Multi-View Representation Learning with Manifold Smoothness

Shu Li, Wei Wang *, Wen-Tao Li, Pan Chen
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{lis, wangw, liwt, chenp}@lamda.nju.edu.cn

Abstract

Multi-view representation learning attempts to learn a representation from multiple views and most existing methods are unsupervised. However, representation learned only from unlabeled data may not be discriminative enough for further applications (e.g., clustering and classification). For this reason, semi-supervised methods which could use unlabeled data along with the labeled data for multi-view representation learning need to be developed. Manifold information plays an important role in semi-supervised learning, but it has not been considered for multi-view representation learning. In this paper, we introduce the manifold smoothness into multi-view representation learning and propose MvDGAT which learns the representation and the intrinsic manifold simultaneously with graph attention network. Experiments conducted on real-world datasets reveal that our MvDGAT can achieve better performance than state-of-the-art methods.

Introduction

Multi-view data are ubiquitous in real-world applications, such as the text content and hyperlink in webpage classification (Du et al. 2013; Jing et al. 2017); the local patterns, the local shape descriptors, and the spatial-temporal contexts in object re-identification (Zhao et al. 2018; Zhou, Liu, and Shao 2018). The data from different views usually provide complementary information, and utilizing the information from multiple views together can improve the performance.

To deal with multi-view data, a baseline approach is simply concatenating the multiple views into one feature vector and adopting traditional single-view learning paradigms (Xu, Tao, and Xu 2013). However, this intrinsically goes against the nature of the distinct views and often leads to little improvement over single-view learning. Therefore, many multi-view learning methods are developed to achieve better performance (Blum and Mitchell 1998; Hotelling 1936; Sindhwani, Niyogi, and Belkin 2005). Generally, these multi-view learning methods can be categorized into three groups (Xu, Tao, and Xu 2013): 1) co-training (Blum and Mitchell 1998), 2) multiple kernel learning (Lanckriet et al. 2004) and 3) subspace learning (Hotelling 1936). Subspace learning assumes that the multiple views are generated from a latent subspace, and the goal is to recover the representation in the latent subspace (Yin, Huang, and Gao 2020; Wang et al. 2015; Zhen et al. 2019; Li et al. 2019a), e.g., Canonical Correlation Analysis (CCA) (Hotelling 1936) projects different views onto one common space by maximizing the correlation between pairwise views. In this way, subspace learning is also part of the multi-view representation learning.

Multi-view representation learning aims to learn a good representation that extracts heterogeneous useful information from each view for developing prediction models (Li, Yang, and Zhang 2018). In the past decades, there have been many methods developed to deal with multi-view representation problem. However, most existing methods focus on the unsupervised setting. For example, Kernel CCA (KCCA) (Akaho 2006) extends CCA to learn nonlinear projections, and Deep CCA (DCCA) (Andrew et al. 2013) is a parametric model that can scale to large datasets. Deep Canonically Correlated AutoEncoders (DCCAE) (Wang et al. 2015) consider the reconstruction error and correlation of two views simultaneously to achieve better performance. In general, purely unsupervised representation learning methods could not utilize information in labeled data, and thus may not be sufficiently discriminative in downstream tasks such as classification or clustering. To tackle this, Noroozi et al. (Noroozi et al. 2018) recently proposed the first deep semi-supervised representation learning model Multi-view Discriminative Neural Network (MDNN) for multi-view problem, which could be viewed as a combination of Deep Linear Discriminant Analysis (DLDA) (Dorfer, Kelz, and Widmer 2016) and DCCA (Andrew et al. 2013). However, MDNN did not incorporate the manifold information to maintain the smoothness over the data. How to capture the manifold structure is an important topic in semi-supervised learning and multi-view learning (Belkin, Niyogi, and Sindhwani 2006; Sindhwani and Rosenberg 2008; Wang and Zhou 2010; Zhou et al. 2003; Zhu 2005; Zhu, Ghahramani, and Lafferty 2003).

In this paper, we present a framework for semi-supervised multi-view representation learning and propose our Multi-view Discriminative Graph Attention Network (MvDGAT), which considers the empirical risk and view consistency in the objective function and embeds the manifold smoothness information with graph attention network to learn the

*Corresponding author.

intrinsic manifold of the original data distribution. Experiments conducted on the real-world datasets reveal that our MvDGAT can achieve better performance than state-of-the-art multi-view representation learning methods.

In the following, we first briefly review some related works. Then we present the preliminaries and our method. After the experiments compared with the baselines, we make a conclusion.

Related Works

Multi-view representation learning (Li, Yang, and Zhang 2018) can be categorized into alignment-based methods and fusion-based methods. The alignment-based methods usually assume the multiple views are generated from a common latent subspace and learn the transformation mapping from each view onto a common subspace. Canonical Correlation Analysis (CCA) (Hotelling 1936) is the first subspace learning method, which explores basis vectors for the examples in two views by mutually maximizing the correlation between the projections onto these basis vectors. Kernel CCA (KCCA) (Akaho 2006) extends CCA to learn nonlinear projections, but the nonlinearity is limited and the kernel trick makes it difficult to scale to large datasets. DNN-based models have been introduced into multi-view representation learning. Deep CCA (DCCA) (Andrew et al. 2013) is a parametric method that uses deep neural networks to learn nonlinear transformation; Deep Canonically Correlated AutoEncoders (DCCAE) (Wang et al. 2015) simultaneously consider the reconstruction objective function of two autoencoders and the correlation of paired feed-forward networks to enhance the performance. Nonetheless, CCA, KCCA, DCCA, and DCCAE are unsupervised representation learning methods, and they cannot exploit the supervised information during the training process. The representation they learned lacks class discriminativeness that is critical to the success of some tasks, i.e., classification and clustering. Recently, Noroozi et al. (Noroozi et al. 2018) proposed the first deep semi-supervised representation learning method Multi-view Discriminative Neural Network (MDNN) for multi-view problems. MDNN extends DCCA by considering both the correlation of all the data and the Linear Discriminant Analysis (LDA) objective function to maximize between-class separations and minimize within-class variations. In this way, MDNN simultaneously utilizes unlabeled and labeled data. However, MDNN does not incorporate the manifold regularizer to maintain the smoothness in each view.

The manifold information plays an important role in semi-supervised learning. The main assumption is that the distribution of the data follows an intrinsic manifold, which usually implies two examples that are close in the original space should be close after the transformation. In applications, it is usually assumed that a weight matrix $\mathbf{W}$ is the indication about the adjacency of examples and consequently implies the potential similarity of the learned representations. Previous graph-based methods mainly focused on classification task (Belkin, Niyogi, and Sindhwani 2006; Zhou et al. 2003; Zhu 2005; Zhu, Ghahramani, and Lafferty 2003). Among these methods, manifold regularization (Belkin, Niyogi, and Sindhwani 2006) was introduced to exploit the geometry of the marginal distribution. Recently, a line of works on Graph Neural Networks (GNNs) (Scarselli et al. 2009) provide another possibility for the utilization of graph-structured data. Graph Convolutional Network (GCN) (Kipf and Welling 2017) is a spectral convolution method that restricts spectral filters to operate in a 1-step neighborhood around each node, which can improve scalability and classification performance. An analysis of GCN (Li, Han, and Wu 2018) brought deeper insight and addressed that GCN is actually a special form of Laplacian smoothing and the vertices in the same cluster tend to be densely connected. A more recent attention mechanism was introduced into GNNs by Graph Attention Network (GAT) (Velickovic et al. 2018), and it can adaptively consider the significance of edges, which is effective when the observed graphs are inaccurate or noisy.

The manifold regularization was also extended to the multi-view setting (Sindhwani, Niyogi, and Belkin 2005; Sindhwani and Rosenberg 2008), which adds a Laplacian-style regularizer in co-regularization. Co-regularization shares similar intuition with co-training. Co-training was proposed by (Blum and Mitchell 1998), which learns two classifiers with initial labeled data on the two views respectively and lets them label unlabeled data for each other to augment the training data. The development of co-training is supported by a series of theoretical analyses (Balcan, Blum, and Yang 2004; Blum and Mansour 2017; Blum and Mitchell 1998; Wang and Zhou 2010). Among these analyses, Wang and Zhou (Wang and Zhou 2010) revealed that the process of co-training could be viewed as the label propagation process on a combinative graph, where the classifier trained in each view can utilize the cross-view structural information from the other view. This analysis also inspires us to utilize the neighborhood information from the combinative graph in multi-view learning.

Preliminaries

In the multi-view setting, examples are described with several disjoint sets of features. Most multi-view representation learning methods focus on two views (Akaho 2006; Andrew et al. 2013; Hotelling 1936; Noroozi et al. 2018; Wang et al. 2015), we also discuss the two-view setting here. For a two-view problem, we can denote the data matrices on the two views as $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively. For the v-th view ($v \in \{1, 2\}$), we have $\mathbf{X}_v = [\mathbf{x}_v^1, \mathbf{x}_v^2, ..., \mathbf{x}_v^l, \mathbf{x}_v^{l+1}, ..., \mathbf{x}_v^n]^\top$, where n is the number of data. We let l denote the number of labeled data and u denote the number of unlabeled data (i.e., $n = l+u$). For the labeled data, let $\mathbf{y}^l = [y^1, y^2, ..., y^l] \in \mathbb{R}^l$ be the labels. We denote $|C|$ as the number of classes and $C_v^i = \{\mathbf{x}_v^j | y^j = i\}$.

In modern representation learning (Bengio, Courville, and Vincent 2013), the representations are usually learned by neural networks. We denote $f_v(\mathbf{X}_v; \mathbf{\Theta}_v)$ as the neural network on the v-th view where $\mathbf{\Theta}_v = \{\mathbf{\Theta}_v^{(1)}, \mathbf{\Theta}_v^{(2)}, ..., \mathbf{\Theta}_v^{(K_v)}\}$ are its parameters and denote K_v as the number of layers in f_v. $\mathbf{H}_v^{(k)} = [\mathbf{h}_{v(1)}^{(k)}, \mathbf{h}_{v(2)}^{(k)}, ..., \mathbf{h}_{v(n)}^{(k)}]^\top \in \mathbb{R}^{n \times d_v^{(k)}}$ denotes the v-th view's representation in the k-th layer, $d_v^{(k)}$ is the di-

mension of the representation, and specifically $\mathbf{H}_v^{(0)} = \mathbf{X}_v$. For simplicity, we denote $\mathbf{Z}_v$ as the final representation $\mathbf{H}_v^{(K_v)}$. In multi-view representation learning, it is usually assumed that the learned representations of two views are similar or highly-correlated.

Our Method

As mentioned above, most existing multi-view representation methods focus on the unsupervised setting, which results in a lack of discriminativeness in the learned space. Adding a set of labeled data and learning in a semi-supervised way would help to tackle this problem. Previous works on semi-supervised learning have indicated that manifold information is important, e.g., label propagation (Zhu, Ghahramani, and Lafferty 2003) and manifold regularization (Belkin, Niyogi, and Sindhwani 2006).

To utilize manifold information, we would like to first introduce manifold regularization, which allows us to exploit the geometry structure. Let $\mathcal{H}$ be the hypothesis space and $f^* \in \mathcal{H}$ denote the hypothesis to learn, then manifold regularization is formulated as

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} V(\mathbf{X}, \mathbf{y}^l, f) + G(\mathbf{X}, \mathbf{W}, f),$$

where V is the loss function to evaluate the empirical loss on labeled data and G is the manifold regularizer. In real-world application, V is usually formed as

$$V(\mathbf{X}, \mathbf{y}^l, f) = \frac{1}{l} \sum_{i=1}^{l} (f(\mathbf{x}_i) - y_i)^2,$$

for classification tasks. This method can be generalized into the representation learning, in which V is usually based on Deep Linear Discriminant Analysis (DLDA) (Dorfer, Kelz, and Widmer 2016; Wu, Shen, and Van Den Hengel 2017) to increase the discriminativeness. We denote $f(\mathbf{X}; \mathbf{\Theta})$ as the neural network and $\mathbf{\Theta}$ are the parameters to learn, then DLDA is formed as

$$\mathbf{\Theta}^* = \operatorname*{argmax}_{\mathbf{\Theta}} \operatorname{Tr}\left(\frac{S_B(f(\mathbf{X}; \mathbf{\Theta}))}{S_W(f(\mathbf{X}; \mathbf{\Theta})) + r_1 \mathbf{I}}\right),$$

where $r_1 > 0$ is a positive constant to make the matrix positive definite. We denote $\mathbf{Z}$ as the final representation $f(\mathbf{X}; \mathbf{\Theta})$ for simplicity and denote $\mathbf{m}_v^i$ as the average vector for class i in the v-th view, then

$$S_W(\mathbf{Z}) = \frac{1}{l} \sum_{i=1}^{|C|} \sum_{\mathbf{z}^j \in \mathbf{C}^j} (\mathbf{z}^j - \mathbf{m}^i)(\mathbf{z}^j - \mathbf{m}^i)^\top$$

is the within-class scatter matrix, where $\mathbf{C}^j$ is the set of representation of labeled data from class j. We denote

$$S_B(\mathbf{Z}) = \frac{1}{2l^2} \sum_{i,j=1}^{|C|} l_i l_j (\mathbf{z}^j - \mathbf{m}^i)(\mathbf{z}^j - \mathbf{m}^i)^\top$$

as the between-class scatter matrix, where l_i is the number of labeled data from class i. The term $G(\mathbf{X}, \mathbf{W}, f)$ usually considers the loss of adjacency matrix $\mathbf{W}$ intrinsic manifold smoothness, which is usually formulated as

$$G(\mathbf{X}, \mathbf{W}, f) = \sum_{i,j=1}^{n} \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \mathbf{W}_{ij}. \quad (1)$$

$\mathbf{W}_{ij}$ are edge weights in the adjacency graph. $G(\mathbf{X}, \mathbf{W}, f)$ encourages similar examples to get closer after the transformation to maintain the manifold smoothness.

When the data have two views, the view consistency is usually calculated by CCA objective function. Then the multi-view learning can be formulated as

$$\begin{aligned}(\mathbf{\Theta}_1^*, \mathbf{\Theta}_2^*) = &\operatorname*{argmin}_{\mathbf{\Theta}_1, \mathbf{\Theta}_2} \sum_{v=1}^{2} V_v(\mathbf{y}_l, f_v(\cdot; \mathbf{\Theta}_v)) \\ &+ \lambda_G \sum_{v=1}^{2} G_v(\mathbf{X}_v, \mathbf{W}_v, f_v(\cdot; \mathbf{\Theta}_v)) \\ &+ \lambda_\Gamma \Gamma(\mathbf{X}_1, \mathbf{X}_2, f_1(\cdot; \mathbf{\Theta}_1), f_2(\cdot; \mathbf{\Theta}_2)).\end{aligned} \quad (2)$$

Besides the empirical loss and manifold regularizer, the term Γ evaluates the view consistency, e.g., $\Gamma(\mathbf{X}_1, \mathbf{X}_2, f_1, f_2) = (f_1(\mathbf{X}_1; \mathbf{\Theta}_1) - f_2(\mathbf{X}_2; \mathbf{\Theta}_2))^2$ is used in co-regularization. Under this setting, unlabeled data are proved to be helpful because it can reduce the hypothesis space (Balcan and Blum 2010). It is worth noting that though the forms of view-consistency and manifold regularizer are sometimes similar, they represent the cross-view and within-view constraints respectively. In representation learning, the term Γ usually considers the correlation of the representation for the two views, which is based on Deep Canonical Correlation Analysis (DCCA) (Andrew et al. 2013) that maps multiple views of data into a space with deep neural network where the paired examples are highly correlated. DCCA is formulated as

$$(\mathbf{\Theta}_1^*, \mathbf{\Theta}_2^*) = \operatorname*{argmax}_{\mathbf{\Theta}_1, \mathbf{\Theta}_2} \operatorname{corr}(f_1(\mathbf{X}_1; \mathbf{\Theta}_1), f_2(\mathbf{X}_2; \mathbf{\Theta}_2)).$$

To find $(\mathbf{\Theta}_1^*, \mathbf{\Theta}_2^*)$, we follow the gradient of the correlation objective function estimated on the training data. Let $\bar{\mathbf{Z}}_v = \mathbf{Z}_v - \frac{1}{n}\mathbf{Z}_v \mathbf{1}$ be the centered matrix and define $\mathbf{\Sigma}_{ij} = \frac{1}{n}\bar{\mathbf{Z}}_i \bar{\mathbf{Z}}_j^\top + r_2 \mathbf{I}$ where $r_2 > 0$ is a positive constant to make $\mathbf{\Sigma}_{ij}$ positive definite. The total correlation of the top k components of $\mathbf{Z}_1$ and $\mathbf{Z}_2$ is the sum of the top k singular values of the matrix $\mathbf{T} = \mathbf{\Sigma}_{11}^{-1/2} \mathbf{\Sigma}_{12} \mathbf{\Sigma}_{22}^{-1/2}$. Then we have $\operatorname{corr}(\mathbf{Z}_1, \mathbf{Z}_2) = \operatorname{Tr}(\mathbf{T}^\top \mathbf{T})^{1/2}$ as the sum of the singular values and the goal is to maximize it.

A problem of the above manifold regularization is that we usually cannot get in touch with the intrinsic manifold smoothness $\mathbf{W}_v$ but an inaccurate observation $\tilde{\mathbf{W}}_v (v = 1, 2)$. In this paper, we solve this problem by using graph neural networks and attention mechanism. We apply the shared attentional mechanism $a : \mathbb{R}^{d_v^{(k)}} \times \mathbb{R}^{d_v^{(k)}} \to \mathbb{R}$ to calculate the attention coefficients

$$\alpha_{ij} = a(\mathbf{\Theta}_v^{(k)} \mathbf{h}_{v(i)}^{(k)}, \mathbf{\Theta}_v^{(k)} \mathbf{h}_{v(j)}^{(k)})$$

that indicate the importance of node j's representation to node i. The function a is usually formed as a single-layer

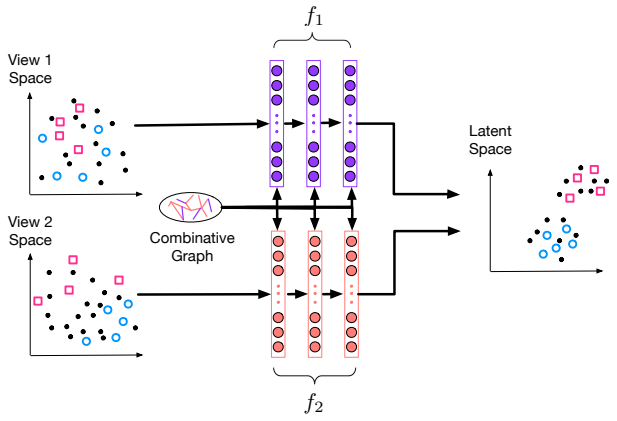


Figure 1: MvDGAT structure for two views. The hollow circles and squares are labeled examples with different labels. The black circles are unlabeled data. On the left side of the figure, the examples are represented in their original input space in the two views, respectively. After passing them through MvDGAT, the latent feature space is obtained, which is depicted on the right side of the figure.

neural network with LeakyReLU as nonlinearity. For the sake of simplicity, we denote $\mathcal{N}_i$ as the neighborhood set of the i-th example. To make the coefficients easily comparable across different nodes, a normalization process is applied across all choices of node j using the softmax function

$$\tilde{\mathbf{W}}_{ij} = \text{softmax}(\alpha_{ij}) = \frac{\exp(\alpha_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(\alpha_{ik})}.$$

Then we need to get the neighborhood set $\mathcal{N}_i$. We define $\mathcal{N}_i = \{j|\mathbf{W}_{c(ij)} \neq 0\}$ inspired by the combinative graph $\mathbf{W}_c$ used in previous multi-view learning analysis (Wang and Zhou 2010). The regularizer in Equation (2) can be simplified into the form with a combinative graph. Considering that the representations from multiple views are projected into a common subspace where they are highly correlated, we assume that $f_1(\mathbf{x}_1^i) \approx \lambda_f f_2(\mathbf{x}_2^i)$, and let $\lambda_f f^i = f_1(\mathbf{x}_1^i) \approx \lambda_f f_2(\mathbf{x}_2^i)$. The manifold regularizer with observed $\tilde{\mathbf{W}}_1$ and $\tilde{\mathbf{W}}_2$ could be written as

$$\begin{aligned}&\sum_{i,j=1}^{n}\left(\left\|f^i - f^j\right\|\tilde{\mathbf{W}}_{1(ij)} + \left\|\lambda_f(f^i - f^j)\right\|\tilde{\mathbf{W}}_{2(ij)}\right)\\ =&\sum_{i,j=1}^{n}\left\|f^i - f^j\right\|(\tilde{\mathbf{W}}_{1(ij)} + \lambda_f\tilde{\mathbf{W}}_{2(ij)}),\end{aligned}$$

where the combinative graph $\tilde{\mathbf{W}}_c = \tilde{\mathbf{W}}_1 + \lambda_f\tilde{\mathbf{W}}_2$.

In practice, to stabilize the learning process of self-attention, we also use the multi-head attention mechanism like GAT (Velickovic et al. 2018), then the feedforward process of each layer is formulated as

$$\mathbf{h}_{v(i)}^{(k+1)} = \Big\Vert_{p=1}^{P}\sigma\Big(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(p)}\mathbf{\Theta}_v^{(k,p)}\mathbf{h}_{v(j)}^{(k)}\Big),$$

where $\|$ represents the concatenation and P is the number of heads. For the output layer, we employ averaging and delay applying the final nonlinearity as

$$\mathbf{h}_{v(i)}^{(K_v)} = \sigma\Big(\frac{1}{P}\sum_{p=1}^{P}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(p)}\mathbf{\Theta}_v^{(K_v-1,p)}\mathbf{h}_{v(j)}^{(K_v-1)}\Big).$$

We call the method Multi-view Discriminative Graph Attention Network (MvDGAT). Instead of using manifold regularizer like Equation (1), we consider to use state-of-the-art graph neural networks to incorporate the manifold information, and the loss function is formulated as

$$\begin{aligned}\mathcal{L}_{\text{MvDGAT}} = &-\sum_{v=1}^{2}\text{Tr}\Big(\frac{S_B(f_v(\mathbf{X}_v))}{S_W(f_v(\mathbf{X}_v)) + r_1\mathbf{I}}\Big)\\ &-\lambda_\Gamma\,\text{corr}(f_1(\mathbf{X}_1), f_2(\mathbf{X}_2)).\end{aligned} \tag{3}$$

Let $f_v(\mathbf{X}_v, \mathbf{W}_c; \mathbf{\Theta}_v, \alpha_v)$ denote the GAT model trained with combinative graph $\mathbf{W}_c$ in the v-th view. We summarize the process of our method in Algorithm 1.

Algorithm 1 MvDGAT

Input: Two views $\mathbf{X}_1, \mathbf{X}_2$, adjacency matrices $\tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2$, and $\mathbf{y}^l$.
Parameter: T_1, T_2, λ_f and η.

```
Construct the combinative graph W̃_c = W̃_1 + λ_f W̃_2;
for v = 1 to 2 do
    for t = 1 to T_v do
        Forward X_v on f_v(·; Θ_v, α_v) with W̃_c;
        Calculate the gradient of f_v with the loss function in Equation (3);
        Update α_v and Θ_v in f_v with learning rate η;
    end for
end for
```

Output: $f_1(\mathbf{X}_1)$ and $f_2(\mathbf{X}_2)$.

Experiments

In order to evaluate our MvDGAT, we test its performance for clustering and classification tasks compared with other multi-view representation learning methods. We also visualize the learned representation to show that MvDGAT can learn better representation by exploiting the manifold smoothness.

Dataset	n	$\lvert C\rvert$	d_v
Course	1051	2	3447, 427
Ads12	983	2	457, 495
Ads13	983	2	457, 472
Ads23	983	2	495, 472
FOX	1523	4	996, 5477
CNN	2107	7	996, 7989
NoisyMNIST	70000	10	784, 784

Table 1: Statistics of the seven datasets. n represents the number of examples, $|C|$ is the number of classes, and d_v is the number of attributes of each view.

	Course	*Ads12*	*Ads13*	*Ads23*	*FOX*	*CNN*	*NoisyMNIST*
CCA	0.851	0.860	0.865	0.865	0.625	0.328	0.668
KCCA	0.781	0.860	0.865	0.865	0.617	0.336	-
DCCA	0.914	0.860	0.866	0.860	0.612	0.262	0.632
DCCAE	0.876	0.860	0.880	0.864	0.599	0.269	0.476
MDNN(5%)	0.782	0.862	0.862	0.873	0.505	0.289	0.655
MvDGAT(5%)	**0.924**	**0.862**	**0.870**	**0.876**	**0.665**	**0.481**	**0.844**
MDNN(10%)	0.794	0.861	**0.876**	0.875	0.542	0.308	0.664
MvDGAT(10%)	**0.928**	**0.866**	0.874	**0.901**	**0.716**	**0.490**	**0.866**
MDNN(15%)	0.815	0.868	0.883	0.879	0.543	0.366	0.679
MvDGAT(15%)	**0.931**	**0.871**	**0.900**	**0.889**	**0.718**	**0.484**	**0.893**

Table 2: Clustering results in terms of Purity on seven datasets. Results are averaged over 10 trials. ($\gamma\%$) represents the performance of trained representation by using $\gamma\%$ labeled data. - represents that the KCCA cannot scale to large datasets.

	Course	*Ads12*	*Ads13*	*Ads23*	*FOX*	*CNN*	*NoisyMNIST*
CCA	0.379	0.011	0.088	0.033	0.232	0.075	0.550
KCCA	0.125	0.015	0.088	0.088	0.276	0.112	-
DCCA	0.484	0.029	0.098	0.031	0.212	0.040	0.477
DCCAE	0.486	0.024	0.106	0.088	0.171	0.049	0.420
MDNN(5%)	0.473	0.047	0.131	0.104	0.102	0.079	0.563
MvDGAT(5%)	**0.553**	**0.077**	**0.142**	**0.157**	**0.471**	**0.295**	**0.579**
MDNN(10%)	0.530	0.053	0.163	0.121	0.115	0.087	**0.603**
MvDGAT(10%)	**0.560**	**0.088**	**0.167**	**0.189**	**0.479**	**0.295**	0.594
MDNN(15%)	0.588	0.062	0.174	0.157	0.118	0.117	0.632
MvDGAT(15%)	**0.591**	**0.090**	**0.308**	**0.229**	**0.482**	**0.300**	**0.634**

Table 3: Clustering results in terms of NMI on seven datasets. Results are averaged over 10 trials. ($\gamma\%$) represents the performance of trained representation by using $\gamma\%$ labeled data. - represents that the KCCA cannot scale to large datasets.

Datasets

The experiments on several real-world multi-view datasets are conducted to demonstrate the effectiveness and superiority of our proposed method. A summary of the datasets is presented in Table 1.

Course dataset is a webpage classification dataset, which contains 1,051 home pages collected from web sites of Computer Science departments of several universities, and has two views. These pages are manually labeled as course or non-course, each with a full-text view and an in-links view.

Advertisement dataset contains 983 images and is described in 5 views, i.e., the text information like image properties, the image caption and words occurring in the image source URL, the words occurring in the affiliated web page URL and the words occurring in the image anchor URL. Each example describes an image on the web, and the images are manually labeled as ads or non-ads. By using the texts from different views, we create datasets named *Ads12*, *Ads13* and *Ads23*.

FOX and CNN datasets are crawled from FOX and CNN web news. The category information extracted from their RSS feeds is considered as their class label. Each instance is represented in two views: the text view and image view. Titles, abstracts, and text body contents are considered as the text view data (view 1), and the image associated with the article is the image view (view 2). These datasets are named *FOX* and *CNN* respectively.

MNIST dataset is generated from the MNIST dataset and we adopt the setting used in (Wang et al. 2015). Specifically, we use the original dataset as view 1 and randomly select within-class images with additive noise as view 2. Each image consists of 28×28 grayscale digits. Thus, we obtain a two-view dataset named *NoisyMNIST* consisting of 70,000 samples.

Settings

The performance of MvDGAT is compared with several multi-view representation learning methods, i.e., CCA (Hotelling 1936), KCCA (Akaho 2006; Sun, Dong, and Liu 2020), DCCA (Andrew et al. 2013; Zhen et al. 2019), DCCAE (Wang et al. 2015; Hu et al. 2019) and MDNN (Noroozi et al. 2018). For each method, 10 trials are performed and the average performance is reported. The hyperparameters are chosen according to the validation per-

γ		*Course*	*Ads12*	*Ads13*	*Ads23*	*FOX*	*CNN*	*NoisyMNIST*
5%	CCA	0.874	0.860	0.867	0.862	0.612	0.285	0.760
	KCCA	0.785	0.860	0.860	0.860	0.623	0.289	-
	DCCA	0.899	0.864	0.872	0.851	0.631	0.280	0.785
	DCCAE	0.912	0.842	0.890	0.866	0.627	0.273	0.812
	MDNN	0.903	0.882	0.906	0.924	0.665	0.304	0.852
	MvDGAT	**0.940**	**0.886**	**0.909**	**0.941**	**0.721**	**0.609**	**0.904**
10%	CCA	0.930	0.859	0.880	0.865	0.671	0.306	0.769
	KCCA	0.930	0.859	0.919	0.864	0.682	0.303	-
	DCCA	0.923	0.860	0.903	0.862	0.644	0.309	0.793
	DCCAE	0.924	0.857	0.891	0.863	0.633	0.311	0.799
	MDNN	0.933	0.897	0.934	0.931	0.670	0.328	0.871
	MvDGAT	**0.951**	**0.904**	**0.948**	**0.949**	**0.762**	**0.652**	**0.922**
15%	CCA	0.928	0.890	0.881	0.889	0.686	0.316	0.777
	KCCA	0.896	0.897	0.873	0.883	0.697	0.308	-
	DCCA	0.932	0.909	0.904	0.883	0.647	0.316	0.798
	DCCAE	0.927	0.909	0.891	0.892	0.644	0.314	0.808
	MDNN	0.941	0.921	**0.950**	0.926	0.673	0.354	0.901
	MvDGAT	**0.953**	**0.929**	0.945	**0.936**	**0.776**	**0.670**	**0.947**

Table 4: Accuracy of the methods with γ ($\gamma = 5\%, 10\%, 15\%$) labeled data. Results are averaged over 10 trials. - represents that KCCA cannot scale to the large datasets.

formance in the first trial and then fixed.

For MvDGAT, we first construct k-nearest-neighbor graph for each view with $\exp(-\frac{d(x_{v(i)},x_{v(j)})}{\sigma^2})$ for different distance metrics, $k \in \{1,3,5,7,9\}$, $d(x_{v(i)}, x_{v(j)})$ is set to be Euclidean distance or cosine distance, and $\sigma \in \{10^{-2}, 10^{-1}, 1\}$, $v = 1, 2$. For each dataset, we randomly sample 10% data as the validation set, randomly sample γ ($\gamma = 5\%, 10\%, 15\%$) of the remaining data as the labeled data, and use the rest data as the unlabeled data. In the experiments, each transformation $f_v(\mathbf{X}_v)$ in MvDGAT consists of two hidden layers and an output layer, which is similar to the setting in (Velickovic et al. 2018). The final dimension of each output layer is selected from $\{5, 10, 20, 50, 100\}$ according to the performance of the first view on the validation set. In the training process, we use dropout rate $p = 0.3$ and use ReLU as the activation function in each graph attention layer, and the softmax activation function is used in this output layer. The size of the hidden layer in each view varies according to the dimension of feature in the view, i.e., 128 and 128 for *Course*, *Ads12*, *Ads13*, *Ads23*, *FOX*, *CNN*, 256 and 256 for *NoisyMNIST*. f_1 and f_2 are trained for a maximum of 200 epochs using Adam (Kingma and Ba 2015) and early stopping with a window size of 5, i.e., we stop the training process when the validation performance does not increase for 5 consecutive epochs. For *Course*, *Ads12*, *Ads13*, *Ads23*, *FOX*, *CNN*, the whole datasets are used to evaluate the loss function. For *NoisyMNIST*, the dataset is split into 10 fold according to the proportion of each class.

For the baselines, we consider the regularization parameters chosen from $\{0.1, 1, 10\}$ in CCA and KCCA. Especially, we use RBF kernel ($\sigma \in \{0.1, 1, 10\}$), polynomial kernel ($d \in \{2, 3, 4\}$) and linear kernel in KCCA. The dimension of the representations generated by CCA and KCCA is selected from $\{5, 10, 15\}$ for all datasets. For DCCA, the projection network has one hidden layer, whose size is chosen from $\{2^5, 2^6, 2^7, 2^8\}$. For DCCAE, both encoder and decoder network also have one hidden layer and the setting of the hidden layer is the same as that in DCCA. The dimension of the representations generated by DCCA and DCCAE will also vary in $\{5, 10, 15\}$. All the parameters and kernel types mentioned above are chosen according to the performance on the validation set.

Clustering

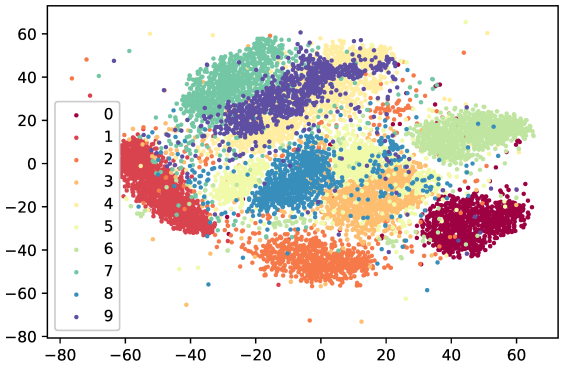
Clustering aims to group a set of data points, the data in the same cluster are as similar as possible, while the data in the different clusters are dissimilar to each other. We measure the separation in the learned feature space by clustering the projected view 1 inputs into several clusters and evaluate how well the clusters agree with the ground-truth labels by using *k-means* algorithm. We learn representation with different labeled and unlabeled data for 10 trials. For MvDGAT, we train MvDGAT representation with $\gamma\%$ ($\gamma = 5, 10, 15$) labeled data. For *k-means*, different initialization methods are considered, i.e., k-means++ and random. We measure clustering performance with two criteria, i.e., purity score (Purity) and Normalized Mutual Information score (NMI) (Li et al. 2019b; Nie, Cai, and Li 2017). Purity is a simple and transparent evaluation measure, but cannot be used to trade off the quality of the clustering against the number of clusters. NMI can be information-theoretically interpreted, which measures the normalized mutual information between the distribution of clustering labels and that of true labels. The clustering results are shown

in Tables 2 and 3. It can be found that MvDGAT can achieve the best performance on most datasets and its performance is increasing with more labeled data.

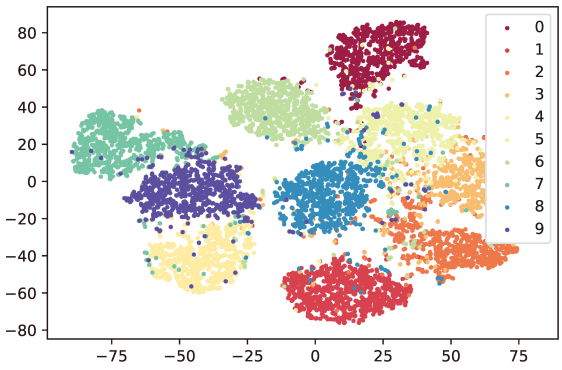
Classification

Given the representation, one would expect that the factors are related to each other through simple, typically linear dependencies, so simple machine learning algorithms could achieve good performance, e.g. linear classifier (Liu et al. 2018). We implement the linear classifier with one-versus-one linear SVMs on the projected labeled data, and evaluate the performance on the projected unlabeled data (using the validation set for selecting the hyperparameters of SVMs, and the regularization parameter C is selected from $\{0.001, 0.01, 0.1, 1, 10\}$).

The evaluated performances on the representation of all trials are averaged for each algorithm. The results of the first view are shown in Table 4, which shows that MvDGAT can achieve the best performance on most datasets and its performance is increasing with more labeled data.



(a) Original Representation



(b) MvDGAT Representation

Figure 2: Visualization of NoisyMNIST dataset and its learned MvDGAT representation.

Subspace Analysis

We also compare the representation learned by MvDGAT with the original feature space on *NoisyMNIST* dataset. We visualize the representation of the dataset with 15% labeled data, and 1000 instances of the samples are randomly selected to be visualized. These instances are visualized by using a dimension reduction algorithm called t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton 2008), which is a representation learning method widely used for visualizing features in a low-dimensional space. It learns mappings from the given feature space to a new space in which the similarity of samples is preserved as much as possible. We reduce the dimension of representation to be 2 by using t-SNE and then visualize it.

For *NoisyMNIST*, its original representation is shown in Figure 2a and the learned representation is shown in Figure 2b. We can see that the different classes are drawn apart after the transformation, which further demonstrates the usefulness of our MvDGAT for learning a good representation.

Conclusion

In this paper, we present a deep neural network model MvDGAT for multi-view representation learning, which can utilize labeled and unlabeled data simultaneously to enhance the discriminativeness of the learned representation. Experiments conducted on both the synthetic and real-world datasets reveal that our MvDGAT can achieve better performance than state-of-the-art representation learning methods.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (2018AAA0101100), the National Science Foundation of China (61673202, 61921006), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Akaho, S. 2006. A kernel method for canonical correlation analysis. *CoRR* .

Andrew, G.; Arora, R.; Bilmes, J. A.; and Livescu, K. 2013. Deep Canonical Correlation Analysis. In *ICML*.

Balcan, M.; and Blum, A. 2010. A discriminative model for semi-supervised learning. *Journal of the ACM* 57(3): 19:1–19:46.

Balcan, M.; Blum, A.; and Yang, K. 2004. Co-Training and Expansion: Towards Bridging Theory and Practice. In *NIPS*.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research* 7: 2399–2434.

Bengio, Y.; Courville, A. C.; and Vincent, P. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8): 1798–1828.

Blum, A.; and Mansour, Y. 2017. Efficient Co-Training of Linear Separators under Weak Dependence. In *COLT*.

Blum, A.; and Mitchell, T. M. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *COLT*.

Dorfer, M.; Kelz, R.; and Widmer, G. 2016. Deep Linear Discriminant Analysis. In *ICLR*.

Du, Y.; Li, Q.; Cai, Z.; and Guan, X. 2013. Multi-view semi-supervised web image classification via co-graph. *Neurocomputing* 122: 430–440.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(3/4): 321–377.

Hu, P.; Peng, D.; Sang, Y.; and Xiang, Y. 2019. Multi-view linear discriminant analysis network. *IEEE Transactions on Image Processing* 28(11): 5352–5365.

Jing, X.; Wu, F.; Dong, X.; Shan, S.; and Chen, S. 2017. Semi-Supervised Multi-View Correlation Feature Learning with Application to Webpage Classification. In *AAAI*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Lanckriet, G. R. G.; Cristianini, N.; Bartlett, P. L.; Ghaoui, L. E.; and Jordan, M. I. 2004. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research* 5: 27–72.

Li, Q.; Han, Z.; and Wu, X. 2018. Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. In *AAAI*.

Li, R.; Zhang, C.; Hu, Q.; Zhu, P.; and Wang, Z. 2019a. Flexible Multi-View Representation Learning for Subspace Clustering. In *IJCAI*.

Li, Y.; Yang, M.; and Zhang, Z. 2018. A Survey of Multi-View Representation Learning. *IEEE Transactions on Knowledge and Data Engineering* 31(10): 1863–1883.

Li, Z.; Wang, Q.; Tao, Z.; Gao, Q.; and Yang, Z. 2019b. Deep Adversarial Multi-view Clustering Network. In *IJCAI*.

Liu, K.; Wang, H.; Nie, F.; and Zhang, H. 2018. Learning multi-instance enriched image representations via non-greedy ratio maximization of the L1-norm distances. In *CVPR*.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9: 2579–2605.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*.

Noroozi, V.; Bahaadini, S.; Zheng, L.; Xie, S.; Shao, W.; and Yu, P. S. 2018. Semi-supervised Deep Representation Learning for Multi-View Problems. In *BigData*.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Transaction on Neural Networks* 20(1): 61–80.

Sindhwani, V.; Niyogi, P.; and Belkin, M. 2005. A co-regularization approach to semi-supervised learning with multiple views. In *ICML workshop*.

Sindhwani, V.; and Rosenberg, D. S. 2008. An RKHS for multi-view learning and manifold co-regularization. In *ICML*.

Sun, S.; Dong, W.; and Liu, Q. 2020. Multi-view representation learning with deep gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Wang, W.; Arora, R.; Livescu, K.; and Bilmes, J. A. 2015. On Deep Multi-View Representation Learning. In *ICML*.

Wang, W.; and Zhou, Z. 2010. A New Analysis of Co-Training. In *ICML*.

Wu, L.; Shen, C.; and Van Den Hengel, A. 2017. Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition* 65: 238–250.

Xu, C.; Tao, D.; and Xu, C. 2013. A Survey on Multi-view Learning. *CoRR* .

Yin, M.; Huang, W.; and Gao, J. 2020. Shared Generative Latent Representation Learning for Multi-View Clustering. In *AAAI*.

Zhao, C.; Wang, X.; Miao, D.; Wang, H.; Zheng, W.; Xu, Y.; and Zhang, D. 2018. Maximal granularity structure and generalized multi-view discriminant analysis for person re-identification. *Pattern Recognition* 79: 79–96.

Zhen, L.; Hu, P.; Wang, X.; and Peng, D. 2019. Deep supervised cross-modal retrieval. In *CVPR*.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2003. Learning with Local and Global Consistency. In *NIPS*.

Zhou, Y.; Liu, L.; and Shao, L. 2018. Vehicle Re-Identification by Deep Hidden Multi-View Inference. *IEEE Transactions on Image Processing* 27(7): 3275–3287.

Zhu, X. 2005. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML*.