CDIMC-net: Cognitive Deep Incomplete Multi-view Clustering Network

Jie Wen[1,2†], **Zheng Zhang**[1,2,3†], **Yong Xu**[1,2,3*], **Bob Zhang**[4], **Lunke Fei**[5] and **Guo-Sen Xie**[6]
[1]Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen, China
[2]Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen, China
[3]Pengcheng Laboratory, Shenzhen, China
[4]Department of Computer and Information Science, University of Macau, Taipa, Macau, China
[5]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China
[6]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
jiewen_pr@126.com, darrenzz219@gmail.com, yongxu@ymail.com, bobzhang@um.edu.mo, flksxm@126.com, gsxiehm@gmail.com

Abstract

In recent years, incomplete multi-view clustering, which studies the challenging multi-view clustering problem on missing views, has received growing research interests. Although a series of methods have been proposed to address this issue, the following problems still exist: 1) Almost all of the existing methods are based on shallow models, which is difficult to obtain discriminative common representations. 2) These methods are generally sensitive to noise or outliers since the negative samples are treated equally as the important samples. In this paper, we propose a novel incomplete multi-view clustering network, called Cognitive Deep Incomplete Multi-view Clustering Network (CDIMC-net), to address these issues. Specifically, it captures the high-level features and local structure of each view by incorporating the view-specific deep encoders and graph embedding strategy into a framework. Moreover, based on the human cognition, *i.e.*, learning from easy to hard, it introduces a self-paced strategy to select the most confident samples for model training, which can reduce the negative influence of outliers. Experimental results on several incomplete datasets show that CDIMC-net outperforms the state-of-the-art incomplete multi-view clustering methods.

1 Introduction

Multi-view clustering is a well-known research topic in fields of machine learning [Chao *et al.*, 2017; Zhang *et al.*, 2018]. Generally speaking, almost all of the previous researches on multi-view clustering are based on the assumption that all views of samples are available and strictly aligned. However, in practical applications, more and more collected multi-view data are incomplete where some views are unavailable. For example, many volunteers only have one or two kinds of examination results of magnetic resonance imaging, positron emission tomography, and cerebrospinal fluid for Alzheimer's disease diagnosing [Xiang *et al.*, 2013]. Multi-view data with missing views are called *incomplete multi-view data* and clustering on such data is the so-called *incomplete multi-view clustering* (IMC) [Wen *et al.*, 2019; Hu and Chen, 2018]. Obviously, conventional methods fail to handle these incomplete multi-view data. In addition, owing to the missing views, it is difficult to explore the complementary and consistent information from the incomplete multi-view data, which makes IMC a very challenging task.

For IMC, Trivedi et al. [Trivedi *et al.*, 2010] proposed a kernel canonical correlation analysis (KCCA) based method, which recovers the kernel matrix of the incomplete view according to that of the complete view. However, it can only handle the two-view data and requires that one view is complete. To address this issue, Li et al. proposed the partial multi-view clustering (PMVC) based on matrix factorization, where the paired views are decomposed into the same representation [Li *et al.*, 2014]. Then various extensions of PMVC, such as graph regularized PMVC (GPMVC) [Rai *et al.*, 2016] and incomplete multi-modal grouping (IMG) [Zhao *et al.*, 2016] have been proposed, which mainly incorporate the graph embedding technique to enhance the separability of the common representation. In [Zhao *et al.*, 2018], the graph embedding and deep feature extraction techniques are simultaneously integrated into PMVC to preserve the local structure and capture the high-level features. Partial multi-view clustering via consistent GAN (PMVC_GAN) unifies the Autoencoder and cycle generative adversarial network (GAN) into a novel IMC framework, which can infer the missing views via GAN and in turn promotes the common representation learning [Wang *et al.*, 2018]. However, the above methods are not applicable to arbitrary incomplete cases where some incomplete samples have more than one views or no samples have complete views [Wen *et al.*, 2020]. To address the issue, weighted matrix factorization technique is introduced for IMC, where the representative works are multiple incomplete views clustering (MIC) [Shao *et al.*, 2015], doubly aligned IMC (DAIMC) [Hu and Chen, 2018], one-pass IMC (OPIMC) [Hu and Chen, 2019], and online multi-view clustering (OMVC) [Shao *et al.*, 2016]. Besides these meth-

*corresponding author, '†' indicates co-first authors.

ods, many graph learning based methods [Wang *et al.*, 2019; Wen *et al.*, 2020] and multiple kernels based methods (such as incomplete multiple kernel k-means with mutual kernel completion (MKKM-IK-MKC) [Liu *et al.*, 2019]) have also been proposed to handle the arbitrary IMC cases.

Although the aforementioned methods provide some schemes to address the IMC problem, these methods still suffer from the following issues: 1) most of the previous methods exploit the shallow models to obtain the common representation, which cannot capture the high-level features from the complex multi-view data. Although some deep network based methods like PMVC_GAN can capture the high-level features, it is inflexible to handle all kinds of incomplete cases. 2) None of them considers the negative effect of marginal samples since these methods treat all samples equally. This is unreasonable because those marginal samples are generally far away from the cluster centers and can be viewed as outliers, which are harmful to model training [Guo *et al.*, 2019]. In this paper, we propose a novel cognitive based deep incomplete multi-view clustering network, referred to as CDIMC-net, to address the above issues. CDIMC-net integrates several view-specific deep encoders, a self-paced kmeans clustering layer, and multiple graph constraints into a unified network for arbitrary IMC. Representatively, the main contributions of our work are illustrated as follows:

1) We propose a novel and flexible deep clustering network for arbitrary IMC cases, which incorporates the graph embedding to promote the network training.

2) This is the first work that introduces the human cognitive based learning into IMC. Compared with the existing works, CDIMC-net can adaptively reduce the negative influence of the marginal samples, and thus is more robust to outliers.

2 Kmeans Clustering

Kmeans is one of the most famous clustering algorithms. For any data $X = [x_1, \dots, x_n] \in R^{m\times n}$ with n samples and m features, kmeans seeks to find k optimal cluster centers $U = [u_1, \dots, u_k] \in R^{m\times k}$ and cluster indicator $S \in \{0,1\}^{k\times n}$ by solving the following problem [Nie *et al.*, 2019]:

$$\min_{U,S} \|X - US\|_F^2 \quad s.t.\ S \in \{0,1\}^{k\times n}, S^T \mathit{1} = \mathit{1} \qquad (1)$$

where $S_{j,i} = 1$ denotes that the corresponding i-th sample x_i is partitioned into the j-th cluster. $\mathit{1}$ is a vector with all elements as 1.

3 The Proposed Method

3.1 Problem Statement

For the given incomplete multi-view data with l views, we use $X^{(v)} = \left[x_1^{(v)}, \dots, x_n^{(v)}\right] \in R^{m_v\times n}$ to represent the instance set of the v-th view, where m_v is the feature dimension, n denotes the number of samples, and elements of the missing instances are denoted as 'NaN' (*i.e.*, not a number). The view available and missing information is recorded in a diagonal matrix $W^{(v)}$ for the v-th view, where $W_{i,i}^{(v)} = 1$ if the i-th instane is available in the v-th view, otherwise $W_{i,j}^{(v)} = 0$. The goal of IMC is to group these n samples into k clusters.

3.2 CDIMC-net

As shown in Fig.1, CDIMC-net groups the incomplete multi-view data via two phases: pre-training and fine-tuning, where an Autoencoder based pre-training phase is used to initialize the network parameters and the fine-tuning phase aims at obtaining the cluster-friendly representations while producing the cluster indicators for all input samples.

Pre-training Network

In our work, based on the conventional under-complete Autoencoder [Guo *et al.*, 2017; Xie *et al.*, 2016], we develop a graph regularized incomplete multi-view Autoencoder for incomplete multi-view cases, where the graph embedding technique is introduced to preserve the local structure of data and a weighted fusion layer is introduced to eliminate the negative influence of missing views. Specifically, the proposed incomplete multi-view Autoencoder is composed of the following three components.

View-specific encoders and decoders: As the basic Autoencoder, encoder network captures the most salient features from the high-dimensional data and the decoder network aims at recovering the data from the encoded features [Guo *et al.*, 2017]. Considering that different views may have different dimensions, information, structures, and appearances, we design several view-specific encoders $\left\{f_{EC}^{(v)}\right\}_{v=1}^{l}$ and corresponding decoders $\left\{f_{DC}^{(v)}\right\}_{v=1}^{l}$ for different views.

Fusion layer: As mentioned in many references, all views share the common semantic information for the same sample, such as the common representation or cluster label [Zhao *et al.*, 2018; Hu and Chen, 2019]. Inspired by this, CDIMC-net seeks to obtain the common representation shared by all views for clustering. Specifically, a simple weighted fusion layer is introduced for such goal as follows:

$$h_i^* = \sum_{v=1}^{l} W_{i,i}^{(v)} h_i^{(v)} \Bigg/ \sum_{v=1}^{l} W_{i,i}^{(v)} \qquad (2)$$

where $h_i^{(v)}$ is the output of instance $x_i^{(v)}$ at the v-th encoder $f_{EC}^{(v)}$. h_i^* is the common representation for the i-th sample.

As shown in Fig.1, the fusion layer is placed between the encoding network and the decoding network. Introducing the weighted fusion layer can solve the issue of incomplete learning by reducing the negative influence of missing views.

Graph embedding: In fields of subspace learning, a recognized manifold assumption is that if two data points x_i and x_j are close to each other, then their corresponding low dimensional representations should also be close in the latent subspace [Cai *et al.*, 2008; Kang *et al.*, 2017; Wen *et al.*, 2018a]. To preserve such neighbor relationships, the following graph embedding constraint is considered:

$$\min \frac{1}{2nl} \sum_{v=1}^{l} \sum_{i=1}^{n} \sum_{j=1}^{n} \left\| h_i^{(v)} - h_j^{(v)} \right\|_2^2 N_{i,j}^{(v)} \qquad (3)$$

where $N^{(v)} \in R^{n\times n}$ denotes the nearest neighbor graph con-

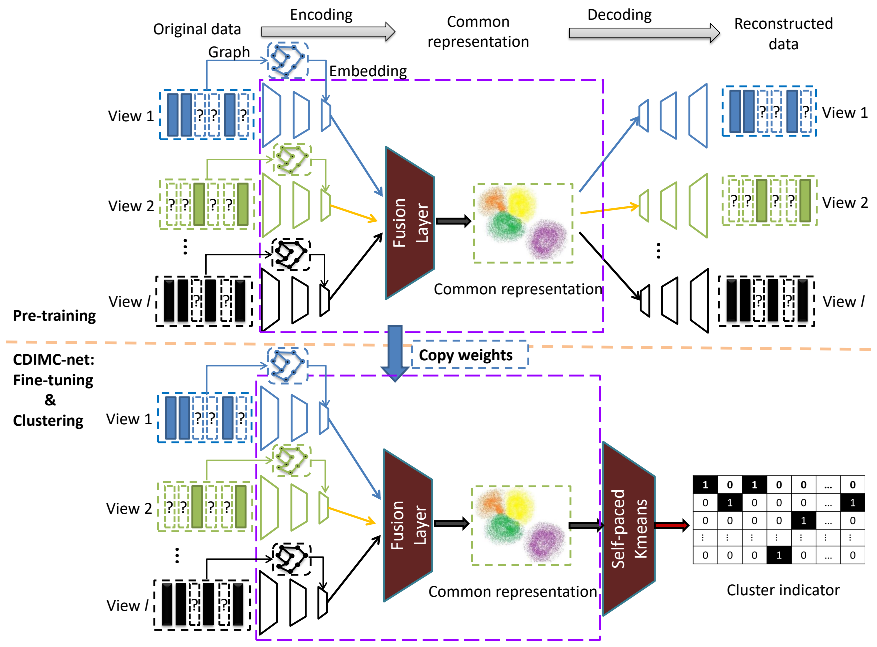


Figure 1: The proposed CDIMC-net for incomplete multi-view clustering.

structed from instance set $X^{(v)}$ as follows:

$$N_{i,j}^{(v)} = \begin{cases} 1, & \begin{array}{c} if\ (x_i^{(v)} \neq NaN, x_j^{(v)} \neq NaN) \& \\ (x_i^{(v)} \in \psi(x_j^{(v)})\ or\ x_j^{(v)} \in \psi(x_i^{(v)})) \end{array} \\ 0, & otherwise \end{cases} \tag{4}$$

where $\psi(x_i^{(v)})$ denotes the nearest instance set to $x_i^{(v)}$.

Loss function of pre-training: Combining the graph embedding loss and Autoencoder loss, the overall loss function of the pre-training network is designed as:

$$\begin{aligned} \min_{\Omega_1,\Omega_2} \sum_{v=1}^{l} \frac{1}{m_v n} \left\| \left(X^{(v)} - \bar{X}^{(v)} \right) W^{(v)} \right\|_F^2 \\ + \alpha \frac{1}{2nl} \sum_{v=1}^{l} \sum_{i=1}^{n} \sum_{j=1}^{n} \left\| h_i^{(v)} - h_j^{(v)} \right\|_2^2 N_{i,j}^{(v)} \end{aligned} \tag{5}$$

where $\bar{X}^{(v)} = f_{DC}^{(v)}\left(f_{EC}^{(v)}\left(X^{(v)}\right)\right)$ denotes the reconstructed data of the v-th view, Ω_1 and Ω_2 denote the parameters of encoder network and decoder network, respectively. α is a positive hyper-parameter.

Fine-tuning and Clustering

By optimizing problem (5), we can obtain a more compact common representation H^* for the incomplete multi-view data and the initialized network parameters. However, it cannot guarantee the obtained common representation to be cluster-friendly. In many previous deep clustering works, kmeans is widely considered [Yang *et al.*, 2017; Guo *et al.*, 2018]. However, as can be seen from objective function (1), the conventional kmeans treats all samples including the cluster-oriented samples and marginal samples (outliers) equally, which makes the trained clustering network be sensitive to outliers. A good approach to address this issue is to select the most cluster-oriented samples for model fine-tuning [Guo *et al.*, 2019]. To this end, we introduce a self-paced kmeans as the clustering layer for fine-tuning.

Self-paced kmeans: The loss function of the self-paced kmeans is denoted as follows [Guo *et al.*, 2019]:

$$\begin{aligned} &\min_{S,v,\lambda} \frac{1}{nk} \sum_{i=1}^{n} \left(r_i \left\| h_i^* - US_{:,i} \right\|_2^2 - \lambda r_i \right) \\ &s.t.\ r_i \in \{0,1\}, S \in \{0,1\}^{k \times n}, S^T \mathit{1} = \mathit{1} \end{aligned} \tag{6}$$

where U and S are the cluster center matrix and cluster indicator matrix as in (1). A difference between the self-paced kmeans and the conventional kmeans is that the cluster centers U is fixed in self-paced approach. This operation can avoid the trivial solution: all samples are trained into a same point in the latent space. $r = [r_1, \ldots, r_n] \in R^n$ is a weight vector, λ is an age parameter.

Generally, parameter λ needs to increase gradually such that more samples can be selected for network training. However, it is difficult to control its growth rate for different tasks. In our work, we adopt a statistic based adaptive approach following [Guo *et al.*, 2019] to update parameter λ:

$$\lambda = \mu\left(Kloss^t\right) + t\sigma\left(Kloss^t\right)/T \tag{7}$$

where $Kloss^t$ is a loss vector at the t-th training step and is calculated as: $Kloss_i^t = \left\| h_i^{*t} - U^t S_{:,i}^t \right\|_2^2$ for the i-th sample. $\mu\left(Kloss^t\right)$ and $\sigma\left(Kloss^t\right)$ denote the average and standard deviation of vector $Kloss^t$, respectively. T is the maximum training iterations.

By introducing the weight vector v and parameter λ, the proposed CDIMC-net can select the most confident samples whose clustering losses are no more than λ for training. Then with the iteration increases, more confident samples will be selected. This process is similar to the human cognitive learning, *i.e.*, learns from easy to hard or less to more.

Loss function of fine-tuning: Combining the losses of k-means and graph embedding, the overall loss function of the fine-tuning network is:

$$\begin{array}{l} \min\limits_{S,v,\Omega_1,\lambda} \frac{1}{nk} \sum\limits_{i=1}^{n} \left(r_i \left\| h_i^* - US_{:,i} \right\|_2^2 - \lambda r_i \right) \\ + \alpha \frac{1}{2nl} \sum\limits_{v=1}^{l} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \left\| h_i^{(v)} - h_j^{(v)} \right\|_2^2 N_{i,j}^{(v)} \\ s.t.\, r_i \in \{0,1\}, S \in \{0,1\}^{k \times n}, S^T I = I \end{array} \tag{8}$$

At the end of fine-tuning, CDIMC-net will produce the clustering result S for the incomplete multi-view data.

3.3 Optimization

Similar to conventional Autoencoer, all parameters of the pre-training network can be directly optimized by the Stochastic Gradient Descent (SGD) algorithm and back-propagation. Thus, in this section, we focus on the optimization of the fine-tuning network, where an alternating optimization algorithm is adopted to optimize loss function (8).

Step 1: Update the encoder parameters Ω_1: The optimization problem for encoder parameters Ω_1 is:

$$\begin{array}{l} \min\limits_{\Omega_1} \frac{1}{nk} \sum\limits_{i=1}^{n} \left(v_i \left\| \frac{\sum_{v=1}^{l} f_{EC}^{(v)}\left(x_i^{(v)}\right) W_{i,i}^{(v)}}{\sum_{v=1}^{l} W_{i,i}^{(v)}} - US_{:,i} \right\|_2^2 \right) \\ + \alpha \frac{1}{2nl} \sum\limits_{v=1}^{l} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \left\| f_{EC}^{(v)}\left(x_i^{(v)}\right) - f_{EC}^{(v)}\left(x_j^{(v)}\right) \right\|_2^2 N_{i,j}^{(v)} \end{array} \tag{9}$$

Problem (9) can be adaptively optimized via SGD and back-propagation.

Step 2: Update cluster indicator S: S is updated by solving the following problem:

$$\min_{S \in \{0,1\}^{k \times n}, S^T I = I} \left\| H^* - US \right\|_F^2 \tag{10}$$

The optimal solution to problem (10) is:

$$S_{i,j} = \begin{cases} 1, & if\, j = \arg\min\limits_{c} \left\| h_i^* - U_{:,c} \right\|_2^2 \\ 0, & otherwise \end{cases} \tag{11}$$

Step 3: Update r: The optimization problem to variable r is degraded as follows by fixing the other variables:

$$\min_{r_i \in \{0,1\}} \sum_{i=1}^{n} \left(r_i \left\| h_i^* - US_{:,i} \right\|_2^2 - \lambda r_i \right) \tag{12}$$

Supposing $Kloss_i = \left\| h_i^* - US_{:,i} \right\|_2^2$, the optimal solution to problem (12) can be expressed as follows:

$$r_i = \begin{cases} 1, & if\, Kloss_i \le \lambda \\ 0, & otherwise \end{cases} \tag{13}$$

Step 4: Update λ: λ is updated via (7).

By alternatively updating the above variables, the proposed CDIMC-net can converge to the local optimal solution.

3.4 Implementation for IMC

It should be noted that deep learning commonly groups the given data into several subsets and then feeds these subsets batch by batch for model training. However, for our method, it is difficult to sufficiently utilize the local geometric information of graphs $N^{(v)}$ via the conventional batch-to-batch training approach. To solve this issue, we propose a simple approach to explore such local information as much as possible. Based on the assumption that samples from the same cluster are more likely to have connections marked by edge value '1', we propose to reorder the given samples $\left\{X^{(v)}\right\}_{v=1}^{l}$ first according to the initialized clustering result obtained by performing kmeans on the features stacked by all views and then construct the nearest neighbor graphs $\left\{N^{(v)}\right\}_{v=1}^{l}$ from the reordered data, followed by feeding the batch of samples one by one for network training. In this way, every sub-block $\left\{N_{batch}^{(v)}\right\}_{v=1}^{l}$ corresponding to the selected batch of samples $\left\{X_{batch}^{(v)}\right\}_{v=1}^{l}$ will carry dense nearest neighbor information as much as possible such that more local information can be utilized. Specifically, the detail implementation steps of our CDIMC-net for IMC are presented as follows:

Step 1: Data rearrangement and nearest neighbor graph construction: 1) Concatenating all views into one single view, where the missing instances denoted by 'NaN' are filled in the average instance of the corresponding view; 2) Performing kmeans on the stacked view; 3) Reorder data according to the clustering result, where samples grouped into the same cluster are placed together; 4) Construct the nearest neighbor graph according to (4) from the reordered data. The

Algorithm 1: Fine-tuning and clustering of CDIMC-net

```
Input: Arranged incomplete multi-view data
       $\{Y^{(v)}\}_{v=1}^{l}$, indicator matrix $\{\bar{W}^{(v)}\}_{v=1}^{l}$, and
       graphs $\{N^{(v)}\}_{v=1}^{l}$; parameter $\alpha$; Maximum
       iterations: $T$; Maximum iterations for inner loop:
       $Maxiter$; Batch size: $b_s$; Stopping threshold: $\xi$.
Output: Clustering indicator $S$.
1  Initialization: Feed $\{Y^{(v)}\}_{v=1}^{l}$, $\{\bar{W}^{(v)}\}_{v=1}^{l}$, and
   $\{N^{(v)}\}_{v=1}^{l}$ into the pre-trained network to obtain the
   consensus representation $H^*$, and then implement
   kmeans on it to obtain the initialized cluster center
   matrix $U$ and cluster indicator matrix $S$. Set all elements
   of $r$ as 1.
2  for $t \in \{1, 2, \ldots, T\}$ do
3      for $j \in \{1, 2, \ldots, Maxiter\}$ do
4          Update the network parameters by optimizing (9)
           batch to batch;
5      Update cluster indicator $S$ using (11);
6      Update weight vector $r$ using (13);
7      Update step-parameter $\lambda$ using (7);
8      if $1 - \frac{1}{n} \sum_{i,j} S_{i,j}^{t} S_{i,j}^{t-1} < \xi$ then
9          Stop training;
10 return $S$.
```

Database	# Class	# View	# Samples	# Features
Handwritten	10	5	2000	76/216/64/240/47
BDGP	5	4	2500	79/1000/500/250
MNIST	10	2	4000	784/784

Table 1: Description of the multi-view databases

rearranged data is denoted by $\left\{Y^{(v)}\right\}_{v=1}^{l}$, the corresponding nearest neighbor graph and view indicator matrix are denoted by $\left\{N^{(v)}\right\}_{v=1}^{l}$ and $\left\{\bar{W}^{(v)}\right\}_{v=1}^{l}$, respectively.

Step 2: Network pre-training: Exploit the rearranged data, graphs, and indicator matrices to train the IMC Autoencoder network, where all features of the missing views are set as 0. For each batch, we exploit SGD to optimize the loss function $\min_{\Omega_1,\Omega_2} \sum_{v=1}^{l} \frac{1}{m_v b_s} \left\| \left(Y_{batch}^{(v)} - \bar{Y}_{batch}^{(v)}\right) \bar{W}_{batch}^{(v)} \right\|_F^2 + \alpha \frac{1}{b_s l} \sum_{v=1}^{l} Tr\left(H_{batch}^{(v)} L_{N_{batch}^{(v)}} H_{batch}^{(v)T}\right)$, where $Y_{batch}^{(v)}$ denotes the selected batch of data, $\bar{Y}_{batch}^{(v)}$ and $H_{batch}^{(v)}$ are the corresponding reconstructed data and feature representation, $\bar{W}_{batch}^{(v)}$ and $L_{N_{batch}^{(v)}}$ are the sub-indicator matrix and Laplacian matrix of the v-th graph corresponding to the batch of data, b_s denotes the batch size.

Step 3: Network fine-tuning: The detailed fine-tuning steps are summarized in Algorithm 1.

4 Experiment

4.1 Experimental Settings

Databases: Three databases listed in Table 1 are adopted. 1) **Handwritten** [Asuncion and Newman, 2007]: It contains five views and 2000 samples from ten numerals (*i.e.*, 0-9), where the five views are obtained by Fourier coefficients, profile correlations, Karhunen-Love coefficient, Zernike moments, and pixel average extractors. 2) Berkeley Drosophila Genome Project gene expression pattern database (**BDGP**): BDGP is composed of 5 categories and 2500 samples, where each class has 500 samples [Cai *et al.*, 2012]. Each sample is represented by four views, *i.e.*, texture feature and three kinds of visual features extracted from the lateral, dorsal, and ventral images. 3) **MNIST** [LeCun, 1998]. Following [Wang *et al.*, 2018], we evaluate CDIMC-net on the same subset of the MNIST database, which is composed of 4000 samples and ten digits. Pixel feature and edge feature are extracted as two views.

Compared methods: Compared methods include: PMVC, IMG, MIC, OMVC, DAIMC, OPIMC, IMC with graph regularized matrix factorization (IMC_GRMF) [Wen *et al.*, 2018b], MKKM-IK-MKC, and PMVC_CGAN. Besides, two baseline methods, *i.e.*, best single view (BSV) [Zhao *et al.*, 2016] and Concat [Zhao *et al.*, 2016] are also evaluated, where BSV reports the results of the best view, and Concat implements the kmeans on the stacked features of all views. For CDIMC-net, the encoder and decoder networks are stacked by four full connected layers with size of $[0.8m_v, 0.8m_v, 1500, k]$ and $[k, 1500, 0.8m_v, 0.8m_v]$, respectively. The activation function is 'ReLU' and the optimizer is 'SGD' for the pre-training network and 'ADAM' for the fine-tuning network. CDIMC-net is implemented on PyTorch and Ubuntu Linux 16.04.

Incomplete data construction: For the data with more than two views, we randomly remove $p\%$ ($p \in \{10, 30, 50, 70\}$) instances from every view under the condition that all samples at least have one view. For MNIST database, $p\%$ ($p \in \{10, 30, 50, 70\}$) instances are randomly selected as paired samples whose views are complete, and the remaining samples are treated as single view samples, where half of them only have the first view and the other half of the samples only have the second view.

Evaluation metric: Clustering accuracy (ACC) and normalized mutual information (NMI) [Hu and Chen, 2018].

4.2 Experimental Results and Analysis

Experimental results on the above three databases are listed in Tables 2. We can observe the following points from the results: 1) CDIMC-net significantly outperforms the other methods on the three databases. For instance, on the Handwritten database with a missing-view rate of 50%, CDIMC-net obtains about 91% ACC and 84% NMI, which are about 10% and 16% higher than those of the second best method, respectively. 2) BSV and Concat obtain worse IMC performance than the other methods in most cases. Thus, we can conclude that exploring the complementary information and consistent information of multiple views is beneficial to improve the performance. 3) CDIMC-net performs better than the advanced deep network based method PMVC_CGAN on the MNIST database. This demonstrates that CDIMC-net is superior to PMVC_CGAN for IMC.

4.3 Parameter Analysis

Fig.2 shows the relationships of ACC, graph embedding hyper-parameter α, and the learning rate of the CDIMC-net on the Handwritten and BDGP databases with a missing-view rate of 10%. We can observe that CDIMC-net obtains relatively better performance when the two parameters are small. Specifically, a large learning rate is harmful to obtain the local optimal clustering results and a large α makes the graph embedding term dominate the training phase. In the applications, we suggest selecting the learning rate and α from [1e-5,1e-3].

4.4 Component Analysis

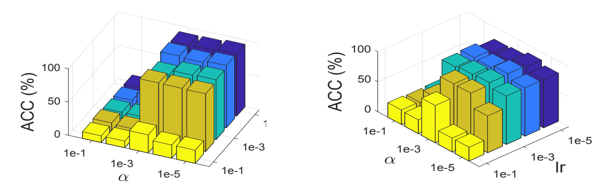
In this subsection, we conduct experiments on the Handwritten and BDGP databases to validate the importance of graph embedding, self-paced learning, and pre-training, where the degenerate models of CDIMC-net without a graph embedding term, self-pace constraint, and pre-training phase, are compared. From Fig.3, we can find that CDIMC-net outperforms the three degenerate models, which demonstrates the effectiveness of the introduced three approaches.

4.5 Convergence Analysis

Fig.4 shows the loss value of fine-tuning network versus the iterations on the Handwritten and BDGP databases with a missing-view rate of 10%. From the figures, we can observe

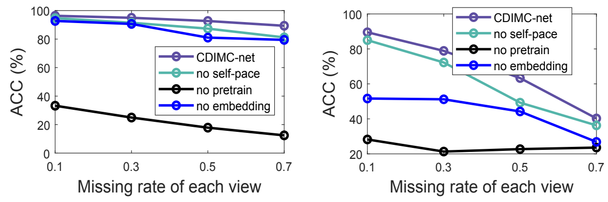
		ACC				NMI			
Database	Method\\$p\%$	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
Handwritten	BSV	68.27±5.66	51.49±2.29	38.24±2.25	27.15±1.31	62.82±3.24	47.01±1.71	32.21±1.00	19.48±0.69
	Concat	75.06±3.86	55.48±1.57	42.19±0.99	28.31±0.75	73.08±2.05	51.66±0.99	38.24±1.59	23.50±0.95
	MIC	77.59±2.41	73.29±3.41	61.27±3.16	41.34±2.69	70.84±2.08	65.39±2.08	52.95±1.33	34.71±2.11
	OMVC	65.04±6.50	55.00±5.06	36.40±4.93	29.80±4.63	56.72±5.05	44.99±4.56	35.16±4.62	25.83±8.37
	DAIMC	88.86±0.63	86.73±0.79	81.92±0.88	60.44±6.87	79.78±0.71	76.65±1.07	68.77±0.99	47.10±4.79
	OPIMC	80.20±5.40	76.45±5.15	69.50±6.54	56.66±10.06	77.26±3.11	73.74±3.42	66.57±4.18	51.86±7.97
	MKKM-IK-MKC	71.78±1.74	69.07±0.73	66.08±3.25	55.55±1.39	69.43±1.28	65.42±0.61	59.04±2.69	47.36±1.78
	CDIMC-net	**95.12±1.11**	**94.22±1.07**	**91.48±0.82**	**88.85±0.77**	**90.10±1.97**	**89.21±1.11**	**84.58±1.08**	**79.99±0.78**
BDGP	BSV	51.48±3.96	41.44±3.55	34.74±1.52	27.95±1.76	35.74±4.01	25.20±2.70	16.39±1.36	9.29±1.67
	Concat	57.66±4.79	50.04±1.58	40.41±3.52	27.52±1.24	44.58±4.78	31.81±1.45	19.76±1.78	6.17±1.27
	MIC	48.31±0.83	40.88±1.18	34.02±1.42	29.45±0.91	28.52±0.49	23.94±1.21	11.05±0.89	7.04±1.17
	OMVC	55.23±4.55	46.22±3.15	39.46±1.12	38.32±2.95	28.78±1.59	19.44±1.20	13.51±1.21	12.74±5.46
	DAIMC	77.34±2.58	69.30±6.42	52.45±8.57	35.68±4.23	55.64±2.68	47.87±4.65	28.33±1.38	9.17±3.64
	OPIMC	79.38±7.69	63.73±7.29	55.17±9.24	35.82±2.68	61.77±7.41	41.47±3.88	25.94±7.29	8.35±2.23
	MKKM-IK-MKC	31.80±1.68	29.12±0.29	29.44±1.39	35.16±1.69	7.21±1.10	5.86±0.56	6.65±1.19	12.19±1.41
	CDIMC-net	**89.02±0.71**	**77.99±1.00**	**62.14±1.67**	**40.98±1.05**	**77.24±1.98**	**57.09±0.77**	**35.64±0.59**	**14.77±0.99**
MNIST	BSV	33.25±1.79	37.37±0.69	42.76±1.30	47.95±1.36	27.20±0.89	31.39±1.28	37.45±1.42	42.49±1.47
	Concat	36.88±1.96	39.24±1.47	43.79±1.71	47.37±1.08	34.48±1.09	33.38±0.54	37.42±1.38	43.17±0.69
	PMVC	41.36±2.29	43.42±2.99	44.68±1.23	45.84±1.59	35.46±0.25	38.51±1.63	39.43±1.37	39.83±1.71
	IMG	46.34±3.36	47.13±2.24	46.88±1.51	48.31±1.22	39.74±2.42	40.71±2.56	39.87±1.05	44.16±1.09
	IMC_GRMF	49.12±2.46	50.59±2.59	52.37±1.67	52.46±1.59	47.36±0.97	48.18±1.57	50.73±0.98	51.57±1.03
	MIC	43.96±2.38	44.42±2.28	44.17±1.37	45.38±2.82	38.77±1.35	40.81±1.28	40.53±0.67	41.61±1.50
	OMVC	40.44±2.95	42.23±2.17	40.36±2.20	41.44±3.39	36.21±1.47	36.68±2.16	35.64±1.89	32.25±2.95
	DAIMC	45.33±4.12	48.19±1.38	49.25±1.67	49.36±1.87	37.46±3.04	41.09±1.58	43.47±0.82	44.15±0.75
	OPIMC	41.40±2.51	48.02±2.63	47.77±3.39	48.71±2.44	34.29±2.33	43.98±1.98	44.63±1.47	45.65±1.15
	MKKM-IK-MKC	47.56±2.18	51.02±0.66	51.72±0.58	52.45±0.41	40.39±1.17	42.76±0.70	43.88±0.54	45.10±0.39
	PMVC_CGAN	45.17±- -	48.36±- -	52.80±- -	52.02±- -	39.33±- -	43.22±- -	49.61±- -	48.22±- -
	CDIMC-net	**51.65±0.14**	**57.64±1.44**	**58.28±0.68**	**59.15±0.21**	**48.25±0.47**	**50.54±1.26**	**51.70±0.67**	**52.87±0.48**

Table 2: Clustering average results and standard deviations of different methods on the Handwritten, BDGP, and MNIST databases with different missing-view rates or paired-view rates $p\%$. Note: the average results of PMVC_CGAN are reported in [Wang *et al.*, 2018].



(a) Handwritten (b) BDGP

Figure 2: ACC (%) v.s. α and learning rate (lr) of CDIMC-net on (a) Handwritten and (b) BDGP databases with a missing-view rate of 10%.



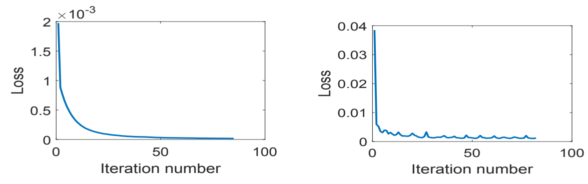
(a) Handwritten (b) BDGP

Figure 3: ACC (%) of CDIMC-net and its three degenerate models on (a) Handwritten and (b) BDGP databases.

that the loss value shows a downward trend overall and decreases quickly in the first few steps. This validates the convergence property of the proposed method.

5 Conclusion

In this paper, we proposed a novel and flexible CDIMC-net, which can handle all kinds of incomplete data. Based on



(a) Handwritten (b) BDGP

Figure 4: Loss v.s. iterations of CDIMC-net on (a) Handwritten and (b) BDGP databases with a missing-view rate of 10%.

human cognitive learning, CDIMC-net introduces the self-paced kmeans to improve the robustness to outliers. Besides this, it incorporates the graph embedding technique to preserve the local structure of data. The superior performance of CDIMC-net are validated on several incomplete cases with the comparison of many state-of-the-art IMC methods.

Acknowledgements

This work is partially supported by Shenzhen Fundamental Research Fund under Grant no. JCYJ20190806142416685, Guangdong Basic and Applied Basic Research Foundation under Grant nos. 2019A1515110582 & 2019A1515110475, Establishment of Key Laboratory of Shenzhen Science and Technology Innovation Committee under Grant no. ZDSYS20190902093015527, University of Macau (File no. MYRG2019-00006-FST), National Natural Science Foundation of China under Grant nos. 61702110 & 61702163, and National Postdoctoral Program for Innovative Talent under Grant no. BX20190100.

References

[Asuncion and Newman, 2007] Arthur Asuncion and David Newman. Uci machine learning repository [http://archive.ics.uci.edu/ml], 2007.

[Cai *et al.*, 2008] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. Non-negative matrix factorization on manifold. In *ICDM*, pages 63–72, 2008.

[Cai *et al.*, 2012] Xiao Cai, Hua Wang, Heng Huang, and Chris Ding. Joint stage recognition and anatomical annotation of drosophila gene expression patterns. *Bioinformatics*, 28(12):i16–i24, 2012.

[Chao *et al.*, 2017] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A survey on multi-view clustering. *arXiv preprint arXiv:1712.06246*, 2017.

[Guo *et al.*, 2017] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[Guo *et al.*, 2018] Xifeng Guo, En Zhu, Xinwang Liu, and Jianping Yin. Deep embedded clustering with data augmentation. In *ACML*, pages 550–565, 2018.

[Guo *et al.*, 2019] Xifeng Guo, Xinwang Liu, En Zhu, Xinzhong Zhu, Miaomiao Li, Xin Xu, and Jianping Yin. Adaptive self-paced deep clustering with data augmentation. *IEEE TKDE*, 2019.

[Hu and Chen, 2018] Menglei Hu and Songcan Chen. Doubly aligned incomplete multi-view clustering. In *IJCAI*, pages 2262–2268, 2018.

[Hu and Chen, 2019] Menglei Hu and Songcan Chen. One-pass incomplete multi-view clustering. In *AAAI*, pages 3838–3845, 2019.

[Kang *et al.*, 2017] Zhao Kang, Chong Peng, and Qiang Cheng. Clustering with adaptive manifold structure learning. In *ICDE*, pages 79–82, 2017.

[LeCun, 1998] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[Li *et al.*, 2014] Shao-Yuan Li, Yuan Jiang, and Zhi-Hua Zhou. Partial multi-view clustering. In *AAAI*, pages 1968–1974, 2014.

[Liu *et al.*, 2019] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, En Zhu, Tongliang Liu, Marius Kloft, Dinggang Shen, Jianping Yin, and Wen Gao. Multiple kernel k-means with incomplete kernels. *IEEE TPAMI*, 2019.

[Nie *et al.*, 2019] Feiping Nie, Cheng-Long Wang, and Xuelong Li. K-multiple-means: A multiple-means clustering method with specified k clusters. In *ACM SIGKDD*, pages 959–967, 2019.

[Rai *et al.*, 2016] Nishant Rai, Sumit Negi, Santanu Chaudhury, and Om Deshmukh. Partial multi-view clustering using graph regularized nmf. In *ICPR*, pages 2192–2197, 2016.

[Shao *et al.*, 2015] Weixiang Shao, Lifang He, and S Yu Philip. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l\_2, 1$ regularization. In *ECML PKDD*, pages 318–334, 2015.

[Shao *et al.*, 2016] Weixiang Shao, Lifang He, Chun-ta Lu, and S Yu Philip. Online multi-view clustering with incomplete views. In *ICBD*, pages 1012–1017, 2016.

[Trivedi *et al.*, 2010] Anusua Trivedi, Piyush Rai, Hal Daumé III, and Scott L DuVall. Multiview clustering with incomplete views. In *NIPS Workshop*, volume 224, pages 1–8, 2010.

[Wang *et al.*, 2018] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu. Partial multi-view clustering via consistent GAN. In *ICDM*, pages 1290–1295, 2018.

[Wang *et al.*, 2019] Hao Wang, Linlin Zong, Bing Liu, Yan Yang, and Wei Zhou. Spectral perturbation meets incomplete multi-view data. *arXiv preprint arXiv:1906.00098*, 2019.

[Wen *et al.*, 2018a] Jie Wen, Na Han, Xiaozhao Fang, Lunke Fei, Ke Yan, and Shanhua Zhan. Low-rank preserving projection via graph regularized reconstruction. *IEEE TCYB*, 49(4):1279–1291, 2018.

[Wen *et al.*, 2018b] Jie Wen, Zheng Zhang, Yong Xu, and Zuofeng Zhong. Incomplete multi-view clustering via graph regularized matrix factorization. In *ECCV Workshops*, pages 593–608, 2018.

[Wen *et al.*, 2019] Jie Wen, Zheng Zhang, Yong Xu, Bob Zhang, Lunke Fei, and Hong Liu. Unified embedding alignment with missing views inferring for incomplete multi-view clustering. In *AAAI*, volume 33, pages 5393–5400, 2019.

[Wen *et al.*, 2020] Jie Wen, Yong Xu, and Hong Liu. Incomplete multiview spectral clustering with adaptive graph learning. *IEEE TCYB*, 50(4):1418–1429, 2020.

[Xiang *et al.*, 2013] Shuo Xiang, Lei Yuan, Wei Fan, Yalin Wang, Paul M Thompson, and Jieping Ye. Multi-source learning with block-wise missing data for alzheimer's disease prediction. In *ACM SIGKDD*, pages 185–193, 2013.

[Xie *et al.*, 2016] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.

[Yang *et al.*, 2017] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, pages 3861–3870. JMLR, 2017.

[Zhang *et al.*, 2018] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. *IEEE TPAMI*, 41(7):1774–1782, 2018.

[Zhao *et al.*, 2016] Handong Zhao, Hongfu Liu, and Yun Fu. Incomplete multi-modal visual data grouping. In *IJCAI*, pages 2392–2398, 2016.

[Zhao *et al.*, 2018] Liang Zhao, Zhikui Chen, Yi Yang, Z Jane Wang, and Victor CM Leung. Incomplete multi-view clustering via deep semantic mapping. *Neurocomputing*, 275:1053–1062, 2018.