PLOS ONE

RESEARCH ARTICLE

Spectral clustering with distinction and consensus learning on multiple views data

Peng Zhou[1], **Fan Ye**[1]*, **Liang Du**[2]

1 School of Computer Science Technology, Anhui University, Hefei, China, **2** School of Computer and Information Technology, Shanxi University, Taiyuan, China

* yfan@ahu.edu.cn

Citation: Zhou P, Ye F, Du L (2018) Spectral clustering with distinction and consensus learning on multiple views data. PLoS ONE 13(12): e0208494. https://doi.org/10.1371/journal.pone.0208494

Editor: Ivan Olier, Liverpool John Moores University, UNITED KINGDOM

Received: April 17, 2018

Accepted: November 19, 2018

Published: December 6, 2018

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work is supported in part by the Key Natural Science Project of Anhui Provincial Education Department KJ2018A0010, and National Natural Science Foundation of China grant 61502289.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Since multi-view data are available in many real-world clustering problems, multi-view clustering has received considerable attention in recent years. Most existing multi-view clustering methods learn consensus clustering results but do not make full use of the distinct knowledge in each view so that they cannot well guarantee the complementarity across different views. In this paper, we propose a Distinction based Consensus Spectral Clustering (DCSC), which not only learns a consensus result of clustering, but also explicitly captures the distinct variance of each view. It is by using the distinct variance of each view that DCSC can learn a clearer consensus clustering result. In order to optimize the introduced optimization problem effectively, we develop a block coordinate descent algorithm which is theoretically guaranteed to converge. Experimental results on real-world data sets demonstrate the effectiveness of our method.

Introduction

Many real-world data sets are represented in multiple views. For example, images on the web may have two views: visual information and textual tags; multi-lingual data sets have multiple representations in different languages. Different views can often provide complementary information which is very helpful to improve the performance of learning. Multi-view clustering aims to do clustering on such multi-view data by using the information from all views.

Over the past years, many multi-view clustering methods are proposed. Roughly speaking, depending on the goal of the clustering learning, they can be categorized into two closely related but different families. In the first family, they aim to learn a common or consensus clustering result from multiple views. These methods [1–6] usually extend single view clustering methods such as spectral clustering or nonnegative matrix factorization (NMF) to deal with multi-view data. For example, Cai et al. extended k-means for multi-view data, leading to a multi-view k-means clustering method [1]; Liu et al. extended the NMF method to multi-view clustering [2]; Kumar et al. presented a co-training approach for multi-view spectral clustering by bootstrapping the clusterings of different views [3]. Kumar et al. also proposed two coregularization based approaches for multi-view spectral clustering by enforcing the clustering hypotheses on different views to agree with each other [4]. Xia et al. proposed a robust

multi-view spectral clustering method by building a Markov chain based on a low rank and sparse decomposition method [5]. Nie et al. presented a parameter-free multi-view spectral clustering method, which learned an optimal weight for each view automatically without introducing an additive parameter and extended it to semi-supervised classification [6, 7]. Zhang et al. learned a uniform projection to map the multiple views to a consensus embedding space [8]. Tang et al. extended the NMF to unsupervised multi-view feature selection by making use of the consensus information of all views [9]. All these methods learn the consensus clustering from each view, while ignoring the distinct information that only exits in one view while does not exist in other views. Therefore, these methods could not guarantee the complementarity across different views [10].

In the second family, instead of learning a consensus clustering result, they tend to learn distinct clustering result on each view. These methods [10–12] make use of complementary information to improve the performance of clustering on each view. For example, Günnemann et al. presented a multi-view clustering method based on subspace learning, which provides multiple generalizations of the data by modeling individual mixture models, each representing a distinct view [11]; Cao et al. also presented a subspace learning based multi-view clustering method which learns better clustering result on each view [10]. Different from the first family, they learn the clustering results on each view instead of the consensus clustering results.

In this paper, we focus on the first family, which is to learn a consensus clustering result. It is well known that multi-view learning follows the complementary principle, that each view of data may contain some knowledge that other views do not have [13, 14], and some theoretical and experimental results [11, 15, 16] have already demonstrated it. However, most existing consensus clustering methods do not make full use of the distinguishing information, and thus could not well guarantee the complementarity across different views [10]. To address this issue, we propose a Distinction based Consensus Spectral Clustering (DCSC), borrows the main idea of the second family that the distinguishing information may be helpful, to learn a better consensus result.

Since spectral clustering is a widely used clustering method in both single view and multiple view settings [4, 5, 17], we adopt it as the basic clustering model for our method. The essential step of spectral clustering is to learn a spectral embedding. In our method, the underlying spectral embedding of each view consists of two parts; one is the consensus embedding of all views and the other is the sparse variance of each view. In order to distinguish between variances, we apply Hilbert Schmidt Independence Criterion (HSIC) to measure and control the diversities of all views. Therefore, we learn a cleaner consensus embedding of all views by explicitly captures the distinct variance of each view. Note that, Liu et al. [18] also considered the consistency and complementarity, i.e., they decomposed the latent factor into two parts: the common part and the specific part. However, they did not impose any constraints (like HSIC in our method) on the specific parts to control the diversity, so that it is possible that the learned specific parts may be similar in their methods.

We develop a block coordinate descent algorithm for effectively learning the consensus embedding and distinguishing variances, which is theoretically guaranteed to converge. The experiments on benchmark data sets show that our method outperforms the closely related algorithms, which indicates the importance of using distinct information in multi-view clustering.

To sum up, we highlight the main contribution of this paper here: we propose a new multi-view spectral clustering method, which uses the HSIC to explicitly capture the distinction information of all views and can obtain a clearer and more accurate consensus result; and

then, we provide a block coordinate descent algorithm to solve it effectively and the experimental results demonstrate that our algorithm outperforms other state-of-the-art methods.

The remainder of this paper is organized as follows: Section 2 introduces some preliminaries. Section 3 presents our Distinction based Consensus Spectral Clustering method. Section 4 shows the experimental results. Section 5 concludes this paper.

Preliminaries

Spectral clustering

Spectral clustering [19] is a widely used clustering method. Given a data set which contains data points $\{x_1, \ldots, x_n\}$, it firstly defines similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ where $S_{ij} \geq 0$ denotes the similarity of x_i and x_j. Then it constructs a Laplacian matrix $\mathbf{L}$ by $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I}$ is an identity matrix and $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the (i, i)-th element $d_{ii} = \sum_{j=1}^{n} S_{ij}$. Spectral clustering aims to learn a spectral embedding $\mathbf{Y} \in \mathbb{R}^{n\times c}$ (c is the dimension of embedding space and is often set to the number of clusters) by optimizing the following objective function

$$\begin{aligned} \min_{\mathbf{Y}} \quad & tr(\mathbf{Y}^T\mathbf{L}\mathbf{Y}), \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}. \end{aligned} \tag{1}$$

When getting spectral embedding $\mathbf{Y}$, it applies k-means or spectral rotation [20] to discretize $\mathbf{Y}$ to obtain the final clustering result.

Hilbert schmidt independence criterion

Let $k_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_2 : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be two positive-definite reproducing kernels that correspond to RKHSs (Reproducing Kernel Hilbert Space) [21] $\mathcal{H}_{k_1}$ and $\mathcal{H}_{k_2}$ respectively with inner-products $k_1(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ and $k_2(y_i, y_j) = \langle \psi(y_i), \psi(y_j) \rangle$, where $\phi : \mathcal{X} \to \mathcal{H}_{k_1}$ and $\psi : \mathcal{Y} \to \mathcal{H}_{k_2}$ are two maps and $\langle \cdot, \cdot \rangle$ denotes the inner product. Then the cross covariance is defined as:

$$\mathbf{C}_{\mathbf{xy}} = E_{xy}[(\phi(x) - \mu_x) \otimes (\psi(y) - \mu_y)] \tag{2}$$

where $\otimes$ is the outer product, $\mu_x = E(\phi(x))$ and $\mu_y = E(\psi(y))$, and $E(\cdot)$ denotes the expectation. Then Hilbert Schmidt Independence Criterion (HSIC) is defined as:

Definition 1. [22] *Given two separable RKHSs and a joint distribution p_{xy}, we define the HSIC as the Hilbert-Schmidt norm of the associated cross-covariance operator* $\mathbf{C}_{\mathbf{xy}}$:

$$HSIC(p_{xy}) \coloneqq \| \mathbf{C}_{\mathbf{xy}} \|^2_{HS}, \tag{3}$$

where $\|\cdot\|_{HS}$ denotes the Hilbert-Schmidt norm of a matrix.

From the definition, we can find that HSIC can be used to measure the independence of two variables, i.e., the less HSIC is, the more independent the two variables are.

Since at most time the joint distribution p_{xy} is unknown, the empirical version of HSIC is often used. Let $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ be two data sets which contain $\{z_1^{(1)}, \ldots, z_n^{(1)}\}$ and $\{z_1^{(2)}, \ldots, z_n^{(2)}\}$ as their data respectively. Then the empirical version of HSIC is defined as:

$$HSIC(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}) = \frac{1}{(n-1)^2} tr(\mathbf{K}^{(1)}\mathbf{H}\mathbf{K}^{(2)}\mathbf{H}), \tag{4}$$

where $\mathbf{K}^{(1)}$ and $\mathbf{K}^{(2)}$ are the Gram matrices with the (i, j)-th element $K_{ij}^{(1)} = k_1(z_i^{(1)}, z_j^{(1)})$ and

$K_{ij}^{(2)} = k_2(z_i^{(2)}, z_j^{(2)})$, $\mathbf{H}$ is a centering matrix defined by $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is an all-ones vector. More details of HSIC can be found in [22].

Distinction based consensus spectral clustering

In this section, we present the framework of DCSC, and then introduce how to solve the introduced optimization problem.

Formulation

Given a multi-view data set $\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}\}$ which contains n instances, where m is the number of views, we can construct m Laplacian matrices $\mathbf{L}^{(1)}, \ldots, \mathbf{L}^{(m)}$ as [19] did. Then for each view we learn the spectral embedding by solving Eq (1). However, in multi-view data, each view contains both common information and distinguishing knowledge. To capture them, we decompose the spectral embedding of the i-th view into two parts: the consensus embedding $\mathbf{Y} \in \mathbb{R}^{n\times c}$ and the distinct variance $\mathbf{V}^{(i)} \in \mathbb{R}^{n\times c}$. Then the objective function of Eq (1) can be rewritten as

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{V}^{(i)}} \quad & \sum_{i=1}^{m} tr((\mathbf{Y}+\mathbf{V}^{(i)})^T\mathbf{L}^{(i)}(\mathbf{Y}+\mathbf{V}^{(i)})), \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}. \end{aligned} \tag{5}$$

where the orthogonal constraint $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$ is to avoid the trivial solution.

Since $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(m)}$ denote the distinct variance of each view, they should be far apart from each other. Here we use the aforementioned HSIC to measure the difference between $\mathbf{V}^{(i)}$ and $\mathbf{V}^{(j)}$. As we wish $\mathbf{V}^{(i)}$ and $\mathbf{V}^{(j)}$ to be far apart, we should minimize $HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)})$, so we add term $\sum_{i=1}^{m}\sum_{j=i+1}^{m} HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)})$ to the objective function. Then, we obtain the following formulation:

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{V}^{(i)}} \quad & \sum_{i=1}^{m} tr((\mathbf{Y}+\mathbf{V}^{(i)})^T\mathbf{L}^{(i)}(\mathbf{Y}+\mathbf{V}^{(i)})) + \lambda_1\sum_{i=1}^{m}\sum_{j=i+1}^{m} HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)}), \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}. \end{aligned} \tag{6}$$

where λ_1 is a balancing parameter to control the diversity.

Moreover, although each view may contain some complementary or distinct information, since we focus on the first family which aims to learn a consensus clustering result, the consensus embedding is the main part and also what we really want. So we wish each view contains **a small quantity of** distinct information, which means the variance $\mathbf{V}^{(i)}$ should be sparse. To make sure that, we impose ℓ_1-norm on each $\mathbf{V}^{(i)}$ and obtain:

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{V}^{(i)}} \quad & \sum_{i=1}^{m} tr((\mathbf{Y}+\mathbf{V}^{(i)})^T\mathbf{L}^{(i)}(\mathbf{Y}+\mathbf{V}^{(i)})) + \lambda_1\sum_{i=1}^{m}\sum_{j=i+1}^{m} HSIC(\mathbf{V}^{(i)},\mathbf{V}^{(j)}) + \lambda_2\sum_{i=1}^{m}\| \mathbf{V}^{(i)} \|_1, \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}. \end{aligned} \tag{7}$$

where λ_2 is another balancing parameter to control the sparsity.

Fig 1 shows a toy example, where $\mathbf{Y}^{(i)}$ is the embedding of i-th view and contains two parts: consensus part $\mathbf{Y}$ and distinct variance $\mathbf{V}^{(i)}$, i.e., $\mathbf{Y}^{(i)} = \mathbf{Y} + \mathbf{V}^{(i)}$. Fig 1(a) illustrates the ideal embedding we aim to learn. The orthogonal consensus embedding $\mathbf{Y}$ should contain most information, i.e., the distinct variance $\mathbf{V}^{(i)}$ only contains little non-zero elements. In addition, variances $\mathbf{V}^{(i)}$ contains the distinct information, and thus should be as different as possible

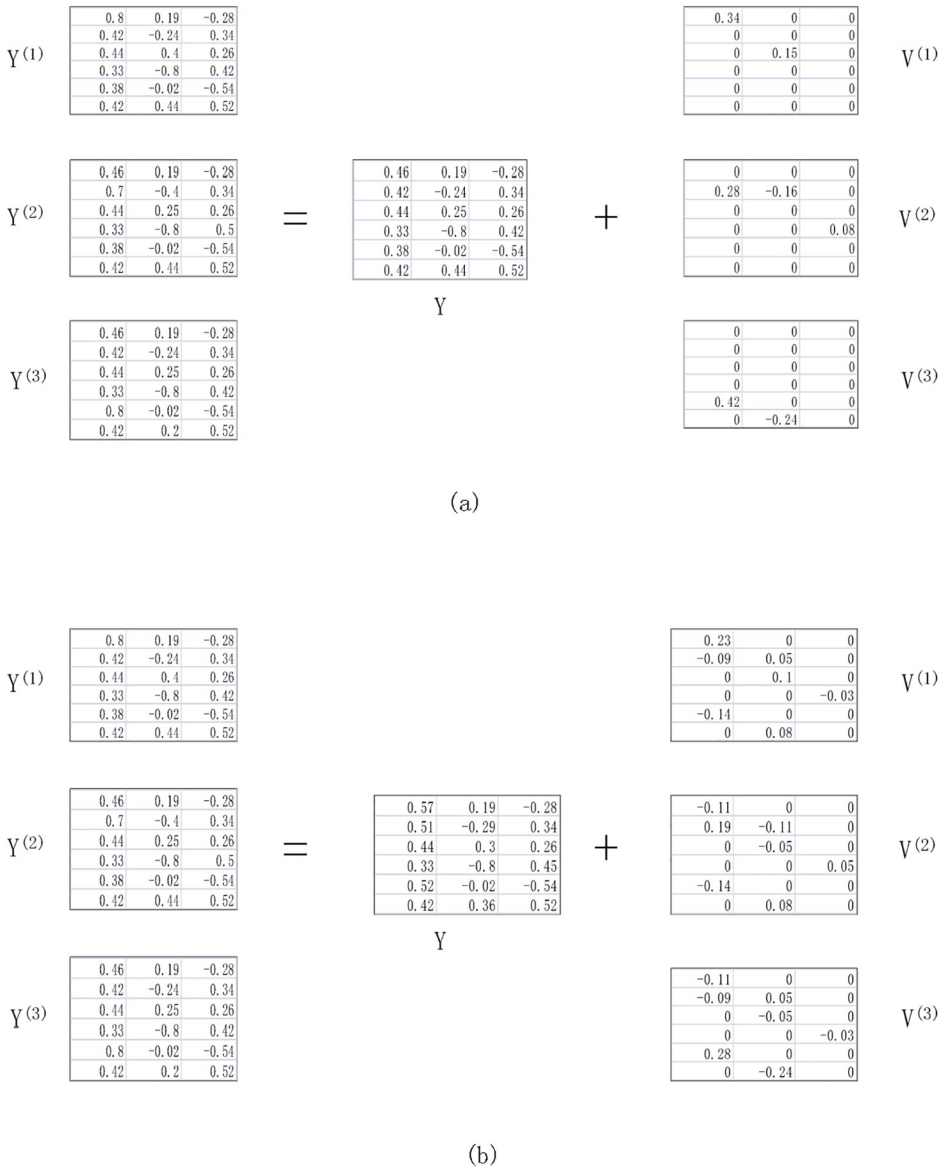

Fig 1. An example of the two parts of the embedding. (a) Consensus embedding and distinct variances satisfy our constraints. (b) Consensus embedding obtained by averaging all views without any constraints on the distinct variances.

https://doi.org/10.1371/journal.pone.0208494.g001

from each other as shown in Fig 1(a). Fig 1(b) shows the result that $\mathbf{Y} = 1/3\Sigma_i \mathbf{Y}^{(i)}$ without any constraints on $\mathbf{V}^{(i)}$. It is easy to verify that $\Sigma_{i,j} HSIC(\mathbf{V}^{(i)} + \mathbf{V}^{(j)})$ in Fig 1(a) is much smaller than that in Fig 1(b), which means $\mathbf{V}^{(i)}$ in Fig 1(a) is more like the distinct parts. Therefore, the consensus $\mathbf{Y}$ obtained by subtracting $\mathbf{V}^{(i)}$ from $\mathbf{Y}^{(i)}$ is cleaner in Fig 1(a).

It is worth noting that, [5] decomposes transition probability matrix of each view into a consensus transition probability matrix and a sparse noise matrix, which is similar to our method. However, the two methods are totally different. Firstly, the motivations are different. Their approach mainly considers robustness, while in our method, we try to discover the distinguishing information in each view. Secondly, in their method, they only impose sparsity on the noise matrices, while do not control the diversity, so it is not necessarily that the noise

matrices are far apart from each other. In our method, we explicitly control the diversity by minimizing the HSIC of each pair of views.

In Eq (7), we treat the difference of each pair of views equally, because in the term $\sum_{i=1}^{m}\sum_{j=i+1}^{m} HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)})$, the weights of all pairs $HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)})$ are all 1. However, in practice, if two views are more different, we wish the variance matrices of these two views to be farther apart. Therefore, we replace the term $\sum_{i=1}^{m}\sum_{j=i+1}^{m} HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)})$ with $\sum_{i=1}^{m}\sum_{j=i+1}^{m} \omega_{ij} HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)})$, where the pre-defined weight ω_{ij} is the prior information to capture the diversity between $\mathbf{V}^{(i)}$ and $\mathbf{V}^{(j)}$ for controlling the diversities more precisely. Intuitively, if the i-th view is more different from the j-th view, we need to impose larger weight ω_{ij} to keep $HSIC(\mathbf{V}^{(i)}, \mathbf{V}^{(j)})$ as small as possible. There are many ways to set ω_{ij}. In this paper, we first use the similarity matrices $\mathbf{S}^{(i)}$ and $\mathbf{S}^{(j)}$ to compute an average similarity score $\beta_{ij} \in [0, 1]$ of the i-th view and the j-th view. In more details, we compute $\beta_{ij} = \langle \mathbf{S}^{(i)}, \mathbf{S}^{(j)} \rangle / n^2$, i.e., β_{ij} is the normalized inner product of $\mathbf{S}^{(i)}$ and $\mathbf{S}^{(j)}$, which can represent the similarity of the i-th and the j-th view. Then we use the similar technique in [23] to get $\omega_{ij} = f(\beta_{ij}) = 1 - \frac{1}{1-log(\beta_{ij})}$. Since $f(\cdot)$ is monotonically decreasing, i.e., smaller β_{ij} leads to larger ω_{ij}, which satisfies the property of weight, that is if the two views are more different then the weight is larger.

Here, for simplicity, we use the linear kernel, i.e., $\mathbf{K}^{(i)} = \mathbf{V}^{(i)}\mathbf{V}^{(i)T}$. Taking Eq (4) into our objective function, we get the final formulation of our method:

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{V}^{(i)}} \quad & \sum_{i=1}^{m} tr((\mathbf{Y}+\mathbf{V}^{(i)})^T \mathbf{L}^{(i)} (\mathbf{Y}+\mathbf{V}^{(i)})) \\ & + \lambda_1 \sum_{i=1}^{m}\sum_{j=i+1}^{m} \omega_{ij} tr(\mathbf{H}\mathbf{V}^{(i)}\mathbf{V}^{(i)T}\mathbf{H}\mathbf{V}^{(j)}\mathbf{V}^{(j)T}) + \lambda_2 \sum_{i=1}^{m} \| \mathbf{V}^{(i)} \|_1 \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}. \end{aligned} \tag{8}$$

Note that for notational convenience, we absorb the scaling factor $(n-1)^{-2}$ of HSIC into the parameter λ_1. By explicitly capturing the variances $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(m)}$, Eq (8) can learn a clearer consensus spectral embedding $\mathbf{Y}$.

Optimization

Eq (8) involves $m + 1$ variables $(\mathbf{Y}, \mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(m)})$, thus we present a block coordinate descent scheme to optimize it. In particular, we optimize the objective w.r.t one variable while fixing the others. This procedure repeats until convergence.

Optimize $\mathbf{V}^{(i)}$ by fixing other variables. When $\mathbf{Y}, \mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(i-1)}, \mathbf{V}^{(i+1)}, \ldots, \mathbf{V}^{(m)}$ are fixed, Eq (8) can be rewritten as

$$\min_{\mathbf{V}^{(i)}} \; tr(\mathbf{V}^{(i)T}\mathbf{C}\mathbf{V}^{(i)}) + 2tr(\mathbf{V}^{(i)T}\mathbf{E}) + \lambda_2 \| \mathbf{V}^{(i)} \|_1$$

where $\mathbf{C}$ and $\mathbf{E}$ are defined as

$$\begin{aligned} \mathbf{C} &= \mathbf{L}^{(i)} + \lambda_1 \sum_{j \neq i} \omega_{ij} \mathbf{H}\mathbf{V}^{(j)}\mathbf{V}^{(j)T}\mathbf{H}, \\ \mathbf{E} &= \mathbf{L}^{(i)}\mathbf{Y}. \end{aligned} \tag{9}$$

For notational convenience, we use $\mathbf{V}$ to replace $\mathbf{V}^{(i)}$. Let

$$\mathcal{F}(\mathbf{V}) = tr(\mathbf{V}^T\mathbf{C}\mathbf{V}) + 2tr(\mathbf{V}^T\mathbf{E}),$$

it is easy to verify that the gradient of $\mathcal{F}$, denoted as $\nabla\mathcal{F}$, is Lipschitz continuous with some

constant Γ [24], i.e.,

$$\| \nabla\mathcal{F}(\mathbf{X}_1) - \nabla\mathcal{F}(\mathbf{X}_2)\|_F \leq \Gamma\| \mathbf{X}_1 - \mathbf{X}_2 \|_F \tag{10}$$

So we can optimize this subproblem with Accelerated Proximal Gradient Descent (APGD) method [25]. More specifically, instead of optimizing $\mathbf{V}$ by solving Eq (9) directly, we linearize $\mathcal{F}(\mathbf{V})$ at $\mathbf{V}^k$ (the result of $\mathbf{V}$ in the k-th iteration) and add a proximal term:

$$\mathcal{F}(\mathbf{V}) \approx \mathcal{F}(\mathbf{V}^k) + \langle\nabla\mathcal{F}(\mathbf{V}^k), \mathbf{V} - \mathbf{V}^k\rangle + \frac{\mu}{2}\| \mathbf{V} - \mathbf{V}^k \|_F^2$$

where $\mu > \Gamma$.

Then we update $\mathbf{V}^{k+1}$ by solving

$$\begin{aligned}\mathbf{V}^{k+1} &= \arg\min_{\mathbf{V}} \mathcal{F}(\mathbf{V}^k) + \langle\nabla\mathcal{F}(\mathbf{V}^k), \mathbf{V} - \mathbf{V}^k\rangle \\ &+ \frac{\mu}{2}\| \mathbf{V} - \mathbf{V}^k \|_F^2 + \lambda_2\| \mathbf{V} \|_1 \\ &= \arg\min_{\mathbf{V}} \frac{\mu}{2}\| \mathbf{V} - \left(\mathbf{V}^k - \frac{1}{\mu}\nabla\mathcal{F}(\mathbf{V}^k)\right) \|_F^2 + \lambda_2\| \mathbf{V} \|_1\end{aligned} \tag{11}$$

Let $\mathbf{Q} = \left(\mathbf{V}^k - \frac{1}{\mu}\nabla\mathcal{F}(\mathbf{V}^k)\right)$ and $\gamma = \frac{\lambda_2}{\mu}$, we can obtain $\mathbf{V}^{k+1}$ by solving

$$\arg\min_{\mathbf{V}} \| \mathbf{V} - \mathbf{Q} \|_F^2 + 2\gamma\| \mathbf{V} \|_1 \tag{12}$$

Eq (12) can be easily solved by thresholding algorithm and has a closed form solution $\tilde{\mathbf{V}}$:

$$\tilde{V}_{ij} = \begin{cases} sign(Q_{ij})(|Q_{ij}| - \gamma), & |Q_{ij}| \geq \gamma \\ 0, & |Q_{ij}| < \gamma \end{cases} \tag{13}$$

where $\tilde{V}_{ij}$ and Q_{ij} are the (i, j)-th element in matrix $\tilde{\mathbf{V}}$ and $\mathbf{Q}$ respectively, $sign(\cdot)$ is a sign function, i.e. $sign(x) = -1$ if x is negative and $sign(x) = 1$ if it is positive; $sign(x) = 0$, otherwise. Then we can set $\mathbf{V}^{k+1} = \tilde{\mathbf{V}}$. Until now, this is the process of Proximal Gradient Descent method. According to [25], we can update $\mathbf{V}^{k+1}$ as follows to get a faster convergence rate.

$$\alpha^{k+1} = \frac{1 + \sqrt{1 + 4(\alpha^k)^2}}{2} \tag{14}$$

$$\mathbf{V}^{k+1} = \tilde{\mathbf{V}} + \left(\frac{\alpha^k - 1}{\alpha^{k+1}}\right)(\tilde{\mathbf{V}} - \mathbf{V}^k) \tag{15}$$

where α^k is an auxiliary variable. Algorithm 1 provides the Accelerated Proximal Gradient Descent algorithm.

Algorithm 1 APGD for solving Eq (9)

Input: $\mathbf{C}$, $\mathbf{E}$, and the initial constant μ_0, $\mathbf{V}^1$.
Output: $\mathbf{V}$.
Initialize $\mu = \mu_0$, $\alpha^1 = 1$, $\rho = 1.02$, and $k = 1$.
while not converge **do**
  Calculate $\tilde{\mathbf{V}}$ by Eq (13).
  while μ is not appropriate **do**
    Set $\mu = \mu\rho$.
    Calculate $\tilde{\mathbf{V}}$ by Eq (13).
  end while

```
Set α^{k+1} and V^{k+1} by Eqs (14) and (15).
Set k = k + 1.
end while
```

Here ρ is a constant rate to update μ. We need to check whether μ is appropriate because μ should satisfy $\mu > \Gamma$, while in most cases, Γ is unknown. We can check it with the method in [25] and for saving space, we omit it here.

We show in the next theorem that this algorithm converges as $O\left(\frac{1}{k^2}\right)$ to the global optima of this subproblem.

Theorem 1. [24] *Let* $\mathbf{V}^k$ *be the sequence generated by Algorithm 1. Then for any* $k \geq 1$,

$$\mathcal{G}(\mathbf{V}^k) - \mathcal{G}(\mathbf{V}^*) \leq \frac{2\rho\Gamma \parallel \mathbf{V}^1 - \mathbf{V}^* \parallel_F^2}{(k+1)^2} \tag{16}$$

where $\mathcal{G}$ *is the objective function defined in* Eq (9) *and* $\mathbf{V}^* = \arg\min_{\mathbf{V}} \mathcal{G}(\mathbf{V})$ *is the global optima of* Eq (9).

Proof. See the proof of Theorem 4.4 in [24].

Optimize Y by fixing $\mathbf{V}^{(i)}$**.** When $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(m)}$ are fixed, we rewrite Eq (8) as follows

$$\begin{aligned} \min_{\mathbf{Y}} \quad & tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) + 2tr(\mathbf{Y}^T\mathbf{B}), \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}. \end{aligned} \tag{17}$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined as

$$\mathbf{A} = \sum_{i=1}^{m} \mathbf{L}^{(i)}, \quad \mathbf{B} = \sum_{i=1}^{m} \mathbf{L}^{(i)}\mathbf{Y}^{(i)}$$

To handle the constraints, we obtain the Lagrangian function of Eq (17) by introducing the Lagrangian multiplier $\mathbf{\Lambda}$,

$$\mathcal{L} = tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) + 2tr(\mathbf{Y}^T\mathbf{B}) - tr(\mathbf{\Lambda}(\mathbf{Y}^T\mathbf{Y} - \mathbf{I})) \tag{18}$$

Set the partial derivative with respect to $\mathbf{Y}$ to zero, we get

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Y}} = 2(\mathbf{A}\mathbf{Y} + \mathbf{B} - \mathbf{Y}\mathbf{\Lambda}) = \mathbf{0} \tag{19}$$

Multiplying both sides of Eq (19) by $\mathbf{Y}^T$ and using the fact that $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$, we can get $\mathbf{\Lambda} = \mathbf{Y}^T(\mathbf{A}\mathbf{Y} + \mathbf{B})$. Since $\mathbf{Y}^T\mathbf{Y}$ is symmetric, the Lagrangian multiplier $\mathbf{\Lambda}$ corresponding to $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$ is also symmetric, so we can rewrite it as $\mathbf{\Lambda} = (\mathbf{A}\mathbf{Y} + \mathbf{B})^T\mathbf{Y}$. Taking it into Eq (19), we have

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Y}} = 2\left(\mathbf{A}\mathbf{Y}\mathbf{Y}^T + \mathbf{B}\mathbf{Y}^T - \mathbf{Y}(\mathbf{A}\mathbf{Y} + \mathbf{B})^T\right)\mathbf{Y} \tag{20}$$

We denote $2(\mathbf{A}\mathbf{Y}\mathbf{Y}^T + \mathbf{B}\mathbf{Y}^T - \mathbf{Y}(\mathbf{A}\mathbf{Y} + \mathbf{B})^T)$ as $\mathbf{W}$, and have the following Lemma which shows the first order condition of Eq (17):

Lemma 1 $\frac{\partial \mathcal{L}}{\partial \mathbf{Y}} = \mathbf{0}$ *if and only if* $\mathbf{W} = \mathbf{0}$, *so* $\mathbf{W} = \mathbf{0}$ *is the first-order optimality condition of* Eq (17).

Proof. On one hand, according to the definition of $\mathbf{W}$, we have $\frac{\partial \mathcal{L}}{\partial \mathbf{Y}} = \mathbf{W}\mathbf{Y}$, so if $\mathbf{W} = \mathbf{0}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{Y}} = \mathbf{0}$.

On the other hand, if $\frac{\partial \mathcal{L}}{\partial \mathbf{Y}} = \mathbf{0}$, that is to say, $(\mathbf{A}\mathbf{Y}\mathbf{Y}^T + \mathbf{B}\mathbf{Y}^T - \mathbf{Y}(\mathbf{A}\mathbf{Y} + \mathbf{B})^T)\mathbf{Y} = \mathbf{0}$. Let $\mathbf{M} = \mathbf{A}\mathbf{Y} + \mathbf{B}$, then we have $\mathbf{M} = \mathbf{Y}\mathbf{M}^T\mathbf{Y}$. Furthermore,

$$\mathbf{M} = \mathbf{Y}\mathbf{M}^T\mathbf{Y} = \mathbf{Y}(\mathbf{Y}\mathbf{M}^T\mathbf{Y})^T\mathbf{Y} = \mathbf{Y}\mathbf{Y}^T\mathbf{M} \tag{21}$$

Taking the transposition of both sides of the equality, we have $\mathbf{M}^T = \mathbf{M}^T\mathbf{Y}\mathbf{Y}^T$. Then we obtain

$$\mathbf{Y}\mathbf{M}^T = \mathbf{Y}\mathbf{M}^T\mathbf{Y}\mathbf{Y}^T = \mathbf{M}\mathbf{Y}^T. \tag{22}$$

which is equal to $\mathbf{M}\mathbf{Y}^T - \mathbf{Y}\mathbf{M}^T = \mathbf{0}$. Note that $\mathbf{W} = 2(\mathbf{M}\mathbf{Y}^T - \mathbf{Y}\mathbf{M}^T)$, so $\mathbf{W} = \mathbf{0}$.

In summary, $\mathbf{W} = \mathbf{0}$ is the first-order optimality condition of our subproblem.

According to this, a natural way to update $\mathbf{Y}$ is gradient descent method which is $\mathbf{Y}^{k+1} \leftarrow \mathbf{Y}^k - \tau\mathbf{W}\mathbf{Y}^k$, where τ is the step size and $\mathbf{Y}^k$ is the result of $\mathbf{Y}$ at the k-th iteration. However, in this problem, since we have the orthogonal constraint on $\mathbf{Y}$, we cannot use gradient descent directly for the reason that gradient descent may violate the constraint. To overcome this problem, we use a constraint preserving descent method to update it inspired by [26–28]. In more details, we compute the initial $\mathbf{Y}$ which is denoted as $\mathbf{Y}^1$ by the following standard spectral clustering objective:

$$\begin{aligned} \min_{\mathbf{Y}} \quad & tr(\mathbf{Y}^{1T}\mathbf{A}\mathbf{Y}^1), \\ s.t. \quad & \mathbf{Y}^{1T}\mathbf{Y}^1 = \mathbf{I}. \end{aligned} \tag{23}$$

Then, we use the following constraint preserving descent formula to compute the new iteration $\mathbf{Y}$:

$$\mathbf{Y}^{k+1} = \mathbf{Y}^k - \tau\mathbf{W}\left(\frac{\mathbf{Y}^k + \mathbf{Y}^{k+1}}{2}\right) \tag{24}$$

Note that according to the definition of $\mathbf{W}$, we can easily verify that $\mathbf{W}^T = -\mathbf{W}$, which means that $\mathbf{W}$ is a skew-symmetric matrix. For a skew-symmetric matrix $\mathbf{W}$, we have the following theorem which gives a closed form solution of Eq (24) that satisfies the orthogonal constraint and updates $\mathbf{Y}$ in a descent direction. Moreover, due to Lemma 1, it converges to a stationary point.

Theorem 2. *1) Given any skew-symmetric matrix* $\mathbf{W} \in \mathbb{R}^{n\times n}$ *and* $\mathbf{Y}^k \in \mathbb{R}^{n\times c}$ *which satisfies* $\mathbf{Y}^{kT}\mathbf{Y}^k = \mathbf{I}$, *the closed form solution of matrix* $\mathbf{Y}^{k+1}$ *defined by* Eq (24) *is*

$$\mathbf{Y}^{k+1} = \left(\mathbf{I} + \frac{\tau}{2}\mathbf{W}\right)^{-1}\left(\mathbf{I} - \frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^k \tag{25}$$

and it satisfies that $\mathbf{Y}^{k+1T}\mathbf{Y}^{k+1} = \mathbf{I}$.

2) Set $\mathbf{W} = 2(\mathbf{A}\mathbf{Y}^k\mathbf{Y}^{kT} + \mathbf{B}\mathbf{Y}^{kT} - \mathbf{Y}^k(\mathbf{A}\mathbf{Y}^k + \mathbf{B})^T)$, *then*

$$\left.\frac{\partial \mathcal{J}(\mathbf{Y}^{k+1})}{\partial \tau}\right|_{\tau=0} = -\frac{1}{2}\| \mathbf{W} \|_F^2 \leq 0 \tag{26}$$

where $\mathcal{J}(\cdot)$ *is the objective function in* Eq (17), *and it means that updating* $\mathbf{Y}$ *is in a descent direction.*

3) This update formula converges to a stationary point of the subproblem.

Proof. 1) Moving all $\mathbf{Y}^{k+1}$ to the left side, we get $\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k+1}=\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k}$. Multiplying both sides by the inverse of $\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)^{-1}$, we obtain the closed form solution of $\mathbf{Y}^{k+1}$:

$$\mathbf{Y}^{k+1}=\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)^{-1}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k}$$

Then we calculate $\mathbf{Y}^{k+1T}\mathbf{Y}^{k+1}$

$$\begin{aligned}&\mathbf{Y}^{k+1T}\mathbf{Y}^{k+1}\\ =\;&\mathbf{Y}^{kT}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)^{T}\left(\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)^{T}\right)^{-1}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)^{-1}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k}\\ =\;&\mathbf{Y}^{kT}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)^{-1}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)^{-1}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k}\\ =\;&\mathbf{Y}^{kT}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\left(\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\right)^{-1}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k}\end{aligned}\tag{27}$$

where the second equality follows that $\mathbf{W}^{T}=-\mathbf{W}$ when $\mathbf{W}$ is a skew-symmetric matrix.

Furthermore, we have

$$\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)=\mathbf{I}-\frac{\tau^{2}}{4}\mathbf{W}\mathbf{W}=\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\tag{28}$$

Take Eq (28) into Eq (27),

$$\begin{aligned}&\mathbf{Y}^{k+1T}\mathbf{Y}^{k+1}\\ =\;&\mathbf{Y}^{kT}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\left(\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\right)^{-1}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k}\\ =\;&\mathbf{Y}^{kT}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\right)^{-1}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)^{-1}\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\right)\mathbf{Y}^{k}\\ =\;&\mathbf{Y}^{kT}\mathbf{Y}^{k}=\mathbf{I}\end{aligned}\tag{29}$$

2) According to the chain rule, we have

$$\frac{\partial\mathcal{J}(\mathbf{Y}^{k+1})}{\partial\tau}=tr\left(\left(\frac{\partial\mathcal{J}(\mathbf{Y}^{k+1})}{\partial\mathbf{Y}^{k+1}}\right)^{T}\frac{\partial\mathbf{Y}^{k+1}}{\partial\tau}\right)\tag{30}$$

When $\tau=0$, $\mathbf{Y}^{k+1}=\mathbf{Y}^{k}$, and $\frac{\partial\mathcal{J}(\mathbf{Y}^{k+1})}{\partial\mathbf{Y}^{k+1}}|_{\tau=0}=2(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})$, $\frac{\partial\mathbf{Y}^{k+1}}{\partial\tau}|_{\tau=0}=-\mathbf{W}\mathbf{Y}^{k}$, so on one hand,

$$\begin{aligned}&\left.\frac{\partial\mathcal{J}(\mathbf{Y}^{k+1})}{\partial\tau}\right|_{\tau=0}=-2tr\left((\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}\mathbf{W}\mathbf{Y}^{k}\right)\\ =\;&-4tr((\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})-(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}\mathbf{Y}^{k}(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}\mathbf{Y}^{k})\end{aligned}$$

On the other hand, we have

$$\begin{aligned}&\|\mathbf{W}\|_{F}^{2}=tr(\mathbf{W}^{T}\mathbf{W})\\ =\;&4tr(((\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})\mathbf{Y}^{kT}-\mathbf{Y}^{k}(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T})^{T}((\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})\mathbf{Y}^{kT}-\mathbf{Y}^{k}(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}))\\ =\;&8tr((\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})-(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}\mathbf{Y}^{k}(\mathbf{A}\mathbf{Y}^{k}+\mathbf{B})^{T}\mathbf{Y}^{k})\end{aligned}$$

So we have $\frac{\partial\mathcal{J}(\mathbf{Y}^{k+1})}{\partial\tau}|_{\tau=0}=-\frac{1}{2}\|\mathbf{W}\|_{F}^{2}$, which means if $\mathbf{Y}$ moves a small step $\Delta\tau>0$ in the update direction, the objective function $\mathcal{J}$ will have a change $-\frac{1}{2}\|\mathbf{W}\|_{F}^{2}\Delta\tau$ and $-\frac{1}{2}\|\mathbf{W}\|_{F}^{2}\leq 0$, so the objective function $\mathcal{J}$ will decrease. Thus the update direction is a descent direction.

3) Since the objective function deceases monotonically and the objective function is lower bounded by 0, the iteration method converges. When it converges, which means $\| \mathbf{W} \|_F^2 = 0$ when $\tau = 0$, i.e., $\mathbf{W} = \mathbf{0}$, it satisfies the first-order optimality condition of our subproblem according to Lemma 1. So the algorithm can converge to a stationary point of this subproblem.

Note that we choose the iteration step size τ by a curvilinear search method as was done in [28], which can guarantee the convergence. Therefore, we compute $\mathbf{Y}^{k+1}$ iteratively using the update formula Eq (25) until the decent process converges. Clearly, the computationally heaviest step in this algorithm is to compute the matrix inverse $\left(\mathbf{I} + \frac{\tau}{2}\mathbf{W}\right)^{-1}$, which is $O(n^3)$. Fortunately, we can find a fast way to calculate it. By the definition of $\mathbf{W}$, we rewrite $\frac{\tau}{2}\mathbf{W} = \mathbf{UG}$, where $\mathbf{U} = \frac{\tau}{2}[\mathbf{AY}^k + \mathbf{B}, -\mathbf{Y}^k]$ and $\mathbf{G} = [\mathbf{Y}^k, \mathbf{AY}^k + \mathbf{B}]^T$, then according to [29] we have

$$(\mathbf{I} + \mathbf{UG})^{-1} = \mathbf{I} - \mathbf{U}(\mathbf{I} + \mathbf{GU})^{-1}\mathbf{G} \tag{31}$$

Since $\mathbf{I} + \mathbf{GU}$ is a $2c \times 2c$ matrix, where c is the dimension of the embedding space and often has $c \ll n$, we can efficiently compute the original inverse by matrix multiplication ($O(n^2 c)$) and the inverse of a much smaller matrix ($O(c^3)$).

After getting $\mathbf{Y}, \mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(m)}$, we use spectral rotation [20] to discretize $\mathbf{Y}$ to get the final clustering result. Algorithm 2 summarizes the whole algorithm.

Algorithm 2 Algorithm of DCSC

Input: Multi-view data $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}$, λ_1, λ_2.
Output: Clustering result **R**.
1: Construct Laplacian matrix $\mathbf{L}^{(i)}$ for each view.
2: **while** not converge **do**
3: Compute $\mathbf{V}^{(i)}$ (i = 1, 2, ..., m) by Algorithm 1.
4: Compute **Y** by Constraint Preserving Descent method.
5: **end while**
6: **R** = *discrete*(**Y**).

Convergence analysis and time complexity

According to Theorem 1 and Theorem 2, no matter updating $\mathbf{Y}$ or $\mathbf{V}^{(i)}$, the objective function decreases monotonically. Moreover, the objective function has a lower bound 0. Thus Algorithm 2 converges. In fact, this algorithm converges very fast (within several ten iterations in practice).

Since we decreases the time complexity of matrix inverse from $O(n^3)$ to $O(n^2c + c^3)$, the computationally heaviest step is matrix multiplication. Among all matrix multiplication in our method, the highest time complexity is $O(n^2c)$ which is generated in the multiplication of an $n \times n$ and an $n \times c$ matrix. So the time complexity is square in the number of instances.

Experiments

In this section, we evaluate the effectiveness of DCSC by comparing it with several state-of-the-art multi-view clustering methods on benchmark data sets.

Data sets

We use totally 8 data sets to evaluate the effectiveness of our method, including WebKb data set [30], which contains webpages collected from four universities: *Cornell*, *Texas*, *Washington* and *Wisconsin* and is available in http://membres-liglab.imag.fr/grimal/data.html; UCI handwritten digit data set [5] which can be found in http://archive.ics.uci.edu/ml/datasets/Multiple+Features; Advertisements data set [31] which is published in http://archive.ics.uci.edu/ml/

Table 1. Details of the multi-view data sets used in our experiments (feature type (dimensionality)).

Feature type	Cornell	Texas	Washington	Wisconsin
1	Cont(1703)	Cont(1703)	Cont(1703)	Cont(1703)
2	Cite(195)	Cite(187)	Cite(230)	Cite(265)
#Instances	195	187	230	265
#Clusters	5	5	5	5
Feature type	UCI Digit	Advertisements	Corel	Flower17
1	FourierCoef(76)	ImageURL(457)	ColorHistogram(64)	ColorVocabulary
2	ProfileCorrelations(216)	BaseURL(495)	ColorMoment(9)	ShapeVocabulary
3	Pixel(240)	DestinationURL(472)	ColorCoherence(128)	TextureVocabulary
4	KarhunenLoeveCoef(64)	Alt(111)	CoarsnessTamuraTexture(10)	HSV
5	ZernikeMoments(47)	Caption(19)	DirectionalityTamuraTexture(8)	HOG
6	Morphological(6)	-	WaveletTexture(104)	ForegroundSIFT
7	-	-	MASARTexture(15)	BoundarySIFT
#Instances	2000	3279	3400	1360
#Clusters	10	2	34	17

https://doi.org/10.1371/journal.pone.0208494.t001

datasets/Internet+Advertisements; Corel image data set [32] which can be found in http://www.cs.virginia.edu/~xj3a/research/CBIR/Download.htm; and Flower17 data set [33] which is available in http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html. Statistics of these data sets are summarized in Table 1. Since Flower17 data set only contains 7 distance matrices constructed by the 7 views, we do not show the dimension of each view in Table 1.

Compared methods

To demonstrate the effectiveness of our method, we compare DCSC with the following algorithms:

- **FeaConcat**, which first concatenates features in all views and then applies spectral clustering on it.
- **RMKMC** [1], which is a robust k-means based multi-view clustering method.
- **MultiNMF** [2], which is a nonnegative matrix factorization based multi-view clustering method.
- **Co-reg SC** [4], which is a co-regularized multi-view spectral clustering method.
- **RMSC** [5], which is a robust multi-view spectral clustering with sparse low-rank decomposition.
- **AMGL** [6], which is a parameter-free multi-view spectral clustering method, i.e., it learns an optimal weight for each view automatically without introducing an additive parameter.
- **SwMC** [34], which is a self-weighted multi-view clustering method with multiple graphs.
- **RAMC** [35], which is a robust auto-weighted multi-view clustering method.
- **DCSC-ω**. To show the effect of the prior weight ω in our method, we remove the ω (or equivalently, we set all ω_{ij} to 1) and obtain DCSC-ω.

Table 2. ACC results on all the data sets.

Data	FeaConcat	RMKMC	MultiNMF	Co-reg SC	RMSC	AMGL	SwMC	RAMC	DCSC-ω	DCSC
Cornell	0.4513	0.4492	0.4256	0.4697	0.4270	0.4451	0.4821	0.4256	0.5026	**0.5385**
Texas	0.3797	0.5599	0.5561	0.5775	0.5892	0.5968	0.5508	0.5508	0.6524	**0.6738**
Washington	0.4783	0.5613	0.4739	0.5691	0.5536	0.5217	0.5130	0.5500	0.6870	**0.6957**
Wisconsin	0.4528	0.5102	0.4717	0.5391	0.5155	0.5725	0.4712	0.4717	0.5736	**0.6189**
UCI Digit	0.7235	0.7579	0.7991	**0.8763**	0.8048	0.7493	0.8450	0.8325	0.8595	**0.8790**
Advertisements	0.7091	0.8618	0.8603	0.7995	0.6908	0.8446	0.8134	0.8600	0.8713	**0.9161**
Corel	0.2362	0.1094	0.0297	0.2842	0.2846	0.2937	0.1485	0.2756	0.3209	**0.3612**
Flower17	-	-	-	0.5325	0.5641	0.5734	0.4860	0.5713	0.5647	**0.5912**

https://doi.org/10.1371/journal.pone.0208494.t002

Table 3. NMI results on all the data sets.

Data	FeaConcat	RMKMC	MultiNMF	Co-reg SC	RMSC	AMGL	SwMC	RAMC	DCSC-ω	DCSC
Cornell	0.1628	0.1704	0.0199	**0.1960**	0.1623	0.1352	0.1349	0.1361	0.1594	**0.1853**
Texas	0.2002	0.1910	0.0288	0.2495	0.2317	0.1821	0.2569	0.2413	0.2323	**0.2744**
Washington	0.2321	0.2550	0.0206	0.2739	0.2729	0.2182	0.2215	0.2416	0.2881	**0.3042**
Wisconsin	0.2476	0.2498	0.0202	0.2834	0.2780	**0.3090**	0.2412	0.2359	0.2773	0.2819
UCI Digit	0.6925	0.7245	0.7186	0.8221	0.7467	0.7166	**0.8942**	**0.8857**	0.8735	**0.8881**
Advertisements	0.0189	0.0319	0.0015	0.0725	0.0196	0.1619	0.2286	0.2216	0.2247	**0.2923**
Corel	0.3172	0.1156	0.0102	0.3741	0.3603	0.3644	0.1966	0.3645	0.3777	**0.4189**
Flower17	-	-	-	0.5502	0.4491	**0.5721**	0.5570	0.5593	**0.5730**	**0.5752**

https://doi.org/10.1371/journal.pone.0208494.t003

Experiment setup

The number of clusters is set to the true number of classes for all data sets and all methods. Since the results of most compared algorithms depend on the initializations, we independently repeat the experiments for 10 times and report the average results and *t*-test results. In our method, we tune λ_1 in $[10^{-5}, 10^5]$ and tune λ_2 in $[10^{-4}, 10^4]$ by grid search. Note that, λ_1 absorbs the scaling factor $(n-1)^{-2}$ in it which is dependent on the size of the data set, and in our experiments, the size is in the range between 100 to 5000, which is relatively narrow. Therefore we tune it in a wider range $[10^{-5}, 10^5]$. Of course, we can also use other parameter tuning strategy, for example, let $\hat{\lambda} = \lambda_1 * (n-1)^2$, so that $\hat{\lambda}$ is independent of n and we tune $\hat{\lambda}$ in a predefined range. For other compared methods, we tune the parameters as suggested in their papers. Three clustering evaluation metrics are adopted to measure the clustering

Table 4. Purity results on all the data sets.

Data	FeaConcat	RMKMC	MultiNMF	Co-reg SC	RMSC	AMGL	SwMC	RAMC	DCSC-ω	DCSC
Cornell	0.4821	0.5046	0.4359	**0.5487**	0.5117	0.4872	0.5385	0.4256	0.5179	**0.5436**
Texas	0.5580	0.6219	0.5668	**0.6727**	0.6574	0.6043	0.5936	0.5668	0.6417	**0.6791**
Washington	0.6174	0.6739	0.4826	0.6817	0.6730	0.6517	0.5435	0.6652	0.6783	**0.7000**
Wisconsin	0.6038	0.6589	0.4755	0.6894	0.6729	**0.7042**	0.6019	0.6004	0.6604	**0.6943**
UCI Digit	0.7435	0.7892	0.7996	**0.8806**	0.8166	0.7704	**0.8820**	0.8675	0.8620	**0.8790**
Advertisements	0.8600	0.8625	0.8603	0.8667	0.8600	0.8631	0.8750	0.8731	0.8731	**0.9161**
Corel	0.2700	0.1163	0.0391	0.3069	0.3105	0.3330	0.1874	0.3262	0.3712	**0.3924**
Flower17	-	-	-	0.5604	0.5768	**0.5966**	0.5000	0.5868	0.5875	**0.6007**

https://doi.org/10.1371/journal.pone.0208494.t004

performance, including clustering Accuracy (ACC), Normalized Mutual Information (NMI) and clustering Purity.

Experimental results

Tables 2–4 show the ACC, NMI and Purity results of all methods on all data sets, respectively. Bold font indicates that the difference is statistically significant (the p-value of t-test is smaller than 0.05). Note that since Flower17 data set only contains distance matrices instead of the original features, we construct Laplacian matrices from distance matrices and only compare our method with spectral clustering based methods.

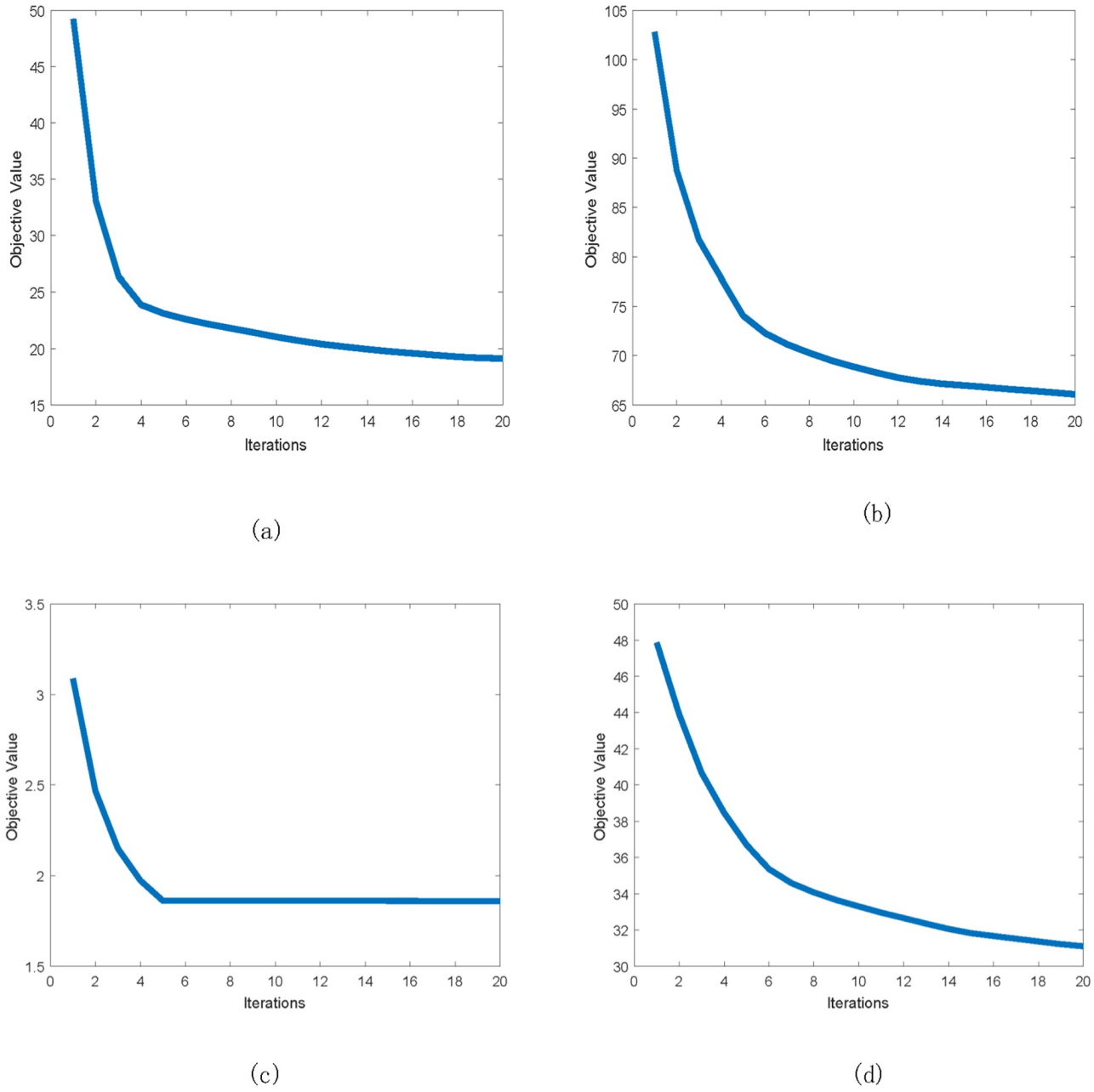

Fig 2. Convergence curve of our method. (a) UCI Digit data set. (b) Corel data set. (c) Advertisements data set. (d) Flower17 data set.

https://doi.org/10.1371/journal.pone.0208494.g002

In Tables 2–4, on most data sets, the performance of spectral based methods (Co-reg SC, RMSC, AMGL, SwMC, RAMC and ours) are much better than spectral clustering on feature concatenating, which demonstrates the effectiveness of multi-view clustering. Moreover, our method outperforms other compared methods on most data sets, which means that taking divergence of each view into consideration is indeed helpful to multi-view clustering.

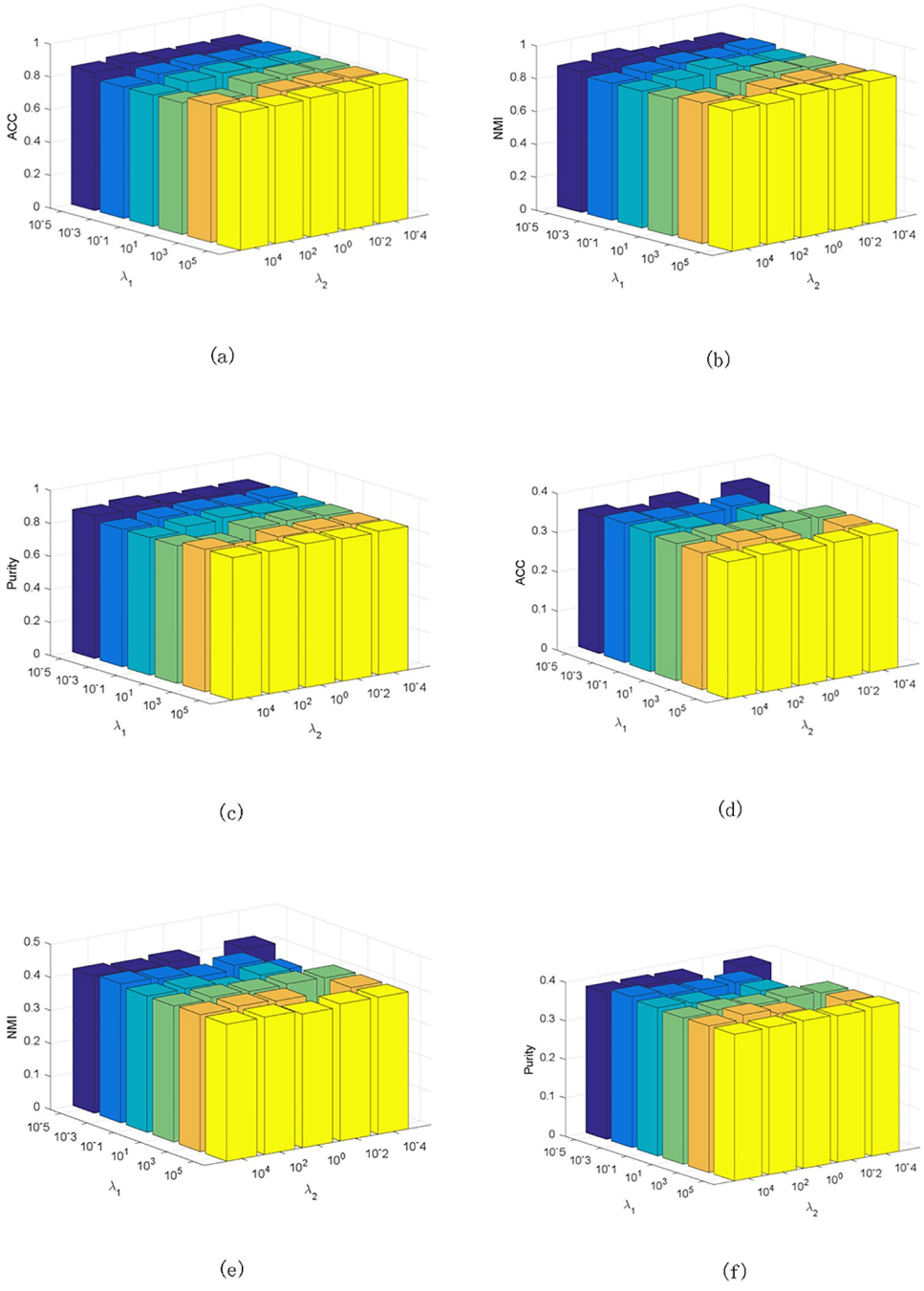

Fig 3. Clustering results w.r.t. λ_1, λ_2 on UCI Digit and Corel data set. (a) ACC on UCI Digit data set. (b) NMI on UCI Digit data set. (c) Purity on UCI Digit data set. (d) ACC on Corel data set. (e) NMI on Corel data set. (f) Purity on Corel data set.

https://doi.org/10.1371/journal.pone.0208494.g003

Especially on Corel data set, which is the most difficult for clustering (it has the most views (7 views) and the most classes (34 classes)), our method has 23%, 12%, and 18% improvements on the second best method on ACC, NMI and Purity, respectively. In our method, since we explicitly capture the distinct variances of all views by minimizing the dependency among them, the remainder is a clearer consensus spectral embedding leading to a better clustering result. Note that, DCSC also obtain better results compared with DCSC-ω, which means considering the similarity between each view can improve the performance of our method.

We show the algorithm convergence on UCI Digit, Advertisements, Corel and Flower17 data set in Fig 2, and the results on other data sets are similar. The example result in Fig 2 shows that our method converges within a small number of iterations, which empirically demonstrates our claims in the previous section.

Parameter study

We explore the effect of the parameters on clustering performance. There are two parameters in our method: λ_1 and λ_2. We show the ACC, NMI and Purity on UCI Digit and Corel data sets and the results are similar on other data sets. Fig 3 shows the results, from which we can see that the performance of our method is stable across a wide range of the parameters.

Conclusion

In this paper, we proposed a novel multi-view spectral clustering method DCSC. We explicitly captured the distinguishing or complementary information in each view and took advantage of the distinct information to learn a better consensus clustering result. To characterize the distinct information effectively, we use HSIC to control the diversity of each view. Since the introduced optimization problem contains several variables, we presented a block coordinate descent algorithm to solve it and proved its convergence. Finally, experiments on benchmark data sets show that the proposed method outperforms the state-of-the-art multi-view clustering methods.

Since the scalability is a serious problem in spectral clustering, in the future, we will study scalability issue with multi-view spectral clustering and apply it to large-scale data sets.

Supporting information

S1 Dataset. Cornell dataset.
(TXT)

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions.

Author Contributions

Conceptualization: Peng Zhou.

Formal analysis: Peng Zhou, Fan Ye.

Funding acquisition: Liang Du.

Software: Peng Zhou.

Writing – original draft: Peng Zhou.

Writing – review & editing: Fan Ye, Liang Du.

References

1. Cai X, Nie F, Huang H. Multi-view k-means clustering on big data. In: Proceedings of the Twenty-Third IJCAI. AAAI Press; 2013. p. 2598–2604.
2. Liu J, Wang C, Gao J, Han J. Multi-view clustering via joint nonnegative matrix factorization. In: Proc. of SDM. vol. 13. SIAM; 2013. p. 252–260.
3. Kumar A, Daumé H. A co-training approach for multi-view spectral clustering. In: ICML; 2011. p. 393–400.
4. Kumar A, Rai P, Daume H. Co-regularized multi-view spectral clustering. In: Advances in NIPS; 2011. p. 1413–1421.
5. Xia R, Pan Y, Du L, Yin J. Robust multi-view spectral clustering via low-rank and sparse decomposition. In: AAAI; 2014. p. 2149–2155.
6. Nie F, Li J, Li X. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification. International Joint Conferences on Artificial Intelligence; 2016.
7. Nie F, Cai G, Li J, Li X. Auto-Weighted Multi-View Learning for Image Clustering and Semi-Supervised Classification. IEEE Transactions on Image Processing. 2018; 27(3):1501–1511. https://doi.org/10.1109/TIP.2017.2754939
8. Zhang Z, Zhai Z, Li L. Uniform Projection for Multi-View Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017; 39(8):1675–1689. https://doi.org/10.1109/TPAMI.2016.2601608
9. Tang C, Chen J, Liu X, Li M, Wang P, Wang M, et al. Consensus learning guided multi-view unsupervised feature selection. Knowledge-Based Systems. 2018; 160:49–60. https://doi.org/10.1016/j.knosys.2018.06.016
10. Cao X, Zhang C, Fu H, Liu S, Zhang H. Diversity-Induced Multi-View Subspace Clustering; 2015.
11. Günnemann S, Färber I, Seidl T. Multi-view clustering using mixture models in subspace projections. In: The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'12, Beijing, China, August 12-16, 2012; 2012. p. 132–140.
12. Niu D, Dy JG, Jordan MI. Multiple Non-Redundant Spectral Clustering Views. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel; 2010. p. 831–838.
13. Xu C, Tao D, Xu C. A survey on multi-view learning. arXiv preprint arXiv:13045634. 2013;.
14. Zhao J, Xie X, Xu X, Sun S. Multi-view learning overview: Recent progress and new challenges. Information Fusion. 2017; 38:43–54. https://doi.org/10.1016/j.inffus.2017.02.007
15. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. ACM; 1998. p. 92–100.
16. Wang W, Zhou ZH. Analyzing co-training style algorithms. In: Machine Learning: ECML 2007. Springer; 2007. p. 454–465.
17. Ng AY, Jordan MI, Weiss Y. On Spectral Clustering: Analysis and an algorithm. In: Dietterich TG, Becker S, Ghahramani Z, editors. Advances in Neural Information Processing Systems 14. MIT Press; 2002. p. 849–856.
18. Liu J, Jiang Y, Li Z, Zhou ZH, Lu H. Partially shared latent factor learning with multiview data. IEEE Trans Neural Netw Learning Syst. 2015; 26(6):1233–1246. https://doi.org/10.1109/TNNLS.2014.2335234
19. Shi J, Malik J. Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2000; 22(8):888–905. https://doi.org/10.1109/34.868688
20. Yu SX, Shi J. Multiclass spectral clustering. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE; 2003. p. 313–319.
21. Scholkopf B, Smola A. Learning with kernels. MIT press Cambridge. 2002;.
22. Gretton A, Bousquet O, Smola A, Schölkopf B. Measuring statistical dependence with Hilbert-Schmidt norms. In: PROCEEDINGS ALGORITHMIC LEARNING THEORY. Springer-Verlag; 2005. p. 63–77.
23. Tang J, Hu X, Gao H, Liu H. Exploiting local and global social context for recommendation. In: Proceedings of the Twenty-Third IJCAI. AAAI Press; 2013. p. 2712–2718.
24. Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences. 2009; 2(1):183–202. https://doi.org/10.1137/080716542
25. Ji S, Ye J. An accelerated gradient method for trace norm minimization. In: Proceedings of the 26th annual international conference on machine learning. ACM; 2009. p. 457–464.

26. Goldfarb D, Wen Z, Yin W. A curvilinear search method for p-harmonic flows on spheres. SIAM Journal on Imaging Sciences. 2009; 2(1):84–109. https://doi.org/10.1137/080726926
27. Vese LA, Osher SJ. Numerical methods for p-harmonic flows and applications to image processing. SIAM Journal on Numerical Analysis. 2002; 40(6):2085–2104. https://doi.org/10.1137/S0036142901396715
28. Wen Z, Yin W. A feasible method for optimization with orthogonality constraints. Mathematical Programming. 2013; 142(1-2):397–434. https://doi.org/10.1007/s10107-012-0584-1
29. Petersen KB, Pedersen MS. The matrix cookbook. Technical University of Denmark. 2008; 7:15.
30. Li SY, Yuan J, Zhou ZH. Partial multi-view clustering. In: AAAI; 2014.
31. Kushmerick N. Learning to remove internet advertisements. In: Proceedings of the third annual conference on Autonomous Agents. ACM; 1999. p. 175–181.
32. French JC, Watson JV, Jin X, Martin W. Integrating multiple multi-channel CBIR systems. In: In: Proc. Inter. Workshop on Multimedia Information Systems (MIS). Citeseer; 2003.
33. Nilsback ME, Zisserman A. Automated Flower Classification over a Large Number of Classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing; 2008.
34. Nie F, Li J, Li X. Self-weighted multiview clustering with multiple graphs. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence; 2017. p. 2564–2570.
35. Ren P, Xiao Y, Xu P, Guo J, Chen X, Wang X, et al. Robust Auto-Weighted Multi-View Clustering. In: IJCAI; 2018. p. 2644–2650.