

## CGD: Multi-View Clustering via Cross-View Graph Diffusion

Chang Tang,<sup>1</sup> Xinwang Liu,<sup>2</sup> Xinzhong Zhu,<sup>3</sup> En Zhu,<sup>2</sup> Zhigang Luo,<sup>2</sup> Lizhe Wang,<sup>1</sup> Wen Gao<sup>4</sup>

<sup>1</sup>School of Computer Science, China University of Geosciences, Wuhan 430074, China

<sup>2</sup>School of Computer Science, National University of Defense Technology, Changsha 410073, China

<sup>3</sup>College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China

<sup>4</sup>School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

<https://github.com/ChangTang/CGD>

### Abstract

Graph based multi-view clustering has been paid great attention by exploring the neighborhood relationship among data points from multiple views. Though achieving great success in various applications, we observe that most of previous methods learn a consensus graph by building certain data representation models, which at least bears the following drawbacks. First, their clustering performance highly depends on the data representation capability of the model. Second, solving these resultant optimization models usually results in high computational complexity. Third, there are often some hyper-parameters in these models need to tune for obtaining the optimal results. In this work, we propose a general, effective and parameter-free method with convergence guarantee to learn a unified graph for multi-view data clustering via cross-view graph diffusion (CGD), which is the first attempt to employ diffusion process for multi-view clustering. The proposed CGD takes the traditional predefined graph matrices of different views as input, and learns an improved graph for each single view via an iterative cross diffusion process by 1) capturing the underlying manifold geometry structure of original data points, and 2) leveraging the complementary information among multiple graphs. The final unified graph used for clustering is obtained by averaging the improved view associated graphs. Extensive experiments on several benchmark datasets are conducted to demonstrate the effectiveness of the proposed method in terms of seven clustering evaluation metrics.

### Introduction

It is not uncommon that an object is usually described by multi-view features. For example, in image/video processing, different visual descriptors such as Local Binary Patterns (LBP) (Ojala, Pietikainen, and Maenpaa 2002), Scale Invariant Feature Transform (SIFT) (Lowe and Lowe 2004) and Histogram of Oriented Gradient (HOG) (Dalal and Triggs 2005) are often used to describe each image/video frame from different views. In biomedical research, both the chemical structure and chemical response in different cells can be used to represent a certain drug, while the sequence and gene expression values can represent a certain protein in different aspects (Li 2014; Li and Cai 2017).

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Multi-view clustering (Tang et al. 2019c; Liu et al. 2018; Zhang et al. 2019; Tang et al. 2019b; 2019a; 2019d), which partitions these multi-view data into different groups by using the complementary information of multi-view feature sets to ensure that highly similar instances are divided into the same group, is an important branch of multi-view learning (Wang, Nie, and Huang 2013). In general, most of previous multi-view clustering methods employ graph-based models since the similarity graph can characterize the data structure effectively. These methods usually learn a common sample similarity graph  $\mathbf{W}$  firstly by deploying features of different views (as shown by Fig. 1a), and then the clustering result is obtained by carrying out the spectral clustering algorithm based on  $\mathbf{W}$  (Cao et al. 2015; Wang et al. 2017; Zhang et al. 2017; Tang et al. 2019c; Wang and Wu 2018; Yang et al. 2017) or obtained directly from  $\mathbf{W}$  without performing any post-processing if  $\mathbf{W}$  is constrained to be with exactly  $k$  connected components, where  $k$  is the number of clusters (Zhan et al. 2018; 2019; Wang, Yang, and Liu 2019; Yang et al. 2019).

In recent years, numbers of graph learning based multi-view clustering methods have been proposed. Due to the page limitation, we just list a few of them here. By regarding the subspace representation matrices of different views as a tensor, Zhang et al. (Zhang et al. 2015) proposed low-rank tensor constrained multi-view subspace clustering (LT-MS) to exploit the high order relation implied in multi-view data. In LT-MS, the low-rank constraint is applied to the constructed tensor for capturing the cross-view information and reducing the redundancy among different views. In order to exploit the complementary information among multiple feature views, Cao et al. (Cao et al. 2015) made use of the Hilbert Schmidt Independence Criterion (HSIC) to exploit the diversity of the new representations of different views. Different from most of previous subspace clustering methods which reconstruct data points using original features, Zhang et al. (Zhang et al. 2017) proposed latent multi-view subspace clustering which seeks the underlying latent representation and assumes that all of different feature views are projected from the learned latent feature space. The unified data representation matrix of different views is simultaneously learned from the latent space. Wang et al. (Wang et

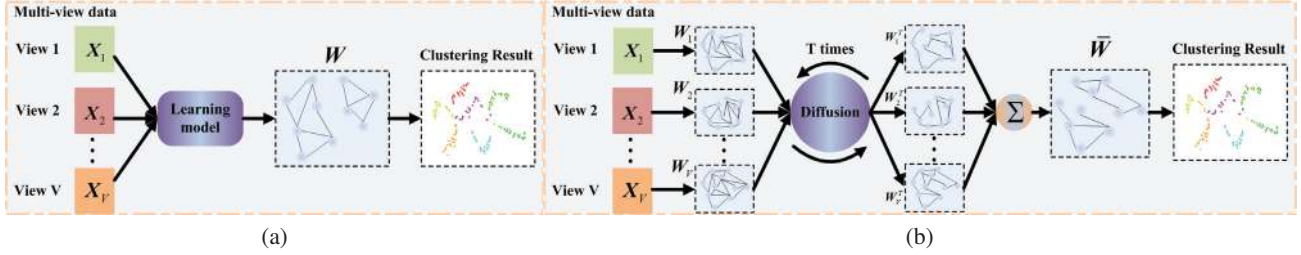


Figure 1: Traditional data representation learning based method (a) and our graph diffusion based method (b). In data representation learning based method, a common similarity graph is learned by using original data. In our CGD method, we aim to directly learn a unified graph from different pre-defined view-specific graphs.

al. 2017) combined subspace learning and spectral clustering together and introduced a novel exclusivity-consistency regularized multi-view subspace clustering model, which simultaneously encourages the representation exclusivity as well as the indicator consistency. In (Zhan et al. 2018; 2019; Wang, Yang, and Liu 2019), a unified graph with an rank constraint on its Laplacian matrix is learned from different views, then the cluster indicators are obtained directly from the global graph without performing any graph cut technique or the  $k$ -means clustering.

Although promising results have been attained by previous graph learning based multi-view clustering methods in various applications, their clustering performance highly depends on the quality of graph learning models, and solving these learning models usually results in high computational complexity. In addition, there are often some hyper-parameters in these models need to tune for obtaining the optimal results. In this work, we attempt to learn a unified graph for multi-view data clustering via cross-view graph diffusion (CGD). Instead of learning a consensus graph from original multi-view features, CGD takes the traditional predefined graph matrices of different views as input, and learns an improved graph for each single view via an iterative cross diffusion process. Compared to previous methods which learn the consensus graph from original features, CGD directly learns the unified graph from multiple predefined graphs, it has three advantages: 1) CGD can capture the underlying manifold geometry structure of original data points from different views, and 2) leverage the complementary information among multiple graphs instead of original multi-view data, which is more intuitive since the multiple graphs directly characterize the relationship of data points in different aspects, and 3) it is parameter-free when the predefined view-specific graphs are given. Fig. 1b gives a brief outline of our proposed method.

In brief, the main contributions of this work can be summarized as follows:

- We design a general and effective unified graph learning method with proven convergence for multi-view clustering via cross-view graph diffusion, which can intuitively capture the complementary information from multiple predefined view-specific graphs. As far as we know, our work is the first attempt to employ diffusion process

for multi-view data clustering;

- Our proposed method is parameter-free when the predefined view-specific graphs are given, i.e., no parameter need to tune for obtaining the final optimal results;
- Extensive experiments on both toy datasets and real-world datasets are conducted to validate the superiority of our proposed method in terms of different clustering evaluation metrics.

## The Proposed CGD Method

### Notations and Problem Definition

Throughout this paper, boldface capital letters denote matrices. Given an arbitrary matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{M}_{i,j}$  or  $\mathbf{M}(i,j)$  denotes its  $(i,j)$ -th entry.  $\mathbf{M}^T$  is the transpose of  $\mathbf{M}$ .  $\mathbf{I}_m$  represents an  $m \times m$  identity matrix (abbreviated as  $\mathbf{I}$  if the size is obviously known).

Suppose we have  $N$  data instances captured from  $V$  different views.  $\mathbf{X} = [\mathbf{X}_1; \mathbf{X}_2; \dots; \mathbf{X}_V] \in \mathbb{R}^{D \times N}$  is the multi-view data matrix which consists of  $V$  different views, where  $\mathbf{X}_v \in \mathbb{R}^{D_v \times N}$  represents the  $v$ -th view and  $D_v$  represents its feature dimension with  $\sum_{v=1}^V D_v = D$ . The aim of multi-view clustering is to classify these data points into different groups by using the multi-view information. For the  $v$ -th view, we generate a weighted graph  $\mathcal{G}_v = (\mathbf{X}_v, \mathbf{W}_v)$ , where  $\mathbf{W}_v$  is the graph adjacency matrix which is calculated by using certain distance metric such as the Gaussian kernel function and cos similarity. CGD aims to generate a unified graph  $\mathcal{G} = (\mathbf{X}, \mathbf{W})$  for multi-view clustering by using the complementarity among multiple graphs  $\mathcal{G}_v, v = 1, 2, \dots, V$ .

### Diffusion Process Revisiting

Given a weighted graph  $\mathcal{G} = (\mathbf{X}, \mathbf{W})$ , diffusion process aims to learn a more faithful similarity matrix  $\mathbf{A}$ . Motivated by the manifold ranking model (Zhou et al. 2004), Bai et al. (Bai et al. 2017) proposed a regularized diffusion process (RDP) for visual retrieval, which obtains the new similarity measure  $\mathbf{A}$  as the closed-form solution of the following optimization

problem:

$$\min_{\mathbf{A}} \frac{1}{2} \sum_{i,j=1}^N \sum_{p,q=1}^N \mathbf{W}_{ij} \mathbf{W}_{pq} \left( \frac{\mathbf{A}_{ip}}{\sqrt{\mathbf{D}_{ii} \mathbf{D}_{pp}}} - \frac{\mathbf{A}_{jq}}{\sqrt{\mathbf{D}_{jj} \mathbf{D}_{qq}}} \right)^2 + \mu \sum_{i,j=1}^N (\mathbf{A}_{ij} - \mathbf{F}_{ij})^2, \quad (1)$$

where  $\mu > 0$  is a regularization parameter.  $\mathbf{F} \in \mathbb{R}^{N \times N}$  represents the initial affinity values of  $\mathbf{W}$ .  $\mathbf{D}$  is a diagonal matrix with elements  $\mathbf{D}_{ii} = \sum_{j=1}^N \mathbf{W}_{ij}$ .

As demonstrated in (Bai et al. 2017), the closed-form solution of the objective function in Eq. (1) is obtained as:

$$\mathbf{A}^* = (1 - \alpha) \text{vec}^{-1}((\mathbf{I} - \alpha \mathbf{S})^{-1} \text{vec}(\mathbf{F})), \quad (2)$$

where  $\alpha = \frac{1}{1+\mu}$ .  $\text{vec}(\cdot)$  is an operator which vectorizes an input matrix by stacking its columns one after the next. The corresponding inverse operator is denoted as  $\text{vec}(\cdot)^{-1}$ .  $\mathbf{S} \in \mathbb{R}^{N^2 \times N^2}$  is the Kronecker product of  $\mathbf{S}$ , i.e.,  $\mathbf{S} \otimes \mathbf{S}$  with  $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$ .

Motivated by the iterative manner of previous works (Donoser and Bischof 2013), the iteratively similarity propagation can be formulated as follows:

$$\mathbf{A}^{(t+1)} = \alpha \mathbf{S} \mathbf{A}^{(t)} \mathbf{S}^T + (1 - \alpha) \mathbf{F}, \quad (3)$$

where  $t$  represents the  $t$ -th iteration time.

Although Eq. (3) can obtain good and stable results, it updates the single graph corresponding to each view separately, the complementary information of multiple graphs has not been fully utilized. In our CGD, we design a cross graph diffusion strategy to improve each single graph by using the complementarity among multiple graphs.

### Formulation of CGD

The critical point for improving multi-view clustering is how to exploit the complementarity among different views. Most of previous works build a certain data representation model from original multi-view features to learn a consensus similarity graph, their clustering performance differ with each other due to the representation capability of the learning models. In addition, solving these resultant learning models usually results in high computational complexity. Instead of learning the unified graph from original features, we attempt to generate the final consensus graph from multiple predefined graphs from different views, which is more intuitive and efficient. For the similarity graph matrix  $\mathbf{W}_v$  of the  $v$ -th view, we design the cross-view diffusion process to update  $\mathbf{W}_v$  as follows:

$$\mathbf{W}_v^{(t+1)} = \alpha \mathbf{S}_v^t \cdot \left( \frac{1}{V-1} \sum_{\substack{u=1 \\ u \neq v}}^V \mathbf{W}_u^{(t)} \right) \cdot (\mathbf{S}_v^t)^T + (1 - \alpha) \mathbf{W}_v^0. \quad (4)$$

As can be seen from Eq. (4), the connection information in different graphs has been interchanged iteratively to achieve the final unified graph. On one hand, similarity values of

different graph matrices are propagated to other ones in each iteration. On the other hand, the information from original graph are partially preserved by the scalar parameter  $\alpha$ . In such a manner, the complementarity among multiple graphs can be utilized to improve the diffusion process.

Note that we aim to fuse multiple similarity graph matrices, we need to apply a normalization step to all graph matrices to ensure them to be the same scale. For  $\mathbf{W}_v$ , one can use a simple way to perform the normalization as  $\bar{\mathbf{W}}_v = \mathbf{W}_v / \mathbf{D}_v$ , where  $\mathbf{D}_v$  is a diagonal matrix with  $\mathbf{D}_v(i, i) = \sum_{j=1}^N \mathbf{W}_v(i, j)$ . In such a manner,  $\sum_{j=1}^N \bar{\mathbf{W}}_v(i, j) = 1$  can be ensured. However, since self-similarities in the diagonal entries of  $\mathbf{W}_v$  are used, this kind of normalization may not be numerical stable (Tong et al. 2016). Here, we perform the normalization over the similarity graph matrix  $\mathbf{W}_v$  as follows:

$$\bar{\mathbf{W}}_v(i, j) = \begin{cases} \frac{\mathbf{W}_v(i, j)}{2 \sum_{j=1}^N \mathbf{W}_v(i, j)}, & i \neq j \\ \frac{1}{2}, & i = j. \end{cases} \quad (5)$$

By this way, we set the self-similarity in each row to  $1/2$  and the sum of rest elements to  $1/2$  so that the sum of the whole row  $\sum_{j=1}^N \bar{\mathbf{W}}_v(i, j) = 1$  still holds.

After  $T$  times of iteration, the final unified graph can be obtained by

$$\bar{\mathbf{W}} = \frac{1}{N} \sum_{v=1}^V \mathbf{W}_v^{(T)}, \quad (6)$$

which is used for final clustering by performing Normalized cuts (Shi and Malik 2000).

In Eq. (4), we need to set the value of  $\alpha$ . Since  $\alpha$  determines the proportion of the information from original graph that should be preserved at each iteration time, we design an adaptive manner to determine  $\alpha$  for each dataset. For different view-specific graphs, if the value of an edge is large in all of different graphs, we call this edge as strong edge and it should be obviously preserved in each graph during the diffusion process. Therefore, the more strong edges in a graph, the more information of original graph should be preserved. To this end, we can use the Hadamard product  $\mathbf{H}$  of different graphs to calculate the number of strong edges as follows:

$$\mathbf{H} = \prod_{v=1}^V \mathbf{W}_v, \quad (7)$$

where  $\prod$  denotes the Hadamard product of a sequence. By using Eq. (7),  $\mathbf{H}_{ij}$  will be large if  $\mathbf{W}_v(i, j)$  has large value in all of the different view-specific graphs. Otherwise,  $\mathbf{H}_{ij}$  will be close to 0. If the number of non-zero edges in  $\mathbf{H}$  is  $\tilde{N}$ , we set  $\alpha = 1 - \tilde{N}/N^2$ , i.e., larger  $\tilde{N}$  will induce small  $\alpha$  and more strong edges in original graph can be preserved.

### Theoretical Analysis

In this subsection, we give the computational complexity and theoretical convergence analysis of our CGD model.

**Computational Complexity** As can be seen from Eq. (4), after we obtain the predefined similarity graph matrices of



different views, the major computational cost of each diffusion iteration lies in the matrix multiplication operation with  $N \times N$  size, of which the computational time complexity is  $\mathcal{O}(N^3)$ . Therefore, the whole computational time complexity of our CGD is  $\mathcal{O}(TN^3)$ , where  $T$  is the final iteration times.

**Theoretical Convergence Guarantee** Similar to (Bai et al. 2017), the theoretical convergence of the iteration process of our CGD can be also guaranteed. Before we present the proof, we first give Lemma 1.

**Lemma 1.** *Given three matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  with appropriate sizes, then  $\text{vec}(\mathbf{ABC}^T) = (\mathbf{C} \otimes \mathbf{A})\text{vec}(\mathbf{B})$ .*

By applying  $\text{vec}(\cdot)$  to both sides of Eq. (4), we have

$$\begin{aligned} \text{vec}(\mathbf{W}_v^{(t+1)}) &= \alpha \mathbb{S} \text{vec}(\mathbf{Z}_v^{(t)}) + (1 - \alpha) \text{vec}(\mathbf{W}_v^{(1)}) \\ &= (\alpha \mathbb{S})^t \text{vec}(\mathbf{Z}_v^{(1)}) + (1 - \alpha) \sum_{k=0}^{t-1} (\alpha \mathbb{S})^k \text{vec}(\mathbf{W}_v^{(1)}) \end{aligned} \quad (8)$$

where  $\mathbf{Z}_v^{(t)} = \frac{1}{\sqrt{V-1}} \sum_{u=1, u \neq v}^V \mathbf{W}_u^{(t)}$ .

Since the spectral radius of  $\mathbf{S}$  is no larger than 1, according to the properties of Kronecker product, the eigenvalues of  $\mathbb{S} = \mathbf{S} \otimes \mathbf{S}$  are also in  $[-1, 1]$ . Since  $0 < \alpha < 1$ , we have

$$\begin{cases} \lim_{t \rightarrow \infty} (\alpha \mathbb{S})^t \text{vec}(\mathbf{Z}_v^{(1)}) = 0 \\ \lim_{t \rightarrow \infty} \sum_{k=0}^{t-1} (\alpha \mathbb{S})^k \text{vec}(\mathbf{W}_v^{(1)}) = (\mathbf{I} - \alpha \mathbb{S})^{-1} \text{vec}(\mathbf{W}_v^{(1)}) \end{cases} \quad (9)$$

Therefore, when  $t \rightarrow \infty$ , Eq. (8) converges as

$$\lim_{t \rightarrow \infty} \text{vec}(\mathbf{W}_v^{(t+1)}) = (1 - \alpha)(\mathbf{I} - \alpha \mathbb{S})^{-1} \text{vec}(\mathbf{W}_v^{(1)}). \quad (10)$$

By applying  $\text{vec}^{-1}$  to both sides of Eq. (10), the iteration converges to exactly the same solution in Eq. (2).

## Experiments

In this section, we present the experimental results of CGD on ten datasets (including two toy datasets and eight real-world datasets) and compare CGD with some with state-of-the-art methods.

### Datasets

**Toy Datasets** In order to give a intuitive visual illustration of the capability of CGD, we generated two toy datasets for testing. The first toy dataset consists of two views and 200 sample points, named RandomGaussian dataset, which is shown in Fig. 2a and Fig. 2f. Each view is generated from multivariate normal distribution with  $2 \times 2$  covariance matrix  $\begin{bmatrix} 0.1 & 0 \\ 0 & 5 \end{bmatrix}$ . The first view with 100 point are generated with mean vector  $[0, 1]$ , and the second view with 100 point are generated with mean vector  $[-1, 0]$ . The points in the left part (yellow) and right part (green) are two clusters, respectively. The second toy dataset named TwoMoon also consists of two views and 200 sample points, but with a moon pattern, as shown in Fig. 3a and Fig. 3f. Each view is generated

with 0.15 percentage of random Gaussian noise adding. The upper moon (yellow) and the lower moon (green) are two clusters, respectively. Each cluster has 100 sample points.

**Real-World Datasets** Six real-world benchmark datasets are used to evaluate the performance of our CGD. They are as follows:

**BBCSport** consists of documents of sport news corresponding to 5 topics, where for each document two different types of features are extracted (Xia et al. 2014); **MSRCV1** consists of 210 images from 7 classes. There are 6 types of features extracted: CENT, CMT, GIST, HOG, LBP, and SIFT (Xu, Han, and Nie 2016); **100leaves** consists of 1600 samples from each of one hundred plant species. For each sample, shape descriptor, fine scale margin and texture histogram are extracted<sup>1</sup>; **3sources** consists of 169 news, which were reported by three news organizations, i.e., BBC, Reuters, and The Guardian<sup>2</sup>; **Scene-15** consists of 15 scene categories with both indoor and outdoor environments, 4485 images in total. Three features including GIST, PHOG, and LBP are extracted (Li and Perona 2005); **Reuters** consists of 18758 samples with 5 types of languages and the documents are represented as a bag of words using a TFIDF-based weighting scheme (Amini, Usunier, and Goutte 2009). The detailed information of the datasets are summarized in Table 1, in which  $N$ ,  $V$ ,  $C$  and  $d_v$  denote the number of instances, views, clusters, and the dimension of features in the  $v$ -th view, respectively.

Table 1: Details of the benchmark datasets

| Datasets  | $N$   | $V$ | $C$ | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ |
|-----------|-------|-----|-----|-------|-------|-------|-------|-------|-------|
| BBCSport  | 544   | 2   | 5   | 3183  | 3203  | -     | -     | -     | -     |
| 100leaves | 1600  | 3   | 100 | 64    | 64    | 64    | -     | -     | -     |
| 3sources  | 169   | 3   | 6   | 3560  | 3631  | 3068  | -     | -     | -     |
| Scene-15  | 4485  | 3   | 15  | 20    | 59    | 40    | -     | -     | -     |
| Reuters   | 18758 | 5   | 6   | 21531 | 24892 | 34251 | 15506 | 11547 | -     |
| MSRCV1    | 210   | 6   | 7   | 1302  | 48    | 512   | 100   | 256   | 210   |

**Compared Baselines** We compare our method with the following ten baselines.

**SPC<sub>BestSV</sub>**: The standard spectral clustering (Ng, Jordan, and Weiss 2001). We report the best results corresponding to a certain single view. **LRR<sub>BestSV</sub>**: The standard LRR (Liu et al. 2013). We also report the results of the best single view. **DiMSC** (Cao et al. 2015): The method which uses the HSIC criterion to enforce the diversity of different views (Gretton et al. 2005), and then the clustering results are obtained by using spectral clustering. **AMGL** (Nie, Li, and Li 2016): A novel framework that automatically learns the affinity graphs and weights for different views, it can perform both multi-view clustering as well as other semi-supervised tasks. **MLAN** (Nie, Cai, and Li 2017): An efficient model for both clustering and semi-supervised classification, in which the local structure of data is learned adaptively. **ECMSC** (Wang et al. 2017): A novel multi-view clustering algorithm in which the data representation exclusivity and indicator con-

<sup>1</sup><https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set>

<sup>2</sup><http://mlg.ucd.ie/datasets/3sources.html>

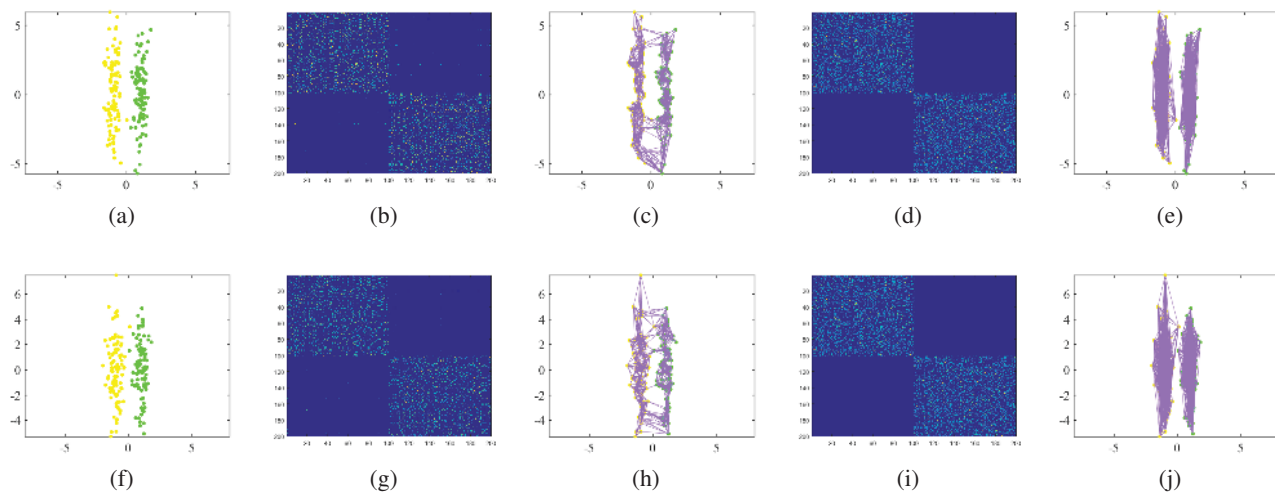


Figure 2: Intuitive results on the RandomGaussian dataset. The first row is the first view. The second row is the second view. (a) and (f) are the data points of the first view and the second view, respectively. (b) and (g) are the initial similarity graphs calculated by using the Gaussian kernel function for the first view and the second view, respectively. (d) and (i) are the learned similarity graphs after the diffusion process for the first view and the second view, respectively. (c) and (h) are the graph node connections of the initial graphs for the first view and the second view, respectively. (e) and (j) are the graph node connections of the learned graphs for the first view and the second view, respectively.

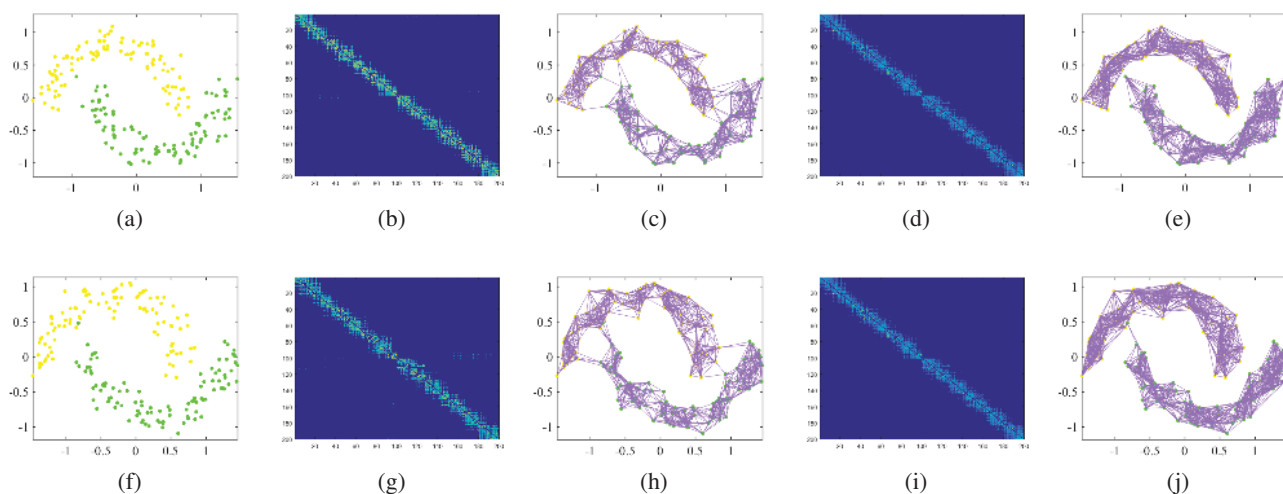


Figure 3: Intuitive results on the TwoMoon dataset. The notes are same to Fig. 2.

Table 2: Clustering performance comparison of different methods (mean  $\pm$  standard deviation)

| Datasets  | Methods            | NMI                   | ACC                | ARI                | F-score            | Precision          | Recall             | Purity      |             |
|-----------|--------------------|-----------------------|--------------------|--------------------|--------------------|--------------------|--------------------|-------------|-------------|
| BBCSport  | Single View        | SPC <sub>BestSV</sub> | 0.022±0.005        | 0.360±0.028        | 0.005±0.003        | 0.386±0.002        | 0.241±0.002        | 0.950±0.002 | 0.362±0.001 |
|           |                    | LRR <sub>BestSV</sub> | 0.775±0.002        | 0.904±0.002        | 0.747±0.001        | 0.812±0.002        | 0.754±0.001        | 0.880±0.002 | 0.904±0.001 |
|           | Multi-view         | DiMSC                 | 0.905±0.001        | 0.838±0.01         | 0.916±0.003        | 0.921±0.001        | 0.910±0.001        | 0.960±0.001 | 0.890±0.002 |
|           |                    | AMGL                  | 0.030±0.000        | 0.364±0.000        | 0.011±0.000        | 0.381±0.000        | 0.243±0.000        | 0.887±0.000 | 0.794±0.000 |
|           |                    | MLAN                  | 0.906±0.000        | 0.971±0.000        | 0.927±0.000        | 0.941±0.000        | 0.938±0.000        | 0.951±0.000 | 0.969±0.000 |
|           |                    | ECMSC                 | 0.065±0.004        | 0.393±0.006        | 0.020±0.004        | 0.376±0.002        | 0.246±0.004        | 0.794±0.002 | 0.400±0.002 |
|           |                    | LMSC                  | 0.842±0.006        | 0.923±0.007        | 0.875±0.002        | 0.905±0.008        | 0.899±0.004        | 0.911±0.006 | 0.923±0.005 |
|           |                    | MVGL                  | 0.097±0.000        | 0.412±0.000        | 0.040±0.000        | 0.398±0.000        | 0.255±0.000        | 0.911±0.000 | 0.432±0.000 |
|           |                    | MCGC                  | 0.112±0.000        | 0.421±0.000        | 0.049±0.000        | 0.401±0.000        | 0.258±0.000        | 0.899±0.000 | 0.444±0.000 |
|           |                    | GMC                   | 0.705±0.000        | 0.739±0.000        | 0.601±0.000        | 0.721±0.000        | 0.573±0.000        | 0.971±0.000 | 0.763±0.000 |
| CGD       | <b>0.910±0.003</b> | <b>0.974±0.004</b>    | <b>0.931±0.002</b> | <b>0.947±0.001</b> | <b>0.943±0.003</b> | <b>0.952±0.004</b> | <b>0.974±0.002</b> |             |             |
| MSRCV1    | Single View        | SPC <sub>BestSV</sub> | 0.556±0.000        | 0.519±0.000        | 0.289±0.000        | 0.431±0.000        | 0.300±0.000        | 0.767±0.000 | 0.523±0.000 |
|           |                    | LRR <sub>BestSV</sub> | 0.539±0.021        | 0.681±0.018        | 0.413±0.019        | 0.498±0.017        | 0.476±0.019        | 0.521±0.018 | 0.681±0.018 |
|           | Multi-view         | DiMSC                 | 0.661±0.018        | 0.751±0.021        | 0.743±0.016        | 0.731±0.016        | 0.755±0.015        | 0.827±0.019 | 0.701±0.016 |
|           |                    | AMGL                  | 0.743±0.000        | 0.738±0.000        | 0.618±0.000        | 0.675±0.000        | 0.626±0.000        | 0.732±0.000 | 0.728±0.000 |
|           |                    | MLAN                  | 0.753±0.000        | 0.719±0.000        | 0.619±0.000        | 0.678±0.000        | 0.601±0.000        | 0.778±0.000 | 0.785±0.000 |
|           |                    | ECMSC                 | 0.831±0.003        | 0.908±0.006        | 0.762±0.003        | 0.801±0.002        | 0.800±0.004        | 0.824±0.002 | 0.908±0.001 |
|           |                    | LMSC                  | 0.718±0.013        | 0.838±0.015        | 0.667±0.016        | 0.714±0.015        | 0.703±0.014        | 0.725±0.017 | 0.838±0.018 |
|           |                    | MVGL                  | 0.833±0.000        | 0.904±0.000        | 0.774±0.000        | 0.802±0.000        | 0.801±0.000        | 0.823±0.000 | 0.904±0.000 |
|           |                    | MCGC                  | 0.692±0.000        | 0.776±0.000        | 0.630±0.000        | 0.685±0.000        | 0.640±0.000        | 0.737±0.000 | 0.785±0.000 |
|           |                    | GMC                   | 0.816±0.000        | 0.895±0.000        | 0.767±0.000        | 0.799±0.000        | 0.786±0.000        | 0.814±0.000 | 0.895±0.000 |
| CGD       | <b>0.842±0.004</b> | <b>0.910±0.006</b>    | <b>0.790±0.003</b> | <b>0.819±0.004</b> | <b>0.804±0.005</b> | <b>0.836±0.005</b> | <b>0.910±0.005</b> |             |             |
| 100leaves | Single View        | SPC <sub>BestSV</sub> | 0.777±0.002        | 0.483±0.014        | 0.203±0.008        | 0.215±0.008        | 0.128±0.007        | 0.674±0.006 | 0.520±0.003 |
|           |                    | LRR <sub>BestSV</sub> | 0.715±0.018        | 0.488±0.013        | 0.307±0.011        | 0.315±0.010        | 0.274±0.007        | 0.371±0.008 | 0.529±0.009 |
|           | Multi-view         | DiMSC                 | 0.713±0.003        | 0.857±0.004        | 0.611±0.002        | 0.580±0.005        | 0.647±0.001        | 0.882±0.002 | 0.607±0.003 |
|           |                    | AMGL                  | 0.889±0.000        | 0.775±0.000        | 0.563±0.000        | 0.568±0.000        | 0.431±0.000        | 0.834±0.000 | 0.815±0.000 |
|           |                    | MLAN                  | 0.941±0.000        | 0.848±0.000        | 0.797±0.000        | 0.799±0.000        | 0.737±0.000        | 0.871±0.000 | 0.876±0.000 |
|           |                    | ECMSC                 | 0.823±0.013        | 0.670±0.012        | 0.516±0.013        | 0.521±0.011        | 0.459±0.007        | 0.602±0.006 | 0.686±0.008 |
|           |                    | LMSC                  | 0.869±0.009        | 0.738±0.017        | 0.645±0.012        | 0.649±0.013        | 0.608±0.011        | 0.695±0.012 | 0.758±0.009 |
|           |                    | MVGL                  | 0.869±0.000        | 0.766±0.000        | 0.506±0.000        | 0.513±0.000        | 0.379±0.000        | 0.789±0.000 | 0.787±0.000 |
|           |                    | MCGC                  | 0.834±0.000        | 0.727±0.000        | 0.410±0.000        | 0.418±0.000        | 0.290±0.000        | 0.745±0.000 | 0.747±0.000 |
|           |                    | GMC                   | 0.902±0.000        | 0.824±0.000        | 0.497±0.000        | 0.504±0.000        | 0.352±0.000        | 0.881±0.000 | 0.851±0.000 |
| CGD       | <b>0.943±0.007</b> | <b>0.859±0.005</b>    | <b>0.821±0.006</b> | <b>0.823±0.004</b> | <b>0.770±0.006</b> | <b>0.884±0.004</b> | <b>0.881±0.005</b> |             |             |
| 3sources  | Single View        | SPC <sub>BestSV</sub> | 0.054±0.014        | 0.331±0.015        | 0.011±0.012        | 0.362±0.011        | 0.228±0.008        | 0.879±0.009 | 0.349±0.013 |
|           |                    | LRR <sub>BestSV</sub> | 0.525±0.016        | 0.627±0.009        | 0.351±0.011        | 0.555±0.012        | 0.411±0.013        | 0.853±0.013 | 0.668±0.008 |
|           | Multi-view         | DiMSC                 | 0.684±0.011        | 0.749±0.007        | 0.608±0.005        | 0.703±0.007        | 0.644±0.006        | 0.882±0.008 | 0.680±0.003 |
|           |                    | AMGL                  | 0.058±0.000        | 0.325±0.000        | 0.026±0.000        | 0.348±0.000        | 0.221±0.000        | 0.812±0.000 | 0.384±0.000 |
|           |                    | MLAN                  | 0.548±0.000        | 0.680±0.000        | 0.365±0.000        | 0.553±0.000        | 0.434±0.000        | 0.762±0.000 | 0.716±0.000 |
|           |                    | ECMSC                 | 0.095±0.006        | 0.378±0.007        | 0.035±0.008        | 0.331±0.006        | 0.250±0.005        | 0.489±0.008 | 0.443±0.006 |
|           |                    | LMSC                  | 0.671±0.012        | 0.721±0.009        | 0.567±0.005        | 0.656±0.006        | 0.744±0.007        | 0.589±0.004 | 0.816±0.008 |
|           |                    | MVGL                  | 0.070±0.000        | 0.302±0.000        | 0.036±0.000        | 0.339±0.000        | 0.218±0.000        | 0.768±0.000 | 0.379±0.000 |
|           |                    | MCGC                  | 0.075±0.000        | 0.301±0.000        | 0.037±0.000        | 0.337±0.000        | 0.216±0.000        | 0.756±0.000 | 0.384±0.000 |
|           |                    | GMC                   | 0.548±0.000        | 0.692±0.000        | 0.443±0.000        | 0.605±0.000        | 0.484±0.000        | 0.804±0.000 | 0.746±0.000 |
| CGD       | <b>0.695±0.005</b> | <b>0.781±0.006</b>    | <b>0.611±0.005</b> | <b>0.709±0.006</b> | <b>0.651±0.007</b> | <b>0.889±0.006</b> | <b>0.828±0.003</b> |             |             |
| scene-15  | Single View        | SPC <sub>BestSV</sub> | 0.384±0.014        | 0.377±0.013        | 0.208±0.001        | 0.272±0.014        | 0.234±0.014        | 0.324±0.013 | 0.404±0.014 |
|           |                    | LRR <sub>BestSV</sub> | 0.369±0.002        | 0.368±0.003        | 0.201±0.001        | 0.263±0.002        | 0.233±0.003        | 0.302±0.001 | 0.395±0.001 |
|           | Multi-view         | DiMSC                 | 0.138±0.003        | 0.204±0.002        | 0.059±0.001        | 0.125±0.002        | 0.123±0.003        | 0.126±0.001 | 0.203±0.002 |
|           |                    | AMGL                  | 0.369±0.012        | 0.339±0.011        | 0.166±0.009        | 0.243±0.010        | 0.183±0.008        | 0.362±0.007 | 0.368±0.012 |
|           |                    | MLAN                  | 0.177±0.001        | 0.157±0.002        | 0.015±0.001        | 0.14±0.001         | 0.076±0.001        | 0.325±0.002 | 0.171±0.001 |
|           |                    | ECMSC                 | 0.297±0.011        | 0.307±0.012        | 0.154±0.011        | 0.214±0.011        | 0.209±0.009        | 0.218±0.008 | 0.380±0.007 |
|           |                    | LMSC                  | 0.353±0.010        | 0.413±0.012        | 0.212±0.009        | 0.270±0.007        | 0.267±0.008        | 0.291±0.008 | 0.436±0.007 |
|           |                    | MVGL                  | 0.378±0.000        | 0.369±0.000        | 0.216±0.000        | 0.283±0.000        | 0.226±0.000        | 0.329±0.000 | 0.398±0.000 |
|           |                    | MCGC                  | 0.142±0.000        | 0.179±0.000        | 0.054±0.000        | 0.170±0.000        | 0.096±0.000        | 0.329±0.000 | 0.186±0.000 |
|           |                    | GMC                   | 0.058±0.000        | 0.140±0.000        | 0.004±0.000        | 0.132±0.000        | 0.071±0.000        | 0.354±0.000 | 0.146±0.000 |
| CGD       | <b>0.419±0.006</b> | <b>0.428±0.004</b>    | <b>0.256±0.003</b> | <b>0.315±0.003</b> | <b>0.277±0.002</b> | <b>0.364±0.003</b> | <b>0.484±0.004</b> |             |             |
| Reuters   | Single View        | SPC <sub>BestSV</sub> | 0.112±0.012        | 0.296±0.008        | 0.059±0.000        | 0.378±0.007        | 0.238±0.009        | 0.912±0.006 | 0.329±0.007 |
|           |                    | LRR <sub>BestSV</sub> | 0.206±0.006        | 0.397±0.003        | 0.064±0.005        | 0.324±0.004        | 0.240±0.005        | 0.899±0.004 | 0.294±0.005 |
|           | Multi-view         | DiMSC                 | 0.182±0.003        | 0.401±0.011        | 0.071±0.007        | 0.285±0.004        | 0.255±0.008        | 0.901±0.006 | 0.338±0.007 |
|           |                    | AMGL                  | 0.194±0.001        | 0.361±0.003        | 0.059±0.002        | 0.291±0.003        | 0.242±0.004        | 0.893±0.002 | 0.325±0.002 |
|           |                    | MLAN                  | 0.225±0.002        | 0.359±0.003        | 0.063±0.002        | 0.323±0.003        | 0.218±0.002        | 0.890±0.003 | 0.331±0.002 |
|           |                    | ECMSC                 | 0.267±0.004        | 0.384±0.006        | 0.065±0.004        | 0.301±0.003        | 0.239±0.005        | 0.873±0.003 | 0.314±0.003 |
|           |                    | LMSC                  | 0.278±0.006        | 0.479±0.006        | 0.077±0.005        | 0.401±0.005        | 0.263±0.006        | 0.905±0.007 | 0.347±0.005 |
|           |                    | MVGL                  | 0.271±0.000        | 0.461±0.000        | 0.075±0.000        | 0.394±0.000        | 0.243±0.000        | 0.898±0.000 | 0.352±0.000 |
|           |                    | MCGC                  | 0.263±0.000        | 0.439±0.000        | 0.072±0.000        | 0.388±0.000        | 0.257±0.000        | 0.879±0.000 | 0.349±0.000 |
|           |                    | GMC                   | 0.274±0.000        | 0.472±0.000        | 0.078±0.000        | 0.391±0.000        | 0.262±0.000        | 0.910±0.000 | 0.351±0.000 |
| CGD       | <b>0.287±0.005</b> | <b>0.492±0.004</b>    | <b>0.082±0.003</b> | <b>0.422±0.003</b> | <b>0.279±0.003</b> | <b>0.923±0.002</b> | <b>0.367±0.003</b> |             |             |

sistency are simultaneously exploited in a unified manner. **LMSC** (Zhang et al. 2017): The latent multi-view subspace clustering method, which clusters data points with latent representation and simultaneously explores underlying complementary information from multiple views. **MCGL** (Zhan et al. 2018): The graph learning for multi-view clustering, in which a rank constraint is imposed on the Laplacian matrix to regularize the number of connected elements of the learned graph. **MCGC** (Zhan et al. 2019): Multi-view consensus graph clustering, which learn a consensus graph with minimizing disagreement between different views and constraining the rank of the Laplacian matrix. **GMC** (Wang, Yang, and Liu 2019): Graph-based multi-view clustering, which takes the data graph matrices of all views and fuses them to generate a unified graph matrix.

**Experiment Settings** There are different parameters in other compared methods. For fair comparison, we tune their parameters as suggested in the original paper and the corresponding best results from the optimal parameters for all methods are reported. Without loss of generality, we use the commonly used Gaussian kernel function with Euclidean distance to generate initial view-specific graphs.  $\sigma$  in the Gaussian kernel function is set to 0.5. Seven widely used metrics are used to evaluate the performance: clustering accuracy (ACC), normalized mutual information (NMI), purity, precision, recall, F-score, and adjusted rand index (ARI). For these metrics, the larger value indicates the better clustering performance. We run each algorithm 10 times with the optimal parameters and report the means and standard deviations of the performance measures.

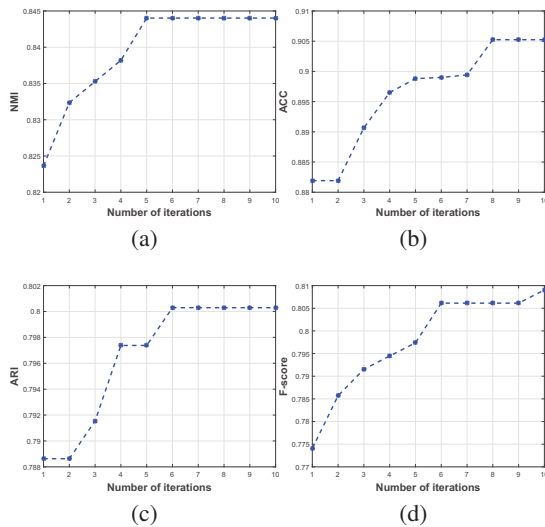


Figure 4: Clustering performance with different iteration times of MSRCV1 dataset.

**Experimental Results on Toy Datasets** Fig. 2b, Fig. 2g, Fig. 3b and Fig. 3g show the constructed graphs of the two views of the two toy datasets. Fig. 2d, Fig. 2i, Fig. 3d and Fig. 3i are the corresponding updated graphs of the two

views of the two toy datasets after  $T$  times of iteration. As can be seen, after the diffusion process, some noisy similarity values are removed, while some credible similarity values can be strengthened, which indicates that the diffusion process can effectively improve the initial similarity graphs. In order Fig. 2e, Fig. 2j, Fig. 3e and Fig. 3j show the connections by using the learned graphs. As shown, the learned similarity graphs can separate different views very well since it can exploit the complementary information from different views.

**Experimental Results on Real-World Datasets** The quantitative comparison results in terms of different metric are shown in Table 2. From the results, we can have the following observations:

- In most cases, multi-view learning methods can obtain better results than single view methods, which demonstrate that the cross-view information can be used to effectively improve the clustering performance for multi-view data;
- Our proposed CGD performs better than all of other state-of-the-art ones, which validate the efficacy of the cross-view graph diffusion process, i.e., the complementary information implied in the graphs of different views can be exploited to boost each other for the final clustering task;
- In some cases, single view methods, i.e.,  $\mathbf{SPC}_{\text{BestSV}}$  and  $\mathbf{LRR}_{\text{BestSV}}$ , are even work better than some multi-view methods. This indicates that exploring stable multi-view clustering method is still necessary.

**Clustering Results with Different Iteration Times** In order to validate that the diffusion process can improve the quality of the similarity graphs step by step, we plot the clustering results of MSRCV1 dataset with different iteration times in Fig. 4. As can be seen, the clustering results have been improved obviously in the first iteration times, and stay stable after several times of iteration.

## Conclusions

In this work, we propose a general, effective and parameter-free method with convergence guarantee for multi-view data clustering via cross-view graph diffusion (CGD). CGD takes the traditional predefined graph matrices of different views as input, and learns an improved graph for each single view via an iterative cross diffusion process. The final unified graph used for clustering is obtained by averaging the improved view associated graphs. Extensive experiments on several benchmark datasets are conducted to demonstrate the efficacy of the proposed method in terms of seven clustering evaluation metrics. In the future, we plan to adopt the Nystrom sampling to improve the computational complexity.

## Acknowledgments

This work was supported in part by NSFC (NO. 61701451, 61773392, 61872371 and 61922088), and in part by the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) under Grant



CUG170654. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for computation acceleration for this research. Xin-wang Liu is the corresponding author of this paper.

## References

- Amini, M.; Usunier, N.; and Goutte, C. 2009. Learning from multiple partially observed views - an application to multilingual text categorization. In *NIPS*, 28–36.
- Bai, S.; Bai, X.; Tian, Q.; and Latecki, L. J. 2017. Regularized diffusion process for visual retrieval. In *AAAI*, 3967–3973.
- Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *CVPR*, 586–594.
- Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*, 886–893.
- Donoser, M., and Bischof, H. 2013. Diffusion processes for retrieval revisited. In *CVPR*, 1320–1327.
- Gretton, A.; Bousquet, O.; Smola, A.; and Schölkopf, B. 2005. Measuring statistical dependence with hilbert-schmidt norms. In *ICALT*, 63–77.
- Li, L., and Cai, M. 2017. Drug target prediction by multi-view low rank embedding. *IEEE/ACM TCBB* PP(99):1–1.
- Li, F. F., and Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 524–531.
- Li, L. 2014. Mprgraph: multi-view penalised graph clustering for predicting drug-target interactions. *Iet Systems Biology* 8(2):67–73.
- Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI* 35(1):171–184.
- Liu, X.; Zhu, X.; Li, M.; Wang, L.; Tang, C.; Yin, J.; Shen, D.; Wang, H.; and Gao, W. 2018. Late fusion incomplete multi-view clustering. *IEEE TPAMI*.
- Lowe, D. G., and Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60(2):91–110.
- Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: analysis and an algorithm. In *NIPS*, 849–856.
- Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*, 2408–2414.
- Nie, F.; Li, J.; and Li, X. 2016. Parameter-free auto-weighted multiple graph learning: a framework for multi-view clustering and semi-supervised classification. In *IJCAI*, 1881–1887.
- Ojala, T.; Pietikainen, M.; and Maenpaa, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI* 24(7):971–987.
- Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE TPAMI* 22(8):888–905.
- Tang, C.; Liu, X.; Wang, P.; Zhang, C.; Li, M.; and Wang, L. 2019a. Adaptive hypergraph embedded semi-supervised multi-label image annotation. *IEEE TMM*.
- Tang, C.; Liu, X.; Zhu, X.; Xiong, J.; Li, M.; Xia, J.; Wang, X.; and Wang, L. 2019b. Feature selective projection with low-rank embedding and dual laplacian regularization. *IEEE TKDE*.
- Tang, C.; Zhu, X.; Liu, X.; Li, M.; Wang, P.; Zhang, C.; and Wang, L. 2019c. Learning a joint affinity graph for multi-view subspace clustering. *IEEE TMM* 21(7):1724–1736.
- Tang, C.; Zhu, X.; Liu, X.; and Wang, L. 2019d. Cross-view local structure preserved diversity and consensus learning for multi-view unsupervised feature selection. In *AAAI*, 5101–5108.
- Tong, T.; Gray, K.; Gao, Q.; Liang, C.; and Rueckert, D. 2016. Multi-modal classification of alzheimer’s disease using nonlinear graph fusion. *Pattern Recognition* 63:171–181.
- Wang, Y., and Wu, L. 2018. Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. *Neural Networks* 103:1–8.
- Wang, X.; Guo, X.; Lei, Z.; Zhang, C.; and Li, S. Z. 2017. Exclusivity-consistency regularized multi-view subspace clustering. In *CVPR*, 923–931.
- Wang, H.; Nie, F.; and Huang, H. 2013. Multi-view clustering and feature learning via structured sparsity. In *ICML*, 352–360.
- Wang, H.; Yang, Y.; and Liu, B. 2019. Gmc: graph-based multi-view clustering. *IEEE TKDE*.
- Xia, R.; Pan, Y.; Du, L.; and Yin, J. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, 2149–2155.
- Xu, J.; Han, J.; and Nie, F. 2016. Discriminatively embedded k-means for multi-view clustering. In *CVPR*.
- Yang, Y.; Shen, F.; Huang, Z.; Shen, H. T.; and Li, X. 2017. Discrete nonnegative spectral clustering. *IEEE TKDE* 29(9):1834–1845.
- Yang, Z.; Xu, Q.; Zhang, W.; Cao, X.; and Huang, Q. 2019. Split multiplicative multi-view subspace clustering. *IEEE TIP* 28(10):5147–5160.
- Zhan, K.; Zhang, C.; Guan, J.; and Wang, J. 2018. Graph learning for multiview clustering. *IEEE TCyb* 48(10):2887–2895.
- Zhan, K.; Nie, F.; Wang, J.; and Yang, Y. 2019. Multiview consensus graph clustering. *IEEE TIP* 28(3):1261–1270.
- Zhang, C.; Fu, H.; Liu, S.; Liu, G.; and Cao, X. 2015. Low-rank tensor constrained multiview subspace clustering. In *ICCV*, 1582–1590.
- Zhang, C.; Hu, Q.; Fu, H.; Zhu, P.; and Cao, X. 2017. Latent multi-view subspace clustering. In *CVPR*, 4279–4287.
- Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2019. Generalized latent multi-view subspace clustering. *IEEE TPAMI*.
- Zhou, D.; Weston, J.; Gretton, A.; Bousquet, O.; and Schölkopf, B. 2004. Ranking on data manifolds. In *NIPS*, 169–176.