216	0.6072	0.8329	0.8454	0.0459	0.8228	**0.8763**
	0.07%	0.3262	0.6529	0.8329	0.8454	0.0617	0.8365	**0.8779**
	0.1%	0.3508	0.6488	0.8329	0.8454	0.0755	0.8661	**0.9048**
Reuters	0.01%	0.1952	0.4831	0.6778	0.8710	0.1113	0.4085	**0.8947**
	0.04%	0.2007	0.5387	0.6778	0.8710	0.1589	0.4847	**0.9126**
	0.07%	0.2145	0.5786	0.6778	0.8710	0.2057	0.5528	**0.9284**
	0.1%	0.2500	0.6015	0.6778	0.8710	0.2794	0.5942	**0.9327**

Table 4
The averaged ACC results on Synthetic data, 20Newsgroups and Reuters for different baselines and MVCSC.

Datasets	Ration	SF_best	SF_fusion	Co-reg	AMGL	MVMC	MMCP	MVCSC
Synthetic data	0.01%	0.5733	0.9133	0.9167	0.9300	0.6066	0.9200	**0.9413**
	0.04%	0.5900	0.9166	0.9167	0.9300	0.7750	0.9233	**0.9501**
	0.07%	0.6267	0.9167	0.9167	0.9300	0.8850	0.9301	**0.9533**
	0.1%	0.7067	0.9233	0.9167	0.9300	0.8933	0.9333	**0.9567**
20News.Comp	0.01%	0.4825	0.7402	0.9100	0.9525	0.3113	0.8975	**0.9620**
	0.04%	0.4901	0.7450	0.9100	0.9525	0.3475	0.9151	**0.9695**
	0.07%	0.5201	0.7551	0.9100	0.9525	0.3688	0.9250	**0.9775**
	0.1%	0.5225	0.7575	0.9100	0.9525	0.3963	0.9325	**0.9802**
20News.Rec	0.01%	0.4625	0.5125	0.9475	0.9728	0.3288	0.9251	**0.9840**
	0.04%	0.4675	0.5200	0.9475	0.9728	0.3401	0.9350	**0.9903**
	0.07%	0.4725	0.5202	0.9475	0.9728	0.3550	0.9425	**0.9922**
	0.1%	0.4751	0.6800	0.9475	0.9728	0.3888	0.9500	**0.9925**
20News.Talk	0.01%	0.4251	0.6825	0.9500	0.9521	0.3200	0.9300	**0.9580**
	0.04%	0.4500	0.6925	0.9500	0.9521	0.3388	0.9451	**0.9613**
	0.07%	0.5075	0.7001	0.9500	0.9521	0.3451	0.9500	**0.9625**
	0.1%	0.5351	0.8352	0.9500	0.9521	0.3725	0.9610	**0.9701**
Reuters	0.01%	0.4800	0.6608	0.7792	0.9450	0.3413	0.6375	**0.9571**
	0.04%	0.4851	0.7283	0.7792	0.9450	0.4092	0.6817	**0.9664**
	0.07%	0.4867	0.7325	0.7792	0.9450	0.4401	0.7275	**0.9736**
	0.1%	0.5108	0.7583	0.7792	0.9450	0.5021	0.7533	**0.9767**

Fig. 3 shows the view weights for Synthetic datasets and 20Newsgroups dataset.

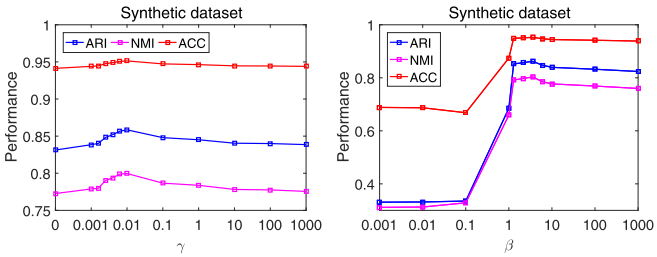
From Synthetic data results in Tables 2–4, we can see the proposed MVCSC gives the best performance in all criteria. On the one hand, when R grows, the performance of MVCSC gradually improves. In addition, MVCSC gives higher accuracy on all three metrics than multi-view clustering methods Co-reg and AMGL. Based on these, we can infer that MVCSC can efficiently handle with pairwise constraints. Besides it, neither MVMC nor MMCP outperforms MVCSC. The first reason is that MVMC and MMCP have a lower utilization of pairwise constraints. Another reason is that they can't differ the importance of different views. In stark contrast, MVCSC sets the weights of unrelated view5 and view6 to 0 in the Fig. 3(a). Meanwhile, the weights of noisy view3 and view4 are lower than those of view1 and view2. On the other hand, the single view constrained clustering SF_best behaves worst due to that it can only exploit single view data. SF_fusion ignores the correlation between multiple views and also can't perform well compared to MVCSC.

From the document clustering results on 20Newsgroups and Reuters datasets in Tables 2–4, we can see MVCSC shows a surprisingly better performance than all baselines on all criteria. With respect to multi-view semi-supervised methods, as the number of pairwise constraints grows, the performance of MVCSC improves more significantly than those of MMCP and MVMC. It demonstrates that the MVCSC can incorporate the pairwise constraints into multi-view clustering more effectively and efficiently. Besides it, MVCSC eliminates the last 5 irrelevant views with zero weight, which is shown in Fig. 3(b)–(d). In contrast, MMCP and MVMC can't differ the importance of different views and perform unsatisfactorily. In addition, it's crucial to mention that the number of sampled pairwise constraints only accounts for 0.01% to 0.1%. It is encouraging that, with such limited constraint information, MVCSC still yields a satisfying behavior. SF_best and SF_fusion, although their performance also benefit from the increasing constraints, perform poorly due to that they only make use of single view data and simply fuse multiple graphs respectively. MVCSC reasonably integrates the multi-view data and outperforms them. As compared with the multi-view unconstrained method Co-reg and AMGL, the improvement which MVCSC achieves is significant by incorporating the pairwise constraints. It demonstrates that pairwise constraints is of great value for multi-view clustering.

From the image clustering on datasets Caltech-07 and Handwritten in Fig. 2, we can see that the proposed MVCSC outperforms all the baselines by a significant margin on almost all cases. We also find an obvious improvement of MVCSC along with the growth of R. Furthermore, MVCSC is able to demonstrate a much better behavior when pairwise constraints extremely scarce. For example, when the number of pairwise constraints only accounts for 0.01%, MVCSC gives a ARI value of 0.8988 while MVMC and MMCP only give 0.4774 and 0.6698 respectively. It proves that MVCSC gives more guidance for partition than MVMC and MMCP. After incorporating pairwise constraints, MVCSC achieves a large improvement compared with AMGL and Co-reg, the multi-view unconstrained clustering methods. This phenomenon implicates that pairwise constraints have large leverage in multi-view clustering. Similar to the situation on the document clustering, SF_best and SF_fusion behave poorly although their performance benefits from the increasing number of constraints. MVCSC learns the underlying consistent structure of multi-view data and outperforms them.

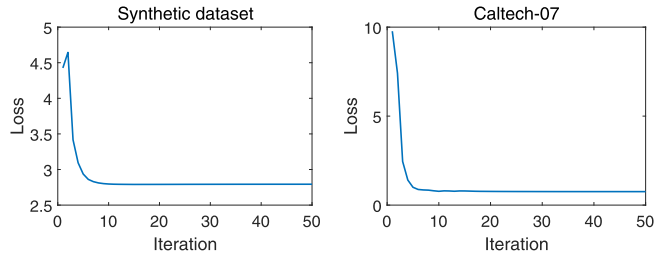
5.4. Parameter sensitivity

In this part, we further study how our approach performs when using different settings of parameters. We tune the trade-off parameters γ, β in the range of 0.001, 0.01, 0.1, 1, 10, 100, 1000 on Synthetic dataset. For studying the effect without constraints, we further consider the case when $\gamma = 0$.



(a) Parameters γ ($\beta = 9$) (b) Parameters β ($\gamma = 0.1$)

Fig. 4. Parameters sensitivity of γ and β on Synthetic dataset with $R = 0.1\%$.



(a) Synthetic data, $\gamma = 0.01$, $\beta = 9$ (b) Caltech-07,$\gamma = 0.03$, $\beta = 1$

Fig. 5. The convergence analysis from the perspectives of the loss on Synthetic and Caltech-07 datasets with $R = 0.01\%$.

Effects of γ. As shown in Fig. 4 (a), the performance of MVCSC gets better when γ increases from 0.001 to 0.01, which reflects that more constraints are satisfied, thereby improving the quality of the partition. When γ keeps going up, the prediction error will not be well controlled, which makes our approach to produce poor prediction results. When γ is 0 (without constraints), MVCSC can still effectively capture the underlying structure of the data and achieve satisfactory performance.

Effects of β. As an illustration in Fig. 4(b), we can see that the optimal locates within [2,5]. When β is too small, weights will be assigned to only a few domains such that not enough relevant views will be integrated. When β is larger than 5, it is easy to see a decreasing trend instead. In these cases, weights will be averagely assigned to all views such that irrelevant views will be involved.

5.5. Convergence analysis

The original problem (1) is not a joint convex problem with respect to $\boldsymbol{F}$ and $\boldsymbol{\mu}$. Thus, it can't be guaranteed to obtain a global optimum. For the first subproblem (2), the indicator matrix $\boldsymbol{F}$ would get a global solution with either full or high probability according to [36]. And the solution of $\boldsymbol{Z}$ is in closed-form. Obviously, the other subproblem (5) is convex and has a closed-form solution. Thus, our method will converge to a local optimum in most cases. In addition, we empirically test the convergence of the proposed method on Synthetic and Caltech-07 datasets from two aspects: the loss and the relative difference of variables. As illustrated in Fig. 5, our algorithm is able to achieve a rapid convergence within only a few iterations (less than 10) from the perspectives of the loss. From Fig. 6, we can see the relative difference between two successive iterates is decreasing to a small value of approximately zero, which further illustrates the convergence of the algorithm. Similar results can be observed on other datasets.

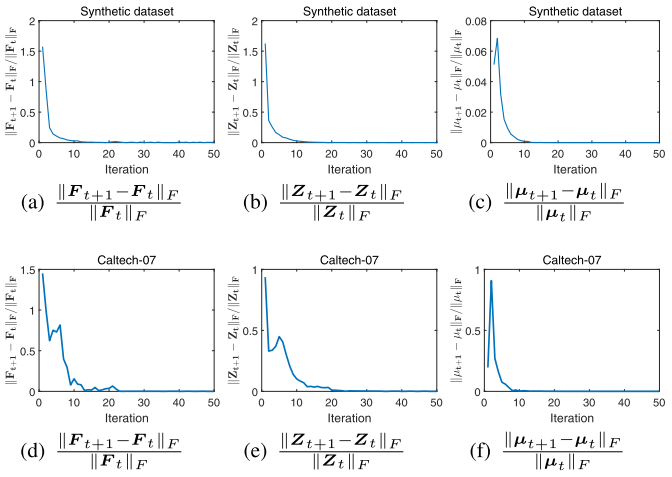


Fig. 6. The convergence analysis from the perspectives of variables on Synthetic and Caltech-07 datasets with $R = 0.01\%$.

6. Conclusion

In this paper, we present a novel multi-view constrained clustering algorithm MVCSC, which is the first study to incorporate pairwise constraints to multi-view clustering based on spectral clustering framework. To efficiently handle with pairwise constraints, MVCSC constructs consistent ML and CL for multi-view clustering by imposing them as a set of linear constraints on the unified indicator matrix. MVCSC has the following characteristics: (1) based on spectral clustering framework, it is applicable to multi-class partition of arbitrary data distribution; (2) it can simultaneously and efficiently handle with both ML and CL for multi-view clustering; (3) it can automatically learn the weights for different views and eliminate the views contains much noisy or irrelevant information. MVCSC has been tested in constrained clustering tasks on various multi-view datasets and outperforms the state-of-the-art approaches.

Declarations of interest

None.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61722214, 11801595, U1811462), the Program for Guangdong Introducing Innovative and Entrepren-eurial Teams (2016ZT06D211) and the Pearl River S&T Nova Program of Guangzhou (201710010046).

References

[1] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., Constrained k-means clustering with background knowledge, in: Proceedings of the ICML, 1, 2001, pp. 577–584, doi:10.1109/TPAMI.2002.1017616.
[2] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning a Mahalanobis metric from equivalence constraints, J. Mach. Learn. Res. 6 (Jun) (2005) 937–965.
[3] M. Ganji, J. Bailey, P.J. Stuckey, A declarative approach to constrained community detection, in: Proceedings of the International Conference on Principles and Practice of Constraint Programming, Springer, 2017, pp. 477–494, doi:10.1007/978-3-642-40994-3_27.
[4] J. Tang, X. Hu, H. Liu, Is distrust the negation of trust? The value of distrust in social media, in: Proceedings of the Twenty-Fifth ACM Conference on Hypertext and Social Media, ACM, 2014, pp. 148–157, doi:10.1145/2631775.2631793.
[5] R. Yan, J. Zhang, J. Yang, A.G. Hauptmann, A discriminative learning framework with pairwise constraints for video object classification, IEEE Trans. Pattern Anal. Mach. Intell. 28 (4) (2006) 578–593, doi:10.1109/TPAMI.2006.65.
[6] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM, 1998, pp. 92–100, doi:10.1145/279943.279962.
[7] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Proceedings of the Twenty-First International Conference on Machine Learning, ACM, 2004, p. 11, doi:10.1145/1015330.1015360.
[8] D. Pelleg, D. Baras, K-means with large and noisy constraint sets, in: Proceedings of the European Conference on Machine Learning, Springer, 2007, pp. 674–682, doi:10.1007/978-3-540-74958-5_67.
[9] X. Wang, I. Davidson, Flexible constrained spectral clustering, in: Proceedings of the Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 563–572, doi:10.1145/1835804.1835877.
[10] E. Eaton, M. Desjardins, S. Jacob, Multi-view clustering with constraint propagation for learning with an incomplete mapping between views, in: Proceedings of the Nineteenth ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 389–398, doi:10.1145/1871437.1871489.
[11] Z. Fu, H.H. Ip, H. Lu, Z. Lu, Multi-modal constraint propagation for heterogeneous image clustering, in: Proceedings of the Nineteenth ACM International Conference on Multimedia, ACM, 2011, pp. 143–152, doi:10.1145/2072298.2072318.
[12] P. Zhao, Y. Jiang, Z.-H. Zhou, Multi-view matrix completion for clustering with side information, in: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2017, pp. 403–415, doi:10.1007/978-3-319-57529-2_32.
[13] Z. Kang, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, Knowl. Based Syst. 163 (2019) 510–517.
[14] Z. Kang, H. Pan, S.C.H. Hoi, Z. Xu, Robust graph learning from noisy data, CoRR (2018) abs/1812.06673.
[15] Z. Kang, C. Peng, Q. Cheng, Kernel-driven similarity learning, Neurocomputing 267 (2017) 210–219.
[16] C. Peng, Z. Kang, Q. Cheng, Subspace clustering via variance regularized ridge regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2931–2940.
[17] C. Peng, Z. Kang, S. Cai, Q. Cheng, Integrate and conquer: double-sided two-dimensional k-means via integrating of projection and manifold construction, ACM Trans. Intell. Syst. Technol. (TIST) 9 (5) (2018) 57.
[18] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416, doi:10.1007/s11222-007-9033-z.
[19] L. Hagen, A.B. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 11 (9) (1992) 1074–1085, doi:10.1109/43.159993.
[20] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905, doi:10.1109/34.868688.
[21] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 2011, pp. 1413–1421. doi:10.1.1.229.2081.
[22] A. Kumar, H.D. III, A co-training approach for multi-view spectral clustering, in: Proceedings of the Twenty-Eighth International Conference on International Conference on Machine Learning, in: ICML'11, 2011, pp. 393–400.
[23] F. Nie, J. Li, X. Li, et al., Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification., in: Proceedings of the IJCAI, 2016, pp. 1881–1887.
[24] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph., in: Proceedings of the AAAI, 2015, pp. 2750–2756.
[25] X. Cai, F. Nie, H. Huang, F. Kamangar, Heterogeneous image feature integration via multi-modal spectral clustering, in: Proceedings of the CVPR, 2011, pp. 1977–1984, doi:10.1109/CVPR.2011.5995740.
[26] Q. Yin, S. Wu, R. He, L. Wang, Multi-view clustering via pairwise sparse subspace representation, Neurocomputing 156 (2015) 12–21, doi:10.1016/j.neucom.2015.01.017.
[27] J. Liu, C. Wang, J. Gao, J. Han, Multi-view clustering via joint nonnegative matrix factorization, in: Proceedings of the SIAM International Conference on Data Mining, SIAM, 2013, pp. 252–260, doi:10.1137/1.9781611972832.28.
[28] D. Guo, J. Zhang, X. Liu, Y. Cui, C. Zhao, Multiple kernel learning based multi-view spectral clustering, in: Proceedings of the Twenty-Second International Conference on Pattern Recognition, IEEE, 2014, pp. 3774–3779, doi:10.1109/ICPR.2014.648.
[29] G. Chao, S. Sun, J. Bi, A survey on multi-view clustering, CoRR abs/1712.06246 (2017) arXiv:1712.06246.
[30] F. Wang, T. Li, C. Zhang, Semi-supervised clustering via matrix factorization, in: Proceedings of the SIAM International Conference on Data Mining, SIAM, 2008, pp. 1–12, doi:10.1137/1.9781611972788.1.
[31] S.D. Kamvar, D. Klein, C.D. Manning, Spectral learning, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, in: IJCAI'03, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, pp. 561–566.
[32] Z. Lu, H.H. Ip, Constrained spectral clustering via exhaustive and efficient constraint propagation, in: Proceedings of the European Conference on Computer Vision, Springer, 2010, pp. 1–14, doi:10.1007/978-3-642-15567-3_1.
[33] X. Wang, J. Wang, B. Qian, F. Wang, I. Davidson, Self-taught spectral clustering via constraint augmentation, in: Proceedings of the SIAM International Conference on Data Mining, SIAM, 2014, pp. 416–424, doi:10.1137/1.9781611973440.48.
[34] J. Kawale, D. Boley, Constrained spectral clustering using l1 regularization, in: Proceedings of the SIAM International Conference on Data Mining, SIAM, 2013, pp. 103–111, doi:10.1137/1.9781611972832.12.
[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., in: Distributed Optimization and Statistical Learning Via the Alternating Direction Method of

Multipliers, 3, Foundations and Trends® in Machine learning, 2011, pp. 1–122, doi:10.1561/2200000016.
[36] Z. Wen, W. Yin, A feasible method for optimization with orthogonality constraints, Math. Program. 142 (1–2) (2013) 397–434, doi:10.1007/s10107-012-0584-1.
[37] J. Barzilai, J.M. Borwein, Two-point step size gradient methods, IMA J. Numer. Anal. 8 (1) (1988) 141–148, doi:10.1093/imanum/8.1.141.
[38] S.J. Wright, R.D. Nowak, M.A. Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process. 57 (7) (2009) 2479–2493, doi:10.1109/TSP.2009.2016892.
[39] Y. Pan, T. Yao, T. Mei, H. Li, C.-W. Ngo, Y. Rui, Click-through-based cross-view learning for image search, in: Proceedings of the Thirty-Seventh International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2014, pp. 717–726, doi:10.1145/2600428.2609568.
[40] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, Comput. Vis. Image Underst. 106 (1) (2007) 59–70, doi:10.1109/CVPR.2004.383.
[41] S. Wagner, D. Wagner, Comparing Clusterings: An Overview, Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.
[42] F. Nie, D. Xu, I.W. Tsang, C. Zhang, Spectral embedded clustering., in: Proceedings of the IJCAI, 2009, pp. 1181–1186.

Chuan Chen received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2016. He is currently an Associate Research Fellow with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include machine learning, numerical linear algebra, and numerical optimization

Hui Qian received the B.S. degree from the Northeastern University at Qinhuangdao, Qinhuangdao, China, in 2017. She is currently pursuing the M.S. degree with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. Her research interests include multi-view learning, data fusion, network representation learning.

Wuhui Chen is an associate professor in Sun Yat-sen University, China. He received his bachelor's degree from Northeast University, China, in 2008. He received his master's and Ph.D. degrees from University of Aizu, Japan, in 2011 and 2014, respectively. From 2014 to 2016, he was a JSPS research fellow in Japan. From 2016 to 2017, he was a researcher in University of Aizu, Japan. His research interests include Edge/Cloud Computing, CloudRobotics, and Blockchain.

Zibin Zheng is a professor at School of Data and Computer Science, Sun Yat-sen University. He received Outstanding Ph.D. Thesis Award of the Chinese University of Hong Kong at 2012, ACM SIGSOFT Distinguished Paper Award at ICSE2010, Best Student Paper Award at ICWS2010, and IBM Ph.D. Fellowship Award at 2010. He served as PC member of IEEE CLOUD, ICWS, SCC, ICSOC, SOSE, etc. His research interests include service computing and cloud computing.

Hong Zhu received the Ph.D. degree in optimization from the Department of Mathematics, Hong Kong Baptist University, Hong Kong, in 2016. She is currently an Associate Professor with the Faculty of Science, Jiangsu University, Zhenjiang, China. Her current research interests include matrix optimization, data mining and image processing.