Deep Adversarial Multi-view Clustering Network

Zhaoyang Li[1] , **Qianqian Wang**[1*] , **Zhiqiang Tao**[2] , **Quanxue Gao**[1†] and **Zhaohua Yang**[3]
[1]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China.
[2]Department of Electrical and Computer Engineering, Northeastern University, USA.
[3]School of Instrumentation Science and Opto-electronics Engineering, Beihang University, China.

Abstract

Multi-view clustering has attracted increasing attention in recent years by exploiting common clustering structure across multiple views. Most existing multi-view clustering algorithms use shallow and linear embedding functions to learn the common structure of multi-view data. However, these methods cannot fully utilize the non-linear property of multi-view data that is important to reveal complex cluster structure. In this paper, we propose a novel multi-view clustering method, named Deep Adversarial Multi-view Clustering (DAMC) network, to learn the intrinsic structure embedded in multi-view data. Specifically, our model adopts deep auto-encoders to learn latent representations shared by multiple views, and meanwhile leverages adversarial training to further capture the data distribution and disentangle the latent space. Experimental results on several real-world datasets demonstrate the proposed method outperforms the state-of art methods.

1 Introduction

Clustering analysis is a fundamental task in a wide range of fields, such as machine learning, pattern recognition, computer vision and data mining. A great deal of research efforts have been made in this topic, among which multi-view clustering [Yang and Wang, 2018] is with particular interest. Multi-view data provide complementary information for the clustering task, which is accessible in many real-world applications. For example, an image could be characterized by various descriptors, such as SIFT [Lowe, 2004], histograms of oriented gradients (HOG) [Dalal and Triggs, 2005], GIST [Oliva and Torralba, 2001] and local binary pattern (LBP) [Ojala *et al.*, 2002]. As these features describe the objects' characteristics from distinct perspectives, they are regarded as multi-view data. Recently, multi-view clustering (MVC) methods [Zhao *et al.*, 2017; Luo *et al.*, 2018] have been developed rapidly, the key of which is to explore the complementary information shared among multiple views. On the basis, many advanced MVC algorithms have been investigated in the last few decades.

For instance, [Liu *et al.*, 2013b] solves this problem from the perspective of non-negative matrix factorization, which seeks a common latent factor through non-negative matrix factorization among multiple views. Consistent and Specific Multi-View Subspace Clustering (CSMSC) [Luo *et al.*, 2018] formulates the self-expression property of multi-view data using a common consistent representation and a set of specific representations, which better fits real-world multi-view datasets. Although traditional multi-view clustering algorithms have achieved promising results, they mainly use shallow and linear embedding functions to reveal the intrinsic structure of data, which are unable to model the non-linear nature of complex data.

Recently, deep clustering has been proposed to exploit deep neural networks for modeling the relationship among data samples to get clustering results. For the single-view clustering methods, DSC [Ji *et al.*, 2017] uses stacked auto-encoders as their based model and utilizes self-expressiveness property to learn the affinity of data in a latent space. DAC [Chang *et al.*, 2017] recasts the clustering problem into a binary pairwise-classification framework, which pushes towards similar image pairs into the same cluster. DEC [Xie *et al.*, 2016] designs a new clustering objective function by minimizing the KL divergence between the predicted cluster label distribution with the predefined one. On another hand, several recent attempts have been made to introduce deep learning for solving the multi-view clustering problem. For example, [Andrew *et al.*, 2013] proposes a DNN extension of CCA, termed as deep CCA, for multi-view clustering. For another example, [Abavisani and Patel, 2018] employs convolutional neural networks for unsupervised multi-modal subspace clustering. However, learning a low-dimensional latent space across multiple views via deep neural networks is still under explored.

In this paper, we propose a novel Deep Adversarial Multi-view Clustering (DAMC) network to learn the intrinsic structure embedded in multi-view data (see Figure 1). Our model develops multi-view auto-encoder networks with shared weights to learn effective mapping from original features to a common low-dimensional embedding space. Compared with traditional algorithms, the proposed method can reveal the non-linear property underling multi-view data,

*Contact Author: Q. Wang. (qianqian174@foxmail.com)
†Contact Author: Q. Gao. (qxgao@xidian.edu.cn)

which is important to handle complex and high-dimensional data. Moreover, we adopt adversarial training [Goodfellow *et al.*, 2014] as a regularizer to guide the training of our encoder, which captures the data distribution of each single view and further disentangles the common latent space. Experimental results on image and text datasets demonstrate that the proposed method outperforms other multi-view clustering methods. We summarize our main contributions as follows.

- We propose a novel Deep Adversarial Multi-view Clustering (DAMC) network. Different from existing multi-view clustering methods, the proposed method can fully model multi-layer nonlinear correlations between arbitrary views.
- We develop a discriminator network for each view specifically, which could further capture the data distribution and disentangle the latent space.
- We design a clustering loss to constrain common representation by minimizing the relative entropy between the predicted label distribution with the predefined one.

2 Related Work

2.1 Multi-view Clustering

For traditional multi-view clustering algorithms, we can divide the existing methods into five groups: First, some methods [Liu *et al.*, 2013b; Zhao *et al.*, 2017] use non-negative matrix factorization techniques for multi-view clustering, which seeks a common latent factor among multi-view data. Another methods are to use the multi-kernel learning (MKL) strategy to solve this problem. Among the multi-view clustering, different predefined kernels are used to deal with different views. These kernels are combined either linearly or nonlinearly in order to arrive at a unified kernel. The methods in the third stream firstly project each view of features into a common low-dimensional subspace, and then conduct clustering in this subspace. A representative method in this stream is canonical correlation analysis(CCA) for multi-view clustering [Chaudhuri *et al.*, 2009], which uses CCA to project the multi-view high dimensional data into a low-dimensional subspace. In addition, the subspace clustering methods have been proposed to explore the relationships between samples with self-representation [Elhamifar and Vidal, 2013; Liu *et al.*, 2013a], the work in [Guo, 2013] formulated the subspace learning with multiple views as a joint optimization problem with a common subspace representation matrix and a group sparsity inducing norm. And the work in [Luo *et al.*, 2018] simultaneously learned a view-consistent representation and a set of view-specific representations for multi-view subspace clustering. At last, most of people exploit the multi-view features with graph-based models [Xia *et al.*, 2014; Tao *et al.*, 2017; Tao *et al.*, 2019; Nie *et al.*, 2017a]. This category of methods seeks to find a fusion graph across all views and then uses graph-cut algorithms or other technologies (e.g., spectral clustering) on the fusion graph in order to produce the clustering result.

2.2 Deep Multi-view Learning

Deep neural networks (DNN) composed of multiple nonlinear transformations can learn a better feature representation than traditional shallow models. One representative method is based on deep auto-encoders, where the goal is to extract a common representation that can reconstruct the inputs of multiple views. In this scenario, a common encoder is utilized to extract common representations for all views, and different decoders are used to reconstruct view-specific input features from the common representation. SplitAE [Ngiam *et al.*, 2011] has been presented to be effective for multi-view learning in speech and vision tasks based on deep auto-encoder networks. The other is based on canonical correlation analysis (CCA), such as, [Andrew *et al.*, 2013] , which proposes a DNN extension of CCA (DCCA) to learn the common representation of two views. In DCCA, two networks were employed to extract non-linear features for each view and the correlations between the extracted features were maximized by CCA on the top layer. Following this line, the deep canonically correlated auto-encoders (DCCAE) was developed in [Wang *et al.*, 2016]. Different from DCCA, DCCAE optimized the canonical correlation between the learned features and the reconstruction errors of the auto-encoders for two views together.

3 Deep Adversarial Multi-view Clustering

3.1 Network Architecture

Given a dataset of V views $\chi = \{\mathbf{X}^1, \ldots, \mathbf{X}^v, \ldots, \mathbf{X}^V\}$, where $\mathbf{X}^v \in R^{d_v \times n}$ denotes the n samples of dimension d_v from the v-th view, we build a DAMC network consists of one fully connected multi-view denoising encoder $\mathbf{E}$, one fully connected multi-view denoising generator $\mathbf{G}$, V fully connected discriminators, and one deep embedding clustering layer on the top of our encoder. Figure 1 illustrates a DAMC network for the V-views case.

1. Multi-view denoising Encoder $\mathbf{E}$: In our multi-view denoising encoder network, for each view, there are M-layer independent fully connected networks and N-layer fully connected networks with shared parameters. The independent layers are used to handle the different feature dimensions of each view. For v-th view, given $\mathbf{X}^v = \{x_1^{(v)}, x_2{}^{(v)}, ..., x_n^{(v)}\}$, the multi-view denoising encoder $\mathbf{E}$ aims to learn a latent representation $\mathbf{Z}^v = \{z_1^{(v)}, z_2^{(v)}, ..., z_n^{(v)}\} (\mathbf{Z}^v \in R^{m \times n})$ for v-th view. Specifically, it maps the d_v-dimensional input data $x_i^{(v)}$ to a low-dimensional representation $z_i^{(v)}$. This mapping could be represented as $\mathbf{Z}^v = f_v(\mathbf{X}^v; \Theta_E)$, where f_v refers to v-th view's encoding network parameterized by Θ_E.

2. Multi-view denoising generator $\mathbf{G}$: Our multi-view denoising generator network has an opposite architecture to our mult-view denoising encoder $\mathbf{E}$. It consists of N-layer fully connected networks with the same parameters and M-layer independent fully connected networks for each view, which can generate all visual reconstructed samples with the latent representations corresponding to each view. Specifically, we suppose $\{\mathbf{Y}^1, \mathbf{Y}^2, ..., \mathbf{Y}^v, ..., \mathbf{Y}^V\} = \mathbf{G}(\mathbf{Z}^v)$, where $\mathbf{Y}^v$ represents the reconstructed sample matrix of the v-th view.

3. Discriminator network $\mathbf{D}_v$: The discriminator network consists of V fully connected-layer discriminators. Each discriminator $\mathbf{D}_v$ consists of 3 fully connected layers, and it should distinguish that $y_i^{(v)}$ is a generated sample and $x_i^{(v)}$ is

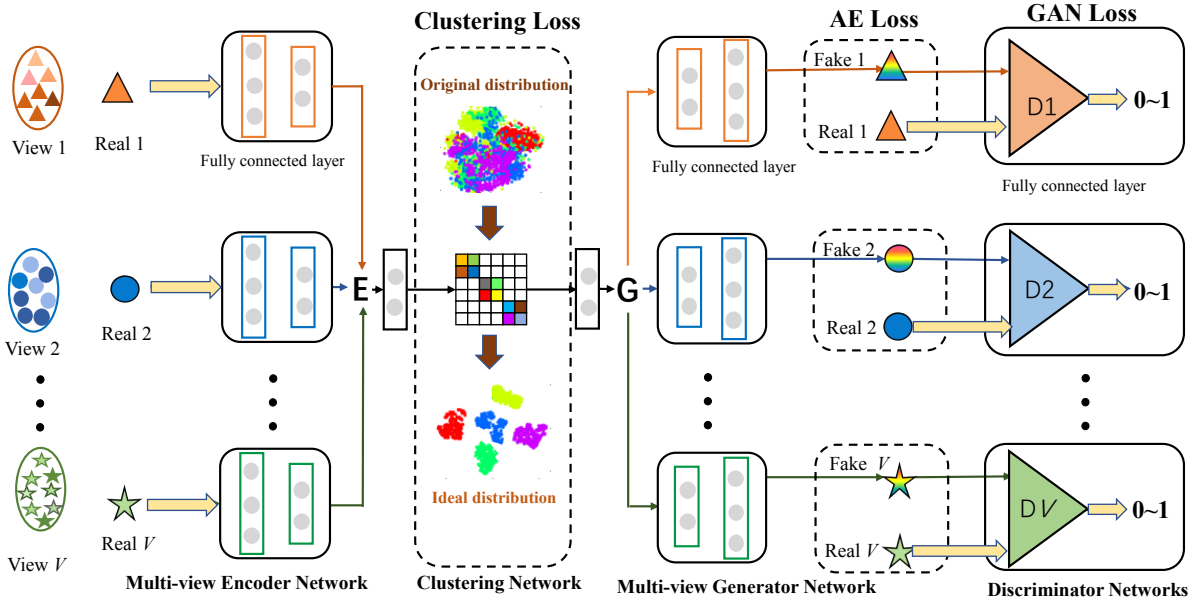


Figure 1: Illustration of Deep Adversarial Multi-view Clustering (DAMC) network. DAMC consists of one multi-view encoder network **E**, one multi-view generator network **G**, V discriminator networks, and one deep embedding clustering layer. Multi-view encoder network **E** outputs a low-dimensional latent layer feature z^v for each view. For each z^v, multi-view generator network **G** generates reconstructed samples. Discriminator network is used to distinguish generated sample or real one. Clustering layer can improve clustering performance by minimizing the KL divergence between the data distribution with the ideal distribution.

a real instance. $\mathbf{D}_v$ feeds back the result to generator network and updates the parameters of generator. By this means, the discriminator works as a regularizer to guide the training of our multi-view encoder network, which enhances the robustness of embedding representations and avoids the over-fitting issue effectively.

4. Deep embedding clustering layer: In order to seek for a clustering-friendly latent space, we embed a unique clustering layer in the network. The embedded clustering layer contains the new clustering centroids after each iteration. We obtain the current data distribution and target data distribution based on the common space $\mathbf{Z}$ and the cluster centroids $\{\mu_j\}_{j=1}^k$. We employ the KL divergence of the current data distribution and the target data distribution as the objective function to iteratively update the parameters of the multi-view denoising encoder network **E** and multi-view denoising generator **G** network.

3.2 Loss Function

The total loss function of our model is defined as follows

$$\mathcal{L} = \min_{\mathbf{E},\mathbf{G}} \max_{\mathbf{D}_1,\ldots,\mathbf{D}_V} \mathcal{L}_{AE} + \lambda_1 \mathcal{L}_{GAN} + \lambda_2 \mathcal{L}_{CLU}, \quad (1)$$

which consists of three parts: the AE loss $\mathcal{L}_{AE}$, the GAN loss $\mathcal{L}_{GAN}$, and the clustering loss $\mathcal{L}_{CLU}$. λ_1 and λ_2 are two parameters to maintain the impact of GAN loss and clustering loss.

Auto-Encoder Loss

AE loss is measured by the mean square error between the generated sample and the real sample. When inputing the first view $\mathbf{X}^1$, the multi-view denoising encoder **E** outputs a low-dimensional latent layer representation $\mathbf{Z}^1 = f_1(\mathbf{X}^1; \theta_E)$. Similarly, for the V-th view, the result is $\mathbf{Z}^V = f_V(\mathbf{X}^V; \theta_E)$. After that, the multi-view denoising generator **G** reconstructs the V views from any latent representation $\mathbf{Z}^v$. The outputs are $\{\mathbf{Y}_1^1, \mathbf{Y}_1^2, ..., \mathbf{Y}_1^V\} = \mathbf{G}(\mathbf{Z}^1)$, and $\{\mathbf{Y}_V^1, \mathbf{Y}_V^2, ..., \mathbf{Y}_V^V\} = \mathbf{G}(\mathbf{Z}^V)$, where $\mathbf{Y}_1^1, \mathbf{Y}_2^1, ..., \mathbf{Y}_V^1$ are the generated samples corresponding to the first view, and $\mathbf{Y}_1^V, \mathbf{Y}_2^V, ..., \mathbf{Y}_V^V$ correspond to the V-th view. Therefore the AE loss is

$$\mathcal{L}_{AE} = \min_{\mathbf{E},\mathbf{G}} \sum_{i=1}^{V} \sum_{v=1}^{V} ||\mathbf{X}^v - \mathbf{Y}_i^v||_{\mathrm{F}}^2. \quad (2)$$

We minimize the AE loss to optimize our multi-view denoising auto-encoders. However, the mean square error may lead to blurred reconstructed results and cannot model the data distribution of each view. To alleviate this issue, we adopt adversarial training to generate (recover) more realistic results and further enhance the model generalization.

The Loss for Generative Adversarial Networks

There are two models in generative adversarial networks (GANs) [Goodfellow *et al.*, 2014], *i.e.*, a generative model **G** and a discriminative model **D**. Generative model **G** continuously learns the probability distribution of real data in the training set. Its goal is to convert the input random noise into an image that can be faked. The discriminator **D** determines whether an image is a real image. In our model, we draw on this idea and use discriminators to distinguish between generated samples and real samples. Suppose that the real data distribution of V views is $x^1 \sim P(\mathbf{X}^1), x^2 \sim P(\mathbf{X}^2), ..., x^V \sim P(\mathbf{X}^V)$, and the generated data distribution of V views is $y^1 \sim P(\mathbf{Y}^1), y^2 \sim P(\mathbf{Y}^2), ..., y^V \sim P(\mathbf{Y}^V)$. So the GAN

loss in our model can be described as

$$\mathcal{L}_{GAN} = \min_{\mathbf{E},\mathbf{G}} \max_{\mathbf{D}_1,...,\mathbf{D}_V} \sum_{v=1}^{V} (\mathbb{E}_{x^v \sim P(\mathbf{X}^v)}[\log \mathbf{D}_v(x^v)] + \mathbb{E}_{y^v \sim P(\mathbf{Y}^v)}[\log(1 - \mathbf{D}_v(y^v))]). \tag{3}$$

The notation $\mathbb{E}$ representa: $\mathbb{E}_{x\sim P(X)}[f(x)] = \frac{1}{N}\sum_{i=1}^{N} f(x^i)$, where N is the number of samples. By training the multi-view denoising encoder and the multi-view denoising generator, we generate fake data similar to real data of each view. The discriminators are trained to distinguish the fake data from the real data of every views. They play a min-max game until convergence. However, GANs are trained to map the same input to any random permutation of samples from a target data distribution. Hence, the GAN loss cannot ensure the desired output at an instance level, which is not suitable for a clustering task. In light of this, we incorporate GAN loss with the AE loss to achieve a high reliability of data reconstruction.

Clustering Loss

The AE loss and the GAN loss enable our multi-view denoising generator to generate fake samples that are more similar to real ones, which encourage our embedding representations to contain original feature information as much as possible. However, they cannot guarantee that the encoded low-dimensional space has a good cluster structure. To seek for a partitioning friendly space, we encapsulate the clustering loss measured by KL-divergence in our DAMC network. Specifically, we learn V latent representations for the V views $\mathbf{Z}^1 = f_1(\mathbf{X}^1;\theta_E), \mathbf{Z}^2 = f_2(\mathbf{X}^2;\theta_E), ..., \mathbf{Z}^V = f_V(\mathbf{X}^V;\theta_E)$. Then we get a common latent representation based on these V views as

$$\mathbf{Z} = \frac{1}{V}\sum_{v=1}^{V} \mathbf{Z}^v. \tag{4}$$

Given the initial cluster centroids $\{\mu_j\}_{j=1}^k$, according to [Xie *et al.*, 2016], we use the Student's t-distribution as a kernel to measure the similarity between common latent representation point z_i and centroid μ_j

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + ||z_i - \mu_{j'}||^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{5}$$

where α are the degrees of freedom of the Student's t-distribution, q_{ij} is interpreted as the probability of assigning sample i to cluster j, it can be also named soft assignment. In our all experiments, we let $\alpha = 1$. We propose to iteratively refine the clusters by learning from their high confidence assignments with the help of an auxiliary target distribution. In our model, it is trained by matching the soft assignment to the target distribution. To this end, we define our objective as a KL divergence loss between the soft assignment q_{ij} and the auxiliary distribution p_{ij} as follows

$$\mathcal{L}_{CLU} = \min_{\mathbf{E},\mathbf{G}} \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{6}$$

In our experiments, we compute p_i by raising q_i to its second power and normalizing it with the frequency per cluster as follows

$$p_{ij} = \frac{q_{ij}^2/f_i}{\sum_{j'} q_{ij'}^2/f_{j'}}, \tag{7}$$

where $f_j = \sum_i q_{ij}$ are soft cluster frequencies. In this way, we can concentrate the same class data by sharping the data distribution and get a more effective and common representation for multi-view clustering.

3.3 Training Procedure

Step 1: Training multi-view denoising encoder $\mathbf{E}$ and multi-view denoising generator $\mathbf{G}$ by minimizing AE loss. Specifically, we take $\{x^1, x^2, ..., x^V\}$ as input for multi-view denoising encoder $\mathbf{E}$ and get V latent layer feature $\{z^1, z^2, ..., z^V\}$, Then we take $\{z^1, z^2, ..., z^V\}$ as the input of multi-view denoising generator $\mathbf{G}$ and get V^2 outputs. For any latent layer feature z^v, it can generate reconstruction samples of V views. Then we update multi-view denoising encoder $\mathbf{E}$ and multi-view denoising generator $\mathbf{G}$ by minimizing AE loss. After step 1, we get common representation $\mathbf{Z}$, then we save the clustering centroids $\{\mu_j\}_{j=1}^k$ for the following training by performing k-means algorithm on $\mathbf{Z}$.

Step 2: Training multi-view denoising encoder $\mathbf{E}$, multi-view denoising generator $\mathbf{G}$ and discriminators $\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_V$ by optimizing the sum of AE loss and GAN loss. As with the first step, we get V^2 outputs corresponding to V views by multi-view denoising encoder $\mathbf{E}$ and multi-view denoising generator $\mathbf{G}$. Then, we send these generated samples and corresponding real samples to the discriminative networks $\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_V$ respectively. After that we iteratively update the multi-view denoising encoder-generator network and the discriminative networks by optimizing the sum of AE loss and GAN loss.

Step 3: Training multi-view denoising encoder $\mathbf{E}$, multi-view denoising generator $\mathbf{G}$, discriminators $\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_V$, and embedded clustering layer. Our embedded clustering layer contains the new clustering centroids after each iteration. In the beginning, we use the clustering centroids $\{\mu_j\}_{j=1}^k$ from step 1 and the common representation $\mathbf{Z}$ to calculate the clustering loss. Then, we use the total loss function to train the entire network. In each iteration, we update the clustering centroids. After the training is completed, we use the obtained common representation to perform spectral clustering to obtain the final clustering result.

4 Experiments

4.1 Experimental Settings

Datasets. To demonstrate the performance of the proposed framework, we evaluate DAMC on four multi-view datasets. A brief introduction is given as follows. *(1) Image dataset*: Handwritten numerals (HW) dataset [Asuncion and Newman, 2007] is composed of 2,000 data points from 0 to 9 ten digit classes and each class has 200 data points. In the experiment, we adopt 76 Fourier coefficients of the character shapes and 216 profile correlations as two different views. *(2) Image & text dataset*: BDGP [Cai *et al.*, 2012] is a two-view dataset including two different modalities, *i.e.,* visual and textual data. It contains 2,500 images about drosophila

Method	ACC	NMI	Purity
$SC_{v=1}$	0.494	0.286	0.494
$SC_{v=2}$	0.940	0.894	0.942
ConSC	0.584	0.384	0.584
RMSC	0.602	0.563	0.602
AMGL	0.958	0.904	0.958
MLAN	0.681	0.488	0.681
MVSC	0.948	0.849	0.948
SwMC	0.953	0.887	0.953
CSMSC	0.968	0.911	0.968
DCCA	0.578	0.409	0.578
DMSC	0.681	0.506	0.738
DAMC	**0.982**	**0.946**	**0.982**

Table 1: Experiments on the BDGP dataset.

Method	ACC	NMI	Purity
$SC_{v=1}$	0.682	0.663	0.699
$SC_{v=2}$	0.651	0.667	0.691
ConSC	0.828	0.802	0.831
RMSC	0.737	0.708	0.763
AMGL	0.806	0.791	0.828
MLAN	0.741	0.754	0.773
MVSC	0.682	0.569	0.684
SwMC	0.804	0.798	0.829
CSMSC	0.798	0.764	0.812
DCCA	0.814	0.781	0.814
DMSC	0.916	0.855	0.916
DAMC	**0.965**	**0.932**	**0.965**

Table 2: Experiments on the HW dataset.

Method	ACC	NMI	Purity
$SC_{v=1}$	0.105	0.007	0.107
$SC_{v=2}$	0.195	0.176	0.220
$SC_{v=3}$	0.107	0.006	0.108
ConSC	0.106	0.006	0.108
RMSC	0.216	0.180	0.241
AMGL	0.110	0.014	0.117
MVSC	0.193	0.152	0.210
SwMC	0.209	0.155	0.235
CSMSC	0.239	0.187	0.278
DCCA	0.207	0.159	0.219
DMSC	0.175	0.135	0.251
DAMC	**0.256**	**0.225**	**0.286**

Table 3: Experiments on the CCV dataset.

embryos belonging to 5 categories. Each image is represented by a 1,750-D visual vector and a 79-D textual feature vector. In our experiment, we use the entire BDGP dataset and evaluate the performance on both visual and textual feature. *(3) Video dataset*: The Columbia Consumer Video (CCV) dataset [Jiang *et al.*, 2011] contains 9,317 YouTube videos with 20 diverse semantic categories. In our experiment, we use the subset (6773 videos) of CCV provided by [Jiang *et al.*, 2011], along with three hand-crafted features: STIP features with 5,000 dimensional Bag-of-Words (BoWs) representation, SIFT features extracted every two seconds with 5,000 dimensional BoWs representation, and MFCC features with 4,000 dimensional BoWs representation. *(4) Large-scale dataset*: MNIST is a widely-used benchmark dataset consisting of handwritten digit images with 28×28 pixels. In our experiment, we employ its two-view version ($70,000$ samples) provided by [Shang *et al.*, 2017], where the first view is the original gray images and the other is given by images only highlighting the digit edge.

Comparison Algorithms. We choose spectral clustering and nine state-of-the-art multi-view clustering algorithms as baselines. (1) Feature Concatenation Spectral Clustering (ConSC) [Kumar *et al.*, 2011] concatenates the features of each view, and performs spectral clustering directly on the concatenated feature representation. (2) Robust Multi-view Spectral Clustering (RMSC) [Xia *et al.*, 2014] recovers a latent transition probability matrix from pair-wise similarity matrices of each view through a low-rank constraint. (3) Auto-weighted Multiple Graph Learning (AMGL) [Nie *et al.*, 2016] utilizes each single view to construct a graph, and learns an optimal weight for each graph automatically without introducing additive parameters. (4) Multi-View Clustering and Semi-Supervised Classification with Adaptive Neighbours (MLAN) [Nie *et al.*, 2017a], which performs clustering and local manifold structure learning simultaneously, and allocates weight for each view automatically. (5) Multi-view Spectral Clustering (MVSC) [Li *et al.*, 2015] conducts clustering on the subspace representation of each view simultaneously, and utilizes a common cluster structure to guarantee the consistence among different views. (6) Self-weighted Multi-view Clustering (SwMC) [Nie *et al.*, 2017b] proposes a self-weighted fusion scheme to address multi-view clustering. (7) Consistent and Specific Multi-View Subspace Clustering (CSMSC) [Luo *et al.*, 2018] formulates the multi-view self-representation property using a shared consistent representation and a set of specific representations, which better fits real-world datasets. (8) Deep canonical correlation analysis (DCCA) [Andrew *et al.*, 2013] learns nonlinear transformations of two views such that the resulting representations are highly linearly correlated. (9) Deep Multimodal Subspace Clustering (DMSC) [Abavisani and Patel, 2018] presents CNN based approaches for unsupervised multimodal subspace clustering.

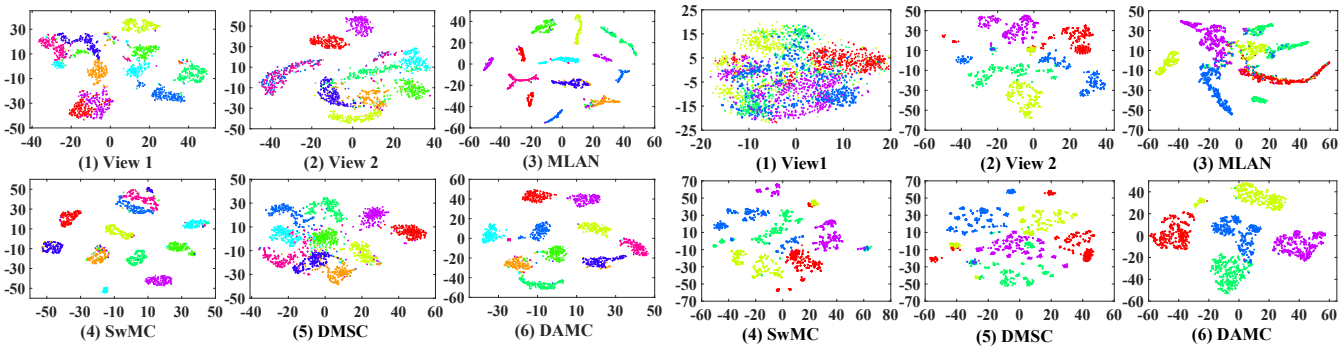
Evaluation Metrics. We evaluate the clustering performance with three standard clustering evaluation metrics, *i.e.*, Accuracy (ACC), Normalized Mutual Information (NMI), and Purity. More details about these metrics could be found in [Kumar *et al.*, 2011]. For all the metrics, higher value indicates better performance.

Implementation Details. We implement our methods and other non-linear methods with the public toolbox of PyTorch. We run all the experiments on the platform of Ubuntu Linux 16.04 with NVIDIA Titan Xp Graphics Processing Units (GPUs) and 32 GB memory size. We use Adam [Kingma and Ba, 2014] optimizer with default parameter setting to train our model and fix the learning rate as 0.0001. We conduct 30 epochs for each training step. All the other linear methods are tested on the same environment by Matlab.

4.2 Experimental Results

In this subsection, we report the comparison on four real-world datasets. Since DCCA can only deal with two views, we choose the best two views on CCV dataset according to their performance. Particularly, several important observations could be made as follows.

Compared with Baselines. Learning discriminative feature representation is crucial for the clustering task. Previous multi-view clustering algorithms mainly employ linear methods to learn common representations shared by multiple views, which cannot handle high-dimensional and complex data due to its non-linear nature. In light of this, our approach fully exploits the non-linear property given by deep neural networks, and further employs adversarial training to capture the data distribution of each view. As shown in Table 1, Table 2 and Table 3, our method significantly outperforms base-



(a) HW (b) BDGP

Figure 2: Visualization of original features for each view and the common latent representations given by different methods with t-SNE [van der Maaten and Hinton, 2008] on the HW and BDGP datsets, where (1) original data of first view, (2) original data of second view, (3) MLAN, (4) SwMC, (5) DMSC, and (6) DAMC.

Method	ACC	NMI	Purity
DCCA	0.468	0.426	0.505
AE	0.607	0.524	0.618
AE+GAN	0.635	0.538	0.639
DAMC	**0.651**	**0.562**	**0.659**

Table 4: Clustering performance on large-scale dataset. We compare our approch with several deep neural network baselines on the two-view MNIST dataset provided by [Shang *et al.*, 2017].

line methods with a clear improvement, which demonstrates the superiority of our algorithm. There MLAN is not available and DMSC can only process one view data on the CCV dataset due to the limited memory.

Compared with CNN Based Methods. Deep Convolutional Neural Networks (CNN) have shown superior performance on learning feature representations for image/video data recently. However, for the clustering task, CNN based methods are limited to grid data, which is not straightforward to handle generic features. For example, DMSC [Abavisani and Patel, 2018] is specifically designed for image data and cannot be directly used with irregular data features (*e.g.*, textual features in BDGP). In our experiment, we adopt zero-padding to make DMSC available on BDGP, HW and CCV datasets, which however, lowers its performance inevitably. Different from CNN based methods, our approach builds on the top of fully-connected network, and thus achieves higher flexibility and generalizability for multi-view clustering.

Clustering on Large-scale Dataset. To show our approach is applicable on the large-scale dataset, we compare the proposed DAMC with DCCA [Andrew *et al.*, 2013] and two strong deep neural network baselines, *i.e.*, AE and AE+GAN, on the two-view MNIST dataset provided by [Shang *et al.*, 2017], where (1) AE empoys the same auto-encoder network architecture to our approach and (2) AE+GAN employs the same AE loss and GAN loss in our model. Both AE and AE+GAN conduct spectral clustering for the final clustering result. It is worth noting that, the other compared multi-view methods are not scalable on this dataset due to their optimization methods and the limited memory. In contrast, our method can easily handle large-scale data with mini-batch training. As shown in Table 4, we consistently outperform other methods with a clear improvement, which validates the effectiveness of DAMC on the large-scale dataset.

Impact of Deep Clustering Layer. In this subsection, we explore the impact of our deep clustering layer. Figure 2 (a) provides a t-SNE [van der Maaten and Hinton, 2008] visualization for feature embeddings in terms of each single view, three competitive compared methods and our proposed DAMC on the HW dataset. In details, we apply t-SNE on the common-view feature representations (*e.g.*, the latent layer features in DAMC) given by different methods, respectively. As can be seen, our approach exhibits a more clear and compact cluster structure than all the other methods and original data. The similar observation could be found on the BDGP dataset as shown in Figure 2 (b). This clearly shows the nice cluster-structured property given by our deep embedded clustering layer, as it explicitly guides our feature learning process with a clustering purpose.

5 Conclusions

In this paper, we proposed a novel Deep Adversarial Multi-view Clustering (DAMC) model, which includes multi-view auto-encoders and a set of view-specific discriminator networks. By using the shared weights, DAMC jointly embeds multi-view data to a common low-dimensional subspace with non-linear mappings. Upon the adversarial training, we employ discriminators to effectively guide the training of our encoder network, which captures the data distribution of each view and further disentangles the latent space. Moreover, we leverage a KL-divergence loss to explicitly encapsulate the clustering task in our network. Experimental results on four real-world datasets demonstrated the superiority of our model over several state-of-the-art multi-view clustering methods.

Acknowledgements

This work is supported by Initiative Postdocs Supporting Program, National Natural Science Foundation of China under Grant 61773302. Natural Science Foundation of Ningbo, China under Grant 2018A610049.

References

[Abavisani and Patel, 2018] Mahdi Abavisani and Vishal M Patel. Deep multimodal subspace clustering networks. *arXiv preprint arXiv:1804.06498*, 2018.

[Andrew *et al.*, 2013] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013.

[Asuncion and Newman, 2007] Arthur Asuncion and David Newman. Uci machine learning repository, 2007, 2007.

[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.

[Chang *et al.*, 2017] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *ICCV*, pages 5880–5888, 2017.

[Chaudhuri *et al.*, 2009] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, pages 129–136, 2009.

[Dalal and Triggs, 2005] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, volume 1, pages 886–893, 2005.

[Elhamifar and Vidal, 2013] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI*, 35(11):2765–2781, 2013.

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[Guo, 2013] Yuhong Guo. Convex subspace representation learning from multi-view data. In *AAAI*, page 2, 2013.

[Ji *et al.*, 2017] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *NIPS*, pages 24–33, 2017.

[Jiang *et al.*, 2011] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, page 29, 2011.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kumar *et al.*, 2011] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[Li *et al.*, 2015] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*, 2015.

[Liu *et al.*, 2013a] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI*, 35(1):171–184, 2013.

[Liu *et al.*, 2013b] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *ICDM*, pages 252–260, 2013.

[Lowe, 2004] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[Luo *et al.*, 2018] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. Consistent and specific multi-view subspace clustering. In *AAAI*, 2018.

[Ngiam *et al.*, 2011] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.

[Nie *et al.*, 2016] Feiping Nie, Jing Li, Xuelong Li, et al. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *IJCAI*, pages 1881–1887, 2016.

[Nie *et al.*, 2017a] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*, pages 2408–2414, 2017.

[Nie *et al.*, 2017b] Feiping Nie, Jing Li, Xuelong Li, et al. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, pages 2564–2570, 2017.

[Ojala *et al.*, 2002] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.

[Oliva and Torralba, 2001] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[Shang *et al.*, 2017] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi. VIGAN: missing view imputation with generative adversarial networks. *CoRR*, abs/1708.06724, 2017.

[Tao *et al.*, 2017] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. From ensemble clustering to multi-view clustering. In *IJCAI*, pages 2843–2849, 2017.

[Tao *et al.*, 2019] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu. Marginalized multiview ensemble clustering. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2019.

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[Wang *et al.*, 2016] Weiran Wang, Raman Arora, Karen Livescu, and Jeff A Bilmes. On deep multi-view representation learning: Objectives and optimization. *arXiv: Learning*, 2016.

[Xia *et al.*, 2014] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, pages 2149–2155, 2014.

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

[Yang and Wang, 2018] Yan Yang and Hao Wang. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2):83–107, 2018.

[Zhao *et al.*, 2017] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017.