MEGAN: A Generative Adversarial Network for Multi-View Network Embedding

Yiwei Sun[1] , **Suhang Wang**[2] , **Tsung-Yu Hsieh**[1] , **Xianfeng Tang**[2] , **Vasant Honavar**[1 2]
[1]Department of Computer Science and Engineering, The Pennsylvania State University, USA
[2]College of Information Sciences and Technology, The Pennsylvania State University, USA
{yus162,szw494,tuh45,xut10,vuh14}@psu.edu

Abstract

Data from many real-world applications can be naturally represented by *multi-view* networks where the different views encode different types of relationships (e.g., friendship, shared interests in music, etc.) between real-world individuals or entities. There is an urgent need for methods to obtain low-dimensional, information preserving and typically nonlinear embeddings of such multi-view networks. However, most of the work on multi-view learning focuses on data that lack a network structure, and most of the work on network embeddings has focused primarily on *single-view* networks. Against this background, we consider the multi-view network representation learning problem, i.e., the problem of constructing low-dimensional information preserving embeddings of *multi-view* networks. Specifically, we investigate a novel Generative Adversarial Network (GAN) framework for Multi-View Network Embedding, namely MEGAN, aimed at preserving the information from the individual network views, while accounting for connectivity across (and hence complementarity of and correlations between) different views. The results of our experiments on two real-world multi-view data sets show that the embeddings obtained using MEGAN outperform the state-of-the-art methods on node classification, link prediction and visualization tasks.

1 Introduction

Network embedding or network representation learning, which aims to learn low-dimensional, information preserving and typically non-linear representations of networks, has been shown to be useful in many tasks, such as link prediction [Perozzi *et al.*, 2014], community detection [He *et al.*, 2015; Cavallari *et al.*, 2017] and node classification [Wang *et al.*, 2016b]. A variety of network embedding schemes have been proposed in the literature [Ou *et al.*, 2016; Wang *et al.*, 2016a; Grover and Leskovec, 2016; Hamilton *et al.*, 2017; Yu *et al.*, 2018; Zhang *et al.*, 2018b]. However, most of these methods focus on *single-view* networks, i.e., networks with only one type relationships between nodes.

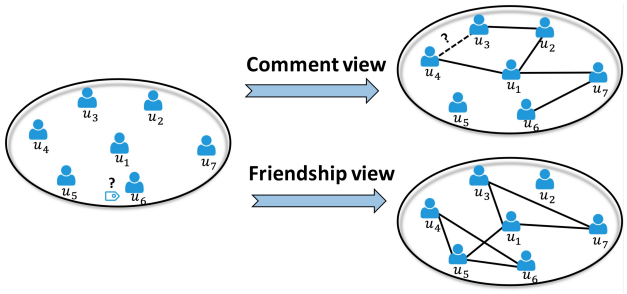


Figure 1: The toy multi-view network containing 7 users (nodes) and comprised of comment view and friendship view

However, data from many real-world applications are best represented by *multi-view* networks [Kivelä *et al.*, 2014], where the nodes are linked by *multiple* types of relations. For example, in Flickr, two users can have multiple relations such as friendship and communicative interactions (e.g., public comments); In Facebook, two users can share friendship, enrollment in the same university, or likes and dislikes. For example, Fig.1 shows a 2-view network comprised of the friendship view and the comment view. Such networks present multi-view counterparts of problems considered in the single-view setting, such as, node classification (e.g., labelling the user to the specific categories of interests(tag of u_6)) and link prediction (e.g., predicting a future link, say between u_3 and u_4). Previous work [Wang *et al.*, 2015; Shi *et al.*, 2016] has shown that taking advantage of the complementary and synergy information supplied by the different views can lead to improved performance. Hence, there is a growing interest in multi-view network representation learning (MVNRL) methods that effectively integrate information from disparate views [Shi *et al.*, 2016; Qu *et al.*, 2017; Zhang *et al.*, 2018a; Huang *et al.*, 2018; Ma *et al.*, 2019] . However, there is significant room for improving both our understanding of the theoretical underpinnings as well as practical methods on real-world applications.

Generative adversarial networks (GAN) [Goodfellow *et al.*, 2014], which have several attractive properties, including robustness in the face adversarial data samples, noise in the data, etc., have been shown to be especially effective for modeling the underlying complex data distributions by

discovering latent representations. A GAN consists of two sub-networks, a *generator* which is trained to generate adversarial data samples by learning a mapping from a latent space to a data distribution of interest, and a discriminator that is trained to discriminate between data samples drawn from the true data distribution and the adversarial samples produced by the generator. Recent work has demonstrated that GAN can be used to effectively perform network representation learning in the single view setting [Wang *et al.*, 2018; Bojchevski *et al.*, 2018]. Against this background, it is natural to consider whether such approaches can be extended to the setting of multi-view networks. However, there has been little work along this direction.

Effective approaches to multi-view network embedding using GAN have to overcome the key challenge that is absent in the single-view setting: in the single-view setting, the generator, in order to produce adversarial samples, needs to model only the connectivity (presence or absence of a link) between pairs of nodes, in multi-view networks, how to model not only the connectivity within each of the different, but also the complex correlations between views. The key contributions of the paper are as follows:

- We propose the Multi-view network Embedding GAN (MEGAN), a novel GAN framework for learning a low dimensional, typically non-linear and information preserving embedding of a given multi-view network.
- Specifically, we show how to design a generator that can effectively produce adversarial data samples in the multi-view setting.
- We describe an unsupervised learning algorithm for training MEGAN from multi-view network data.
- Through experiments with real-world multi-view network data, we show that MEGAN outperforms other state-of-the-art methods for MVNRL when the learned representations are used for node labeling, link prediction, and visualization.

2 Related Work

2.1 Single-view Network Embedding

Single-view network embedding methods seek to learn a low-dimensional, often non-linear and information preserving embedding of a single-view network for node classification and link prediction tasks. There is a growing literature on single-view network embedding methods [Perozzi *et al.*, 2014; Tang *et al.*, 2015; Ou *et al.*, 2016; Wang *et al.*, 2016a; Grover and Leskovec, 2016; Hamilton *et al.*, 2017; Zhang *et al.*, 2018b]. For example, in [Belkin and Niyogi, 2001], spectral analysis is performed on Laplacian matrix and the top-k eigenvectors are used as the representations of network nodes. DeepWalk [Perozzi *et al.*, 2014] introduces the idea of Skip-gram, a word representation model in NLP, to learn node representations from random-walk sequences. SDNE [Wang *et al.*, 2016a] uses deep neural networks to preserve the neighbors structure proximity in network embedding. GraphSAGE [Hamilton *et al.*, 2017] generates embeddings by recursively sampling and aggregating features from a node's local neighborhood. GraphGAN [Wang *et al.*, 2018], which extends GAN [Goodfellow *et al.*, 2014] to work with networks (as opposed to feature vectors) has shown promising results on the network embedding task. It learns a generator to approximate the node connectivity distribution and a discriminator to differentiate the "fake" nodes (adversarial samples) and the nodes sampled from the true data distribution. We differ from their work in mainly two respects: we focus on the complex multi-view network; we develop connectivity discriminator and generator with novel sampling strategy.

2.2 Multi-view Network Embedding

Motivated by real-world applications, there is a growing interest in methods for learning embeddings of multi-view networks. Such methods have to effectively integrate information from the individual network views while exploiting complementarity of information supplied by the different views. To capture the associations across different views, [Ma *et al.*, 2017] utilize a tensor to model the multi-view network and factorize tensor to obtain a low-dimensional embedding; MVE [Qu *et al.*, 2017] combine information from multiple views using a weighted voting scheme; MNE [Zhang *et al.*, 2018a] use a latent space to integrate information across multiple views. In contrast, MEGAN proposed in this paper implicitly models the associations between views in a latent space and employs a generator that effectively integrates information about pair-wise links between nodes across all of the views.

3 Multi-View Network Embedding GAN

In what follows, we define the multi-view network embedding problem before proceeding to describe the key components of MEGAN, our proposed solution to multi-view network embedding.

3.1 Multi-View Network Embedding

A multi-view network is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ denotes the set of nodes and $\mathcal{E} = \{\mathcal{E}^{(1)}, \mathcal{E}^{(2)}, \ldots, \mathcal{E}^{(k)}\}$ describes the edge sets that encode k different relation types (views). For a given relation type l, $\mathcal{E}^{(l)} = \{e_{ij}^{(l)}\}_{i,j=1}^{n}$ specifies the presence of the corresponding relation between node v_i and node v_j. Thus, $e_{ij}^{(l)} = 1$ indicates that v_i and v_j has a link for relation l and 0 otherwise. We use $\mathcal{K}_{ij} = (e_{ij}^{(1)}, \ldots, e_{ij}^{(k)})$ to denote the connectivity of (v_i, v_j) in multi-view topology. The goal of multi-view network representation learning is to learn $\mathbf{X} \in \mathbb{R}^{n \times d}$, a low-dimensional embedding of $\mathcal{G}$, where $d \ll n$ is the latent dimension.

3.2 Multi-view Generative Adversarial Network

Unlike a GAN designed to perform single-view network embedding, which needs to model only the connectivity among nodes within a single view, the MEGAN needs to capture the connectivity among nodes within each view as well as the correlations between views. To achieve this, given the *real* pair of nodes $(v_i, v_j) \sim p_{data}$, MEGAN consists of two modules: a **generator** which generates (or chooses) a *fake* node v_c with the connectivity pattern $\mathcal{K}_{ic}$ between v_i and v_c being sufficiently similar to that of the real pair of node; and a **discriminator** which is trained to distinguish between the real

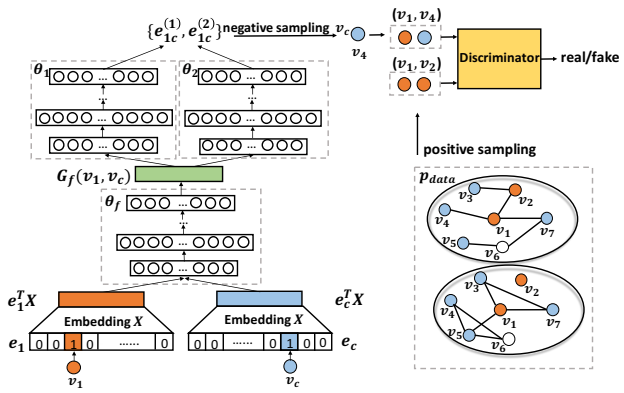


Figure 2: The architecture of MEGAN framework. The pair of the orange nodes (v_1, v_2) denotes *real* pair of nodes. For each blue nodes v_c, the generator learns its connectivity $\{e_{ic}^{(1)}, e_{ic}^{(2)}\}$ and generates *fake* pair of nodes which could fool the discriminator with highest probability. In the example, (v_1, v_4) is selected to fool the discriminator.

pair of nodes (v_i, v_j) and *fake* pair of node (v_i, v_c). Figure 2 illustrates the architecture of MEGAN. For the pair of nodes (v_1, v_2), the generator produces a *fake* node v_4 forming the pair of nodes (v_1, v_4) to fool the discriminator and the discriminator is trained to differentiate if a pair of nodes input to it is *real* or *fake*. With the minmax game between D and G, we will show that upon convergence, we are able to learn embeddings that G can use to generate the multi-view network. In such situation, the learned embeddings is able to capture the connectivity among nodes within each views and the correlations between views. Next, we introduce the details of G, D and an efficient sampling strategy.

Multi-view Generator

Recall that goals of multi-view generator G are to **(i)** generate the multi-view connectivity of fake nodes that fools the discriminator D; and **(ii)** learn an embedding that captures the multi-view network topology. To ensure that it achieves the goals, we design the multi-view generator consisting of two components: one fusing generator G_f and k connectivity generators $G^{(l)}$. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the network embedding matrix we want to learn, then the representation of v_i can be obtained as $\mathbf{e}_i^T \mathbf{X}$, where $\mathbf{e}_i$ is the one-hot representation with i-th value as 1. The fusing generator G_f firstly fuses the representation of v_i and v_j, aiming to capture the correlation between v_i and v_j. We use $G_f(v_i, v_j; \mathbf{X}, \theta_f)$ to denote the fused representation where θ_f is its parameters. Then, the fused representation is used to calculate the probability of $e_{ij}^{(l)} = 1$ for $l = 1, \ldots, k$. This can be formally written as:

$$G(e_{ij}^{(1)}, \ldots, e_{ij}^{(k)} | v_i, v_j, \theta_G) = \prod_{l=1}^{k} G^{(l)}(e_{ij}^{(l)} | G_f(v_i, v_j; \mathbf{X}, \theta_f); \theta^{(l)}) \tag{1}$$

where $G^{(l)}$ is the connectivity generator for generating the connectivity in between (v_i, v_j) in view l given the fused representation and $\theta^{(l)}$ is the parameter for $G^{(l)}$. $\theta_G = \{\mathbf{X}, \theta_f, \theta^{(l), l=1,\ldots,k}\}$ is the parameter for the multi-view generator G. Because each $G^{(l)}$ of k generators are independent given the fused representation, the corresponding parameter $\theta^{(l)}$ can be optimized independently in parallel. We use two layer multi-layer perceptron (MLP) to implement G_f and each $G^{(l)}$. Actually, more complex deep neural networks could replace the generative model outlined here. We leave exploring feasible deep neural networks as a possible future direction. To fool the discriminator, for each $(v_i, v_j) \sim p_{data}$, where p_{data} represents the multi-view connectivity, we propose to sample a pair of negative nodes from G that has connectivity that is similar to that of (v_i, v_j), e.g., $(v_i, v_c) \sim p_g$ with $\tilde{\mathcal{K}}_{ic} = \mathcal{K}_{ij}$, where p_g denotes the distribution modeled by G. In particular, we choose the v_c that has the highest probability as:

$$\arg\max_{v_c \in \mathcal{V}} \prod_{l=1}^{k} G^{(l)}(e_{ic}^{(l)} = e_{ij}^{(l)} | v_i, v_j, G_f(v_i, v_j; \mathbf{X}, \theta_f) \tag{2}$$

The motivation behind the negative sampling strategy in Eq.(2) is that the negative node pair (v_i, v_c) is more likely to fool the connectivity discriminator if connectivity $\tilde{\mathcal{K}}_{ic}$ is the same to the connectivity $\mathcal{K}_{ij}$ of the positive pair (v_i, v_j). *The objective of G is then to update network embedding $\mathbf{X}$ and θ_G so that the generator has higher chance of producing negative samples that can fool the discriminator D.*

Node Pair Discriminator

The goal of D is to discriminate between the positive node pairs from the multi-view network data and the negative node pairs produced by the generator G so as to enforce G to more accurately fit the distribution of multi-view network connectivity. For this purpose, for an arbitrary node pair sampled from real-data, i.e., $(v_i, v_j) \sim p_{data}$, D should output 1, meaning that the sampled node pair is real. Given such a negative or fake edge (pair of nodes), the discriminator should output 0, whereas G should aim to assign high enough probability to negative pair of nodes that can fool D. As D learns to distinguish the negative pair of nodes from the positive ones, the G captures the connectivity distribution of the multi-view graph. We will show this in Section 3.4.

We define the D as the sigmoid function of the inner product of the input node pair (v_i, v_j):

$$D(v_i, v_j) = \frac{\exp(\mathbf{d_i}^T \cdot \mathbf{d_j})}{1 + \exp(\mathbf{d_i}^T \cdot \mathbf{d_j})} \tag{3}$$

where $\mathbf{d}_i$ and $\mathbf{d}_j$ denote the d dimensional representation of node pair (v_i, v_j). It is worth noting that $D(v_i, v_j)$ could be any differentiable function with domain $[0, 1]$. We choose the sigmoid function for its stable property and leave the selections of $D(\cdot)$ as a possible future direction.

Efficient Negative Node Sampling

In practice, for one pair of nodes (v_i, v_j), in order to sample a pair of nodes (v_i, v_c) from G, we need to calculate Eq.(2) for all $v_c \in \mathcal{V}$ and select the v_c with the highest probability, which can be very time-consuming. To make the negative sampling more efficient, instead of calculating the probability for all nodes, we calculate the probability for neighbors of v_i, i.e., $\mathcal{N}(v_i)$, where $\mathcal{N}(v_i)$ is the set of nodes that have connectivity to v_i for at least one view in real network. In other words, we only sample from $\mathcal{N}(v_i)$. The size of $\mathcal{N}(v_i)$ is significantly smaller than n because the network is usually very sparse, which makes the sampling very efficient.

Objective Function of MEGAN

With generator modeling multi-view connectivity to generate fake samples that could fool the discriminator, discriminator differentiates between true pairs of nodes from fake pairs of nodes. We specify the objective function for MEGAN:

$$\min_{\theta_G}\max_{\theta_D} V(G,D) = \sum_{i=1}^{n}(\mathbb{E}_{(v_i,v_j)\sim p_{\text{data}}}[\log D(v_i,v_j;\theta_D)] + \mathbb{E}_{(v_i,v_c)\sim p_g}[\log(1-D(v_i,v_c;\theta_D))]) \tag{4}$$

where $(v_i, v_j) \sim p_{\text{data}}$ denotes the positive nodes pair and $(v_i, v_c) \sim p_g$ denotes the fake pair of nodes obtained using the efficient negative sampling strategy outlined above. *Through such minmax game, we can learn network embedding* $\mathbf{X}$ *and the generator* G *that can approximate the multi-view network to fool the discriminator*. In other words, the learned network embedding $\mathbf{X}$ is able to capture the connectivity among nodes within each views and the correlations between views.

3.3 Training Algorithm of MEGAN

Following the standard approach to training GAN [Goodfellow *et al.*, 2014; Wang *et al.*, 2018], we alternate between the updates D and G with mini-batch gradient descent.

Updating D**:** Given that $\theta_D = \{d_i\}_{i=1}^n$ is differentiable w.r.t to the loss function in Eq.(4), the gradient of θ_D is given as:

$$\nabla_{\theta_D} V(G,D) = \sum_{i=1}^{n}(\mathbb{E}_{(v_i,v_j)\sim p_{\text{data}}}[\nabla_{\theta_D}\log D(v_i,v_j)] + \mathbb{E}_{(v_i,v_c)\sim p_g}[\nabla\theta_D\log(1-D(v_i,v_c))]) \tag{5}$$

Updating G**:** Since we sampled discrete data, i.e., the negative node IDs, from MEGAN, the discrete outputs make it difficult to pass the gradient update from the discriminative model to the generative model. Following the previous work [Yu *et al.*, 2017; Wang *et al.*, 2018], we utilize the policy gradient to update the generator parameters $\theta_G = \{\mathbf{X}, \theta_f, \theta^{(1)}, \ldots, \theta^{(k)}\}$:

$$\begin{aligned}
&\nabla_{\theta_G} V(G,D)\\
&= \nabla_{\theta_G}\sum_{i=1}^{n}\mathbb{E}_{(v_i,v_c)\sim p_g}[\log(1-D(v_i,v_c;\theta_D)]\\
&= \nabla_{\theta_G}\sum_{i=1}^{n}\sum_{c=1}^{n} G(\mathcal{K}_{ic}|v_i,v_c)[\log(1-D(v_i,v_c;\theta_D)]\\
&= \sum_{i=1}^{n}\mathbb{E}_{(v_i,v_c)\sim G}[\nabla_{\theta_G}\log G(\mathcal{K}_{ic}|v_i,v_c)\log(1-D(v_i,v_c;\theta_D))]
\end{aligned} \tag{6}$$

Training Algorithm With the update rules for θ_D and θ_G in place, the overall training algorithm is summarized in Algorithm 1. In Line 1, we initialize and pre-train the D and G. From Line 3 to 6, we update parameters of G, i.e., θ_G. From Line 7 to 10, we update the parameters of D, i.e., θ_D The D and G play against each other until the MEGAN converges.

3.4 Theoretical Analysis

It has been shown in [Goodfellow *et al.*, 2014] that p_g could converge to p_{data} in continuous space. In Proposition 1, we show that p_g also converge p_{data} in discrete space. In other words, upon convergence of Algorithm 1, MEGAN can learn embeddings that makes $p_g \approx p_{data}$, which captures the multi-view network topology.

Algorithm 1 MVGAN framework

Require: embedding dimension d, size of discriminating samples t, generating samples s
Ensure: θ_D, θ_G
1: Initialize and pre-train D and G
2: **while** MVGAN not converged **do**
3: **for** G-steps **do**
4: Sample s negative pairs of nodes (v_i, v_c) for the given positive pair of nodes (v_i, v_j)
5: update θ_G according to Eq.(1) and Eq.(6)
6: **end for**
7: **for** D-steps **do**
8: Sample t positive nodes pairs (v_i, v_j) and t negative node pairs (v_i, v_c) from p_g for each node v_i
9: update θ_D according to Eq.(3) and Eq.(5)
10: **end for**
11: **end while**

Proposition 1. *If* G *and* D *have enough capacity, the discriminator and the generator are allowed to reach its optimum, and* p_g *is converge to* p_{data} *based on the update rule in Algorithm 1.*

Proof. The proof is similar to [Goodfellow *et al.*, 2014], and we omit the details here. □

4 Experiments

We report results of our experiments with two benchmark multi-view network data sets designed to compare the quality of multi-view embeddings learned by MEGAN and other state-of-the-art network embedding methods. We use the embedding learned by each method on three tasks, namely, node classification, link prediction, and network visualization. In these tasks, we use the performance as a quantitative proxy measure for the quality of the embedding. We also examine sensitivity of MEGAN w.r.t the choice of hyperparameters.

4.1 Data Sets

We use the following multi-view network data sets[Bui *et al.*, 2016] in our experiments: (i). **Last.fm**: Last.fm data were collected from the online music network Last.fm[1]. The nodes in the network represent users of Last.fm and the edges denote different types of relationships between users, e.g., shared interest in an artist, event, etc. (ii). **Flickr:** Flickr data were collected from the Flickr photo sharing service. The views correspond to different aspects of shared interest between users in photos (e.g., tags, comments, etc.). The statistics of the data sets are summarized in Table **??**. The number of edges denotes the total number of edges (summed over all of the views). Perhaps not unsurprisingly, the degree distribution of each view of the data approximately follows power-law degree distribution.

[1]https://www.last.fm

Datasets	#nodes	#edges	#view	#label
Last.fm	10,197	1,325,367	12	11
Flickr	6,163	378,547	5	10

Table 1: Summary of Flickr and Last.fm data

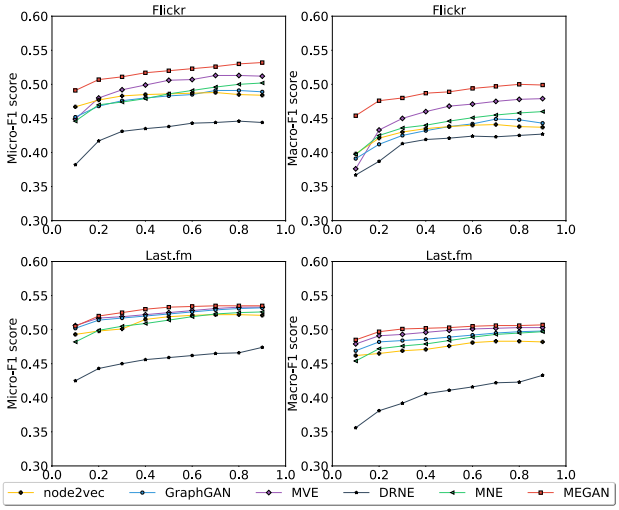


Figure 3: Performance comparison of node classification tasks on Flickr and Last.fm as a function of the fraction of nodes used for training the node classifier

4.2 Experiments

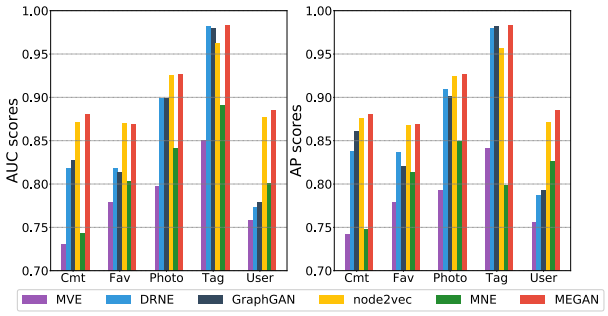
We compare MEGAN with the state-of-the-art single view as well as multi-view network embedding methods. To apply single-view method, we generate a single-view network from the multi-view network by placing an edge between a pair of nodes if they are linked by an edge in at least one of the views. The single view methods included in the comparison are:

- node2vec [Grover and Leskovec, 2016], a single view network embedding method, which learns network embedding that maximizes the likelihood of preserving network neighborhoods of nodes.
- GraphGAN [Wang *et al.*, 2018], which is a variant of GAN for learning single-view network embedding.
- DRNE [Tu *et al.*, 2018], which constructs utilizes an LSTM to recursively aggregate the representations of node neighborhoods.

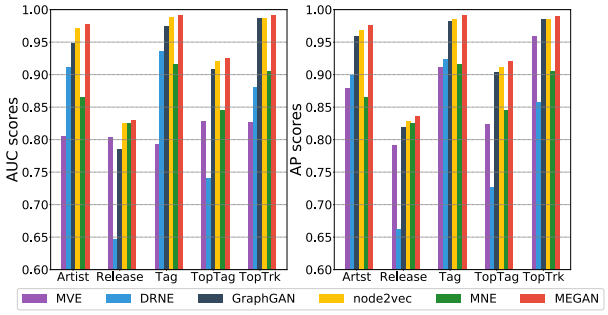
The multi-view methods included in the comparison are:

- MNE [Zhang *et al.*, 2018a], which jointly learns view-specific embeddings and an embedding that is common to all views with the latter providing a conduit for sharing information across views.
- MVE [Qu *et al.*, 2017], which constructs a multi-view network embedding as a weighted combination of the constituent single view embeddings.

In each case, the hyperparameters were set according to the suggestions of the authors of the respective methods. The embedding dimension was set to 128 in all of our experiments.



(a) Flickr



(b) Last.fm

Figure 4: Performance comparison of link prediction tasks on Flickr and Last.fm

4.3 Results

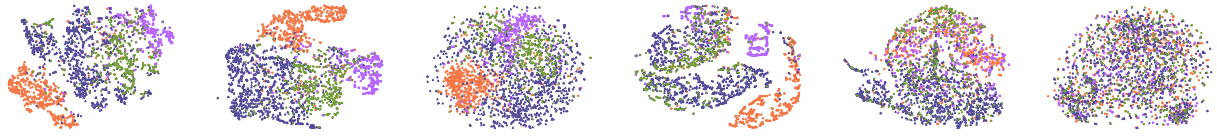
Node Classification

We report results of experiments using the node representations produced by each of the network embedding methods included in our comparison on the transductive node classification task. In each case, the network embedding is learned in an unsupervised fashion from the available multi-view network data without making use of the node labels. We randomly select x fraction of the nodes as training data (with the associated node labels added) and the remaining $(1 - x)$ fraction of the nodes for testing. We run the experiments for different choices of $x \in \{0.1, 0.2, \ldots, 0.9\}$. In each case, we train a standard one-versus-rest L2-regularized logistic regression classifier on the training data and evaluate its performance on the test data. We report the performance of the node classification using the Micro-F1 and Macro-F1 scores averaged over the 10 runs for each choice of x in Fig. 3.

Our experiments results show that: **(i)** The single view methods, GraphGAN and Node2vec achieve comparable performance; **(ii)** Multi-view methods, MVE and MVGAN outperform the single-view methods. **(iii)** MEGAN outperforms all of the other methods on both data sets. These results further show that multi-view methods that construct embeddings that incorporate complementary information from all of the views outperform those that do not. GAN framework offers the additional advantage of robustness and improved generalization that comes from the use of adversarial samples.

Link Prediction

We report results of experiments using the node representations produced by each of the network embedding methods



(a) MEGAN (b) Node2Vec (c) MVE (d) GraphGAN (e) DRNE (f) MNE

Figure 5: Visualization results of the Flickr. The users are projected into 2D space. Color of the node represents the categories of the user.

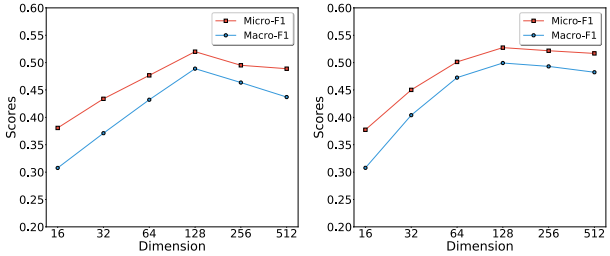
included in our comparison on the link prediction task. Given a multi-view network, we randomly select a view, and randomly remove 50% of the edges present in that view. We then train a classifier on the remaining data to predict the links that were removed. Following [Grover and Leskovec, 2016], we cast the link prediction task as the binary classification problem with the network edges that were not removed used as positive examples, and an equal number of randomly generated edges that do not appear in the network as negative examples. We represent links using embeddings of the corresponding pair of nodes. We train and test a random forest classifier for link prediction. In the case of the Flickr dataset, we repeat the above procedure on each of the five views; and in the case of the Last.fm data set, present the results on five of the most populous views. We report the area under curve (AUC) and average precision (AP) for link prediction for the Flickr and Last.fm data sets in Fig.4(a) and Fig.4(b), respectively. Based on these results, we make the following observations: **(i)** There is fairly large variability in the performance of link prediction across the different views. Such phenomenon reveals the differences in the reliability of the link structures in each view; **(ii)** In some views that are fairly rich in links, e.g., Tag view of the Flickr data, single view methods, such as GraphGAN and DRNE, outperform multi-view methods, MVE and MNE, and approaching MEGAN. This suggests that although single view methods may be competitive with multi-view methods when almost all of the information needed for reliable link prediction is available in a single view, multi-view methods outperform single-view methods when views other than the target view provide complementary information for link prediction in the target view. and **(iii)** The MEGAN outperforms its single-view counterpart GraphGAN, suggesting that the MEGAN is able to effectively integrate complementary information from multiple views into a compact embedding.

Network Visualization

To better understand the intrinsic structure of the learned network embedding [Tang *et al.*, 2016] and reveal the quality of it, we visualize the network by projecting the embeddings onto a 2-dimensional space. We show the resulting network visualizations on the Flickr network using each of the comparison embedding methods with the t-SNE package[Maaten and Hinton, 2008] in Fig. 5. In the figure, each color denotes one category of users' interests and we show the results for four of ten categories. Visually, in Fig.5(a), we find that the results obtained by using MEGAN appear to yield tighter clusters for the 4 categories with more pronounced separation between clusters as compared to the other methods .

Impact of Embedding Dimension

We report results of the choice of embedding dimension on the performance of MEGAN. We chose 50% of the nodes randomly for training and the remaining for testing. We used different choices of the dimension $d \in \{16, 32, 64, 128, 256, 512\}$. We report the performance achieved using the resulting embeddings on the node classification task using Micro-F1 and Macro-F1 for Flickr and Last.fm in Fig.6(a) and Fig.6(b), respectively. On these two data sets, we find that MEGAN achieves its optimal performance when d is set to 128. This suggests that in specific applications, it may be advisable to select an optimal d using cross-validation.



(a) Flickr (b) Last

Figure 6: Parameter sensitivity for node classification on (a) Flickr and (b) Last.fm datasets with varying dimensions d.

5 Conclusions

In this paper, we have considered the multi-view network representation learning problem, which targeting at construct the low-dimensional, information preserving and non-linear representations of *multi-view* networks. Specifically, we have introduced MEGAN, a novel generative adversarial network (GAN) framework for multi-view network embedding aimed at preserving the connectivity within each individual network views, while accounting for the associations across different views. The results of our experiments with several multi-view data sets show that the embeddings obtained using MEGAN outperform the state-of-the-art methods on node classification, link prediction and network visualization tasks.

Acknowledgements

This work was funded in part by grants from the NIH NCATS through the grant UL1 TR002014 and by the NSF through the grants 1518732, 1640834, and 1636795, the Edward Frymoyer Endowed Professorship in Information Sciences and Technology at Pennsylvania State University and the Sudha Murty Distinguished Visiting Chair in Neurocomputing and Data Science funded by the Pratiksha Trust at the Indian Institute of Science (both held by Vasant Honavar). The content is solely the responsibility of the authors and does not necessarily represent the official views of the sponsors.

References

[Belkin and Niyogi, 2001] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[Bojchevski *et al.*, 2018] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. Netgan: Generating graphs via random walks. *arXiv preprint arXiv:1803.00816*, 2018.

[Bui *et al.*, 2016] Ngot Bui, Thanh Le, and Vasant Honavar. Labeling actors in multi-view social networks by integrating information from within and across multiple views. In *BigData*, pages 616–625. IEEE, 2016.

[Cavallari *et al.*, 2017] Sandro Cavallari, Vincent W Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In *CIKM*, pages 377–386, 2017.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of SIGKDD*, pages 855–864. ACM, 2016.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[He *et al.*, 2015] Kun He, Yiwei Sun, David Bindel, John Hopcroft, and Yixuan Li. Detecting overlapping communities from local spectral subspaces. In *ICDM*, pages 769–774. IEEE, 2015.

[Huang *et al.*, 2018] Jiaming Huang, Zhao Li, Vincent W. Zheng, Wen Wen, Yifan Yang, and Yuanmi Chen. Unsupervised multi-view nonlinear graph embedding. In *UAI*, pages 319–328, 2018.

[Kivelä *et al.*, 2014] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[Ma *et al.*, 2017] Guixiang Ma, Lifang He, Chun-Ta Lu, Weixiang Shao, Philip S Yu, Alex D Leow, and Ann B Ragin. Multi-view clustering with graph embedding for connectome analysis. In *CIKM*. ACM, 2017.

[Ma *et al.*, 2019] Yao Ma, Suhang Wang, Chara C Aggarwal, Dawei Yin, and Jiliang Tang. Multi-dimensional graph convolutional networks. In *SDM*, pages 657–665. SIAM, 2019.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Ou *et al.*, 2016] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Sigkdd. pages 1105–1114, 2016.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of SIGKDD*, pages 701–710. ACM, 2014.

[Qu *et al.*, 2017] Meng Qu, Jian Tang, Jingbo Shang, Xiang Ren, Ming Zhang, and Jiawei Han. An attention-based collaboration framework for multi-view network representation learning. In *CIKM*, pages 1767–1776. ACM, 2017.

[Shi *et al.*, 2016] Yu Shi, Myunghwan Kim, Shaunak Chatterjee, Mitul Tiwari, Souvik Ghosh, and Rómer Rosales. Dynamics of large multi-view social networks: Synergy, cannibalization and cross-view interplay. In *KDD*, pages 1855–1864. ACM, 2016.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of WWW*, pages 1067–1077, 2015.

[Tang *et al.*, 2016] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of WWW*, pages 287–297, 2016.

[Tu *et al.*, 2018] Ke Tu, Peng Cui, Xiao Wang, Philip S Yu, and Wenwu Zhu. Deep recursive network embedding with regular equivalence. In *Proceedings of SIGKDD*, pages 2357–2366. ACM, 2018.

[Wang *et al.*, 2015] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, pages 1083–1092, 2015.

[Wang *et al.*, 2016a] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *SIGKDD*, pages 1225–1234, 2016.

[Wang *et al.*, 2016b] Suhang Wang, Jiliang Tang, Charu Aggarwal, and Huan Liu. Linked document embedding for classification. In *CIKM*, pages 115–124. ACM, 2016.

[Wang *et al.*, 2018] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: graph representation learning with generative adversarial nets. In *AAAI*, 2018.

[Yu *et al.*, 2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.

[Yu *et al.*, 2018] Yanwei Yu, Huaxiu Yao, Hongjian Wang, Xianfeng Tang, and Zhenhui Li. Representation learning for large-scale dynamic networks. In *International Conference on Database Systems for Advanced Applications*, pages 526–541. Springer, 2018.

[Zhang *et al.*, 2018a] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. Scalable multiplex network embedding. In *IJCAI*, pages 3082–3088, 2018.

[Zhang *et al.*, 2018b] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In *SIGKDD*, pages 2778–2786, 2018.