Exploiting Node Content for Multiview Graph Convolutional Network and Adversarial Regularization

Qiuhao Lu[1], Nisansa de Silva[1], Dejing Dou[1], Thien Huu Nguyen[1], Prithviraj Sen[2], Berthold Reinwald[2], Yunyao Li[2]
[1]Department of Computer and Information Science, University of Oregon
[2]IBM Research - Almaden
{luqh, nisansa, dou, thien}@cs.uoregon.edu
{senp, reinwald, yunyaoli}@us.ibm.com

Abstract

Network representation learning (NRL) is crucial in the area of graph learning. Recently, graph autoencoders and its variants have gained much attention and popularity among various types of node embedding approaches. Most existing graph autoencoder-based methods aim to minimize the reconstruction errors of the input network while not explicitly considering the semantic relatedness between nodes. In this paper, we propose a novel network embedding method which models the consistency across different views of networks. More specifically, we create a second view from the input network which captures the relation between nodes based on node content and enforce the latent representations from the two views to be consistent by incorporating a multiview adversarial regularization module. The experimental studies on benchmark datasets prove the effectiveness of this method, and demonstrate that our method compares favorably with the state-of-the-art algorithms on challenging tasks such as link prediction and node clustering. We also evaluate our method on a real-world application, i.e., 30-day unplanned ICU readmission prediction, and achieve promising results compared with several baseline methods.

1 Introduction

Over the last few years, network representation learning, or node embedding, has gained increasing interest in the community of machine learning, due to the popularity of the special data form. In reality, datasets from different fields are often in the form of networks, such as social networks, drug-target-interaction networks, mobile phone networks, citation networks, etc. It is therefore very important to find a way to well represent the networks, which is challenging because there is no direct way to encode the high-dimensional data into low-dimensional feature vectors efficiently (Hamilton et al., 2017b). Moreover, network embedding techniques benefit a variety of downstream applications like link prediction, node classification and node clustering.

In recent years, researchers have developed different kinds of network embedding approaches, many of which have shown great performance in analytical evaluation and have been quite effective in downstream applications. These studies range from traditional machine learning techniques like matrix factorization to recent deep-learning-based methods like graph autoencoders.

Traditional models, or shallow models, usually optimize the embeddings of nodes directly. For these shallow models, the mapping from networks to vectors is simply a embedding lookup, i.e., each node corresponds to a unique embedding vector (Hamilton et al., 2017b). Factorization-based approaches like GraRep (Cao et al., 2015), HOPE (Ou et al., 2016) and random walk-based approaches like DeepWalk (Perozzi et al., 2014), node2vec (Grover and Leskovec, 2016) all fall into this category. Shallow models generally suffer from computational inefficiency and lack of ability to well represent complex networks.

More recently, deep models, or autoencoder-based approaches, have been gaining more and more attention, and have shown superior performance in many applications. Compared with shallow models which use a simple lookup table as the encoder function, deep models usually use deep neural networks as the encoder. For example, SDNE (Wang et al., 2016) and DNGR (Cao et al., 2016) use deep neural

Proceedings of the 28th International Conference on Computational Linguistics, pages 545–555
Barcelona, Spain (Online), December 8-13, 2020

networks as the encoder and decoder functions to generate low-dimensional representations. GAE and VGAE (Kipf and Welling, 2016b) aggregate neighborhood messages based on convolutional encoders, e.g., graph convolutional networks (GCN) (Kipf and Welling, 2016a) and its variants, to generate node embeddings. The encoders share parameters across nodes and it leads to better efficiency. Note that GCN variants like GraphSAGE (Hamilton et al., 2017a) and GAT (Veličković et al., 2017) are not discussed as they mostly focus on message passing which is not the main focus of this work.

Another successful variant of graph autoencoders incorporates the generative adversarial networks (GAN) for representation learning. For example, ARGA and ARVGA (Pan et al., 2018) enforce the latent node embeddings to match a prior normal distribution based on an adversarial training mechanism. The adversarial training procedure usually provides regularization and results in more robust and meaningful representations (Makhzani et al., 2015). DBGAN (Zheng et al., 2020) estimates the prior distribution of latent representations by prototype learning and aims to balance both sample-level and distribution-level consistency via a novel bidirectional adversarial learning framework.

A common theme among most of the aforementioned approaches is that they do not explicitly consider the semantic relatedness between nodes. For shallow models like DeepWalk (Perozzi et al., 2014), they mostly only focus on preserving the topological structure of the network while neglecting the rich information in node content. For deep models, they implicitly incorporate node content by aggregating neighborhood node features using powerful encoders like graph convolutional networks.

In this paper, we propose a novel network embedding method based on multiview graph convolutional network and adversarial regularization. The method aims to preserve the distribution consistency across two views of the network, as well as shape the output representations to match an arbitrary prior distribution, by incorporating a multiview adversarial regularization module. More specifically, we regard the topological structure as the first and main view of the network, and create a second view that captures the relatedness between nodes based on node content. Different from DBGAN (Zheng et al., 2020) which tries to reconstruct the node features directly, the proposed method relaxes this requirement and focuses on preserving the semantic relatedness between them. A multiview reconstruction loss function is leveraged to optimize the model jointly. We evaluate the proposed method on three diverse applications. The experimental results on benchmark datasets demonstrate that the method outperforms the state-of-the-art algorithms in link prediction and node clustering. We also evaluate our method on a real-world downstream application, i.e., ICU readmission prediction, and the method compares favorably with several baseline methods. Our contributions can be summarized as follows:

- We propose a novel network embedding method, i.e., Multiview Adversarially Regularized Graph Autoencoder (MRGAE). Unlike previous studies that either neglect node content or aim to reconstruct the entire node feature matrix, we focus on the semantic relatedness between nodes and aim to preserve the consistency of node presentations across two specific views of the network. We incorporate a multiview adversarial regularization module to achieve the objective and enforce the output representations to match a prior distribution.

- We conduct extensive and diverse experiments for evaluation. The experimental studies demonstrate the superb performance of our method, by updating the state-of-the-art results in link prediction and node clustering on benchmark datasets. Our method also compares favorably with baselines in the task of ICU readmission prediction.

2 Methodology

2.1 Preliminaries

2.1.1 Graph Convolutional Networks

Most recent graph neural network models usually use a common architecture, i.e., graph convolutional networks (GCN) (Kipf and Welling, 2016a), to encode the input networks. Essentially, graph convolutional networks transform the original graph or network into a lower-dimensional representation matrix $\mathbf{Z}$, given the adjacency matrix $\mathbf{A}$ and the feature matrix $\mathbf{X}$ as the input. Each of the transformations can

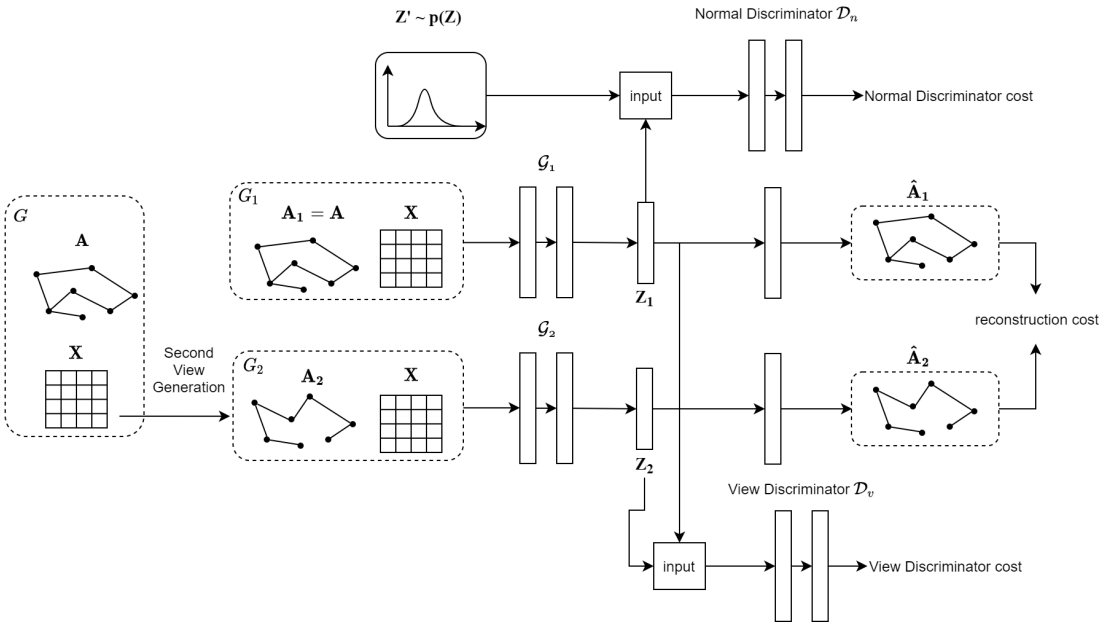

Figure 1: Architecture of MRGAE.

be written as a non-linear convolution function:

$$\mathbf{H}^{(l+1)} = f(\mathbf{H}^{(l)}, \mathbf{A}) \tag{1}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$ which is the input feature matrix, and $\mathbf{H}^{(l)}$ refers to the output representation matrix (i.e., embeddings) $\mathbf{Z}^{(l)}$ for the l-th layer convolutional neural network. Essentially, different types of the convolution function f usually correspond to variants of the GCN model. The standard convolution function can be written as:

$$\mathbf{Z}^{(l+1)} = \mathbf{H}^{(l+1)} = f(\mathbf{H}^{(l)}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \tag{2}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, and $\mathbf{I}$ is the identity matrix of $\mathbf{A}$. $\hat{\mathbf{D}}$ is the diagonal degree matrix of $\hat{\mathbf{A}}$, and $\mathbf{W}^{(l)}$ is the weight matrix for the l-th layer neural network, which is also the parameter to optimize. We use the ReLU function as the activation function σ in this paper, and adopt a two-layer GCN as the encoder for all the experiments.

2.1.2 Adversarial Regularization

Adversarial regularization has proven effective in various network representation learning approaches (Makhzani et al., 2015; Pan et al., 2018; Dai et al., 2018). Generally, in the *encoder-decoder* framework, one can view the encoder as a generator, and incorporate a discriminator (e.g., a multi-layer perceptron) to distinguish whether a latent representation is from the encoder or from an arbitrary prior distribution. By incorporating this module, one can shape the learned representations to match an arbitrary prior distribution, e.g., Gaussian distribution. This is similar in spirit to VGAE, which uses KL divergence instead of adversarial training to achieve the same purpose (Makhzani et al., 2015). In this work, we extend the adversarial regularization module to a multiview scenario, where we aim to enforce the learned representations from the two views to be distribution consistent and to match a prior distribution.

2.2 MRGAE

2.2.1 Overview

The overall framework of the proposed method contains three main parts, as depicted in Figure 1. First, we consider the topological structure as the first and main view of the input network, and create a second view of it. Next, we use two graph convolutional networks (GCN) as the encoders to separately encode

the two views of the input network. Then, we incorporate two discriminators, one to distinguish between the representations from the main view and the prior distribution, the other to distinguish between the representations from the two views. In this paper, we use the Gaussian distribution as the prior distribution since the Gaussian assumption has been widely adopted in various previous studies (Makhzani et al., 2015; Kipf and Welling, 2016b; Pan et al., 2018). Finally, we design a specific multiview reconstruction loss function, combine it with the two discriminators, and optimize the model jointly.

2.2.2 Notations

Given the undirected input network $G = (V, E)$, we regard it as the first view G_1 and create a second view G_2 from it. Specifically, we denote the two views of the network G as $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, respectively. Note that we have $E_1 = E$. Each view $G_i (i = 1, 2)$ has the same node set V with N nodes ($N = |V|$) and a different set of edges E_i. Each view has its own adjacency matrix $\mathbf{A_i}$ and degree matrix $\mathbf{D_i}$. We further introduce a $N \times D$ feature matrix $\mathbf{X}$ for V, where each row corresponds to the input features of D dimensions for each node. For featureless networks, we use the identity matrix as a replacement for $\mathbf{X}$. The goal is to learn a unified representation matrix $\mathbf{Z}$ for the nodes.

2.2.3 Second View Construction

We aim to construct a second view $G_2 = (V, E_2)$ of the network that captures the semantic relatedness between nodes. To define E_2, we adopt a straightforward strategy to calculate cosine similarities between node content. Essentially, if the cosine similarity between two nodes is greater than a threshold α_{prox}, then we create a link between them in the second view.

2.2.4 Encoder-Decoder Framework

In this paper, we follow the generalized *encoder-decoder* framework (Hamilton et al., 2017b) for learning network representations. More specifically, we adopt two-layer GCNs as the encoders, and each of them encodes one single view of the input multiview network. Essentially, the encoder model transforms the nodes in the network into low-dimensional feature representations (i.e., embeddings), and this encoding process can be written as:

$$\mathbf{Z_i} = \mathrm{ENC}(\mathbf{X}, \mathbf{A_i}) = \mathrm{GCN}(\mathbf{X}, \mathbf{A_i}) \tag{3}$$

where $\mathbf{Z_i}$ refers to the representation matrix learned from the i-th view G_i. Along with Equation 2, the encoding process can then be further explained as:

$$\mathbf{Z_i^{(0)}} = \mathbf{X} \tag{4}$$

$$\mathbf{Z_i^{(1)}} = \mathrm{ReLU}(\hat{\mathbf{D}}_\mathbf{i}^{-\frac{1}{2}} \hat{\mathbf{A}}_\mathbf{i} \hat{\mathbf{D}}_\mathbf{i}^{\frac{1}{2}} \mathbf{X} \mathbf{W_i}^{(0)}) \tag{5}$$

$$\mathbf{Z_i^{(2)}} = \hat{\mathbf{D}}_\mathbf{i}^{-\frac{1}{2}} \hat{\mathbf{A}}_\mathbf{i} \hat{\mathbf{D}}_\mathbf{i}^{\frac{1}{2}} \mathbf{Z_i^{(1)}} \mathbf{W_i}^{(1)} \tag{6}$$

where $\hat{\mathbf{A}}_\mathbf{i}$ and $\hat{\mathbf{D}}_\mathbf{i}$ refer to the adjacency matrix and degree matrix of the i-th view G_i, respectively. Similarly, $\mathbf{W_i}^{(l)}$ represents the parameter matrix for the l-th layer graph convolutional network with G_i. Thus, in general this encoding process with Equation 3 can be written as:

$$\mathbf{Z_i} = \mathrm{ENC}(\mathbf{X}, \mathbf{A_i}) = q(\mathbf{Z_i}|\mathbf{X}, \hat{\mathbf{A}}_\mathbf{i}) = \mathbf{Z_i^{(2)}} \tag{7}$$

With regard to the *decoder* model, essentially it decodes the learned low-dimensional representations, and transforms them into some information that can be evaluated in some way, for example, the existence of edges between nodes or label predictions on specific downstream tasks. The evaluations are a good way to measure the quality of the learned representations of nodes. In this paper, we use a simple yet effective pair-wise inner-product decoder to reconstruct the edges of the original network, which is shown as follows:

$$\mathrm{DEC}(\mathbf{z_p}, \mathbf{z_q}) = \mathbf{z_p}^\top \mathbf{z_q} \tag{8}$$

The inner-product decoder model aims to reconstruct the edge set between nodes in the input network, where the reconstructed edge set should be as similar as the original one. In our case, the reconstruction

loss is calculated based on each of the views, i.e., the decoder aims to reconstruct each view from the learned representations from that view, respectively. The decoding process is shown as follows:

$$p(\hat{\mathbf{A}}_\mathbf{i}|\mathbf{Z_i}) = \prod_{p=1}^{N}\prod_{q=1}^{N} p((\hat{A}_i)_{pq}|\mathbf{z_{ip}}, \mathbf{z_{iq}}) \tag{9}$$

$$\begin{aligned} p((\hat{A}_i)_{pq} = 1|\mathbf{z_{ip}}, \mathbf{z_{iq}}) &= \sigma_s(\mathrm{DEC}(\mathbf{z_{ip}}, \mathbf{z_{iq}})) \\ &= \sigma_s(\mathbf{z_{ip}}^\top \mathbf{z_{iq}}) \end{aligned} \tag{10}$$

where $(\hat{A}_i)_{pq}$ refers to the edges between nodes, and σ_s here is the logistic sigmoid function.

2.2.5 Multiview Adversarial Regularization

The intuition is that we want the latent embeddings learned from different views are consistent, i.e., the same nodes from different views are close in the embedding space, and the learned latent embeddings from different views fit a similar distribution. Thus, we propose the loss function should be in the following form:

$$\mathcal{L} = \sum_{i=1}^{2}(\alpha_i \mathbb{E}_{q(\mathbf{Z_i}|\mathbf{X},\hat{\mathbf{A}}_\mathbf{i})}[-\log p(\hat{\mathbf{A}}_\mathbf{i}|\mathbf{Z_i})]) + \mathcal{S} \tag{11}$$

where α_i are the balancing coefficients. Intuitively, the first term corresponds to the addition of the individual reconstruction loss from each view. The second term $\mathcal{S}$ is the term that models the consistency across different views, and the specific methods differ in how this term is chosen and parameterized.

We then introduce a multiview reconstruction loss (MRL) function:

$$\mathcal{L}_{mrl} = \sum_{i=1}^{2}(\alpha_i \mathbb{E}_{q(\mathbf{Z_i}|\mathbf{X},\hat{\mathbf{A}}_\mathbf{i})}[-\log p(\hat{\mathbf{A}}_\mathbf{i}|\mathbf{Z_i})]) + \beta \mathbb{E}_{\mathbf{Z_1}\sim q(\mathbf{X},\hat{\mathbf{A}}_\mathbf{1}),\mathbf{Z_2}\sim q(\mathbf{X},\hat{\mathbf{A}}_\mathbf{2})}[-\log p(\hat{\mathbf{A}}_\mathbf{1}|\mathbf{Z_1},\mathbf{Z_2})] \tag{12}$$

where the first term refers to the addition of the individual reconstruction loss from each view, and the second term is the loss of reconstructing the graph structure of the main view G_1 with the encoded representations from both views, i.e., $\mathbf{Z_1}$ and $\mathbf{Z_2}$. Here instead of only adding the individual reconstruction loss together, we use the encoded representations from both views to jointly reconstruct the main structure, thus achieving better consistency and robustness. More specifically, we have $p((\hat{A}_1)_{pq} = 1|\mathbf{z_{1p}}, \mathbf{z_{2q}}) = \sigma_s(\mathbf{z_{1p}}^\top \mathbf{z_{2q}})$.

Unlike previous work, we incorporate two discriminators, namely the normal discriminator $\mathcal{D}_n$ and the view discriminator $\mathcal{D}_v$, to distinguish between the representations from the main view and the Gaussian distribution, and to distinguish between the representations from the two views, as depicted in Figure 1. We share weights between them. The adversarial loss for the two discriminators is defined as:

$$\begin{aligned} \mathcal{L}_{adv} = &- (\mathbb{E}_{\mathbf{Z_n}\sim\mathcal{N}}[\log \mathcal{D}_n(\mathbf{Z_n})] + \mathbb{E}_{\mathbf{x}\sim p(x)}[1 - \mathcal{D}_n(\mathcal{G}_1(\mathbf{X}, \mathbf{A_1}))]) \\ &- (\mathbb{E}_{\mathbf{x}\sim p(x)}[\log \mathcal{D}_v(\mathcal{G}_1(\mathbf{X}, \mathbf{A_1}))] + \mathbb{E}_{\mathbf{x}\sim p(x)}[1 - \mathcal{D}_v(\mathcal{G}_2(\mathbf{X}, \mathbf{A_2}))]) \end{aligned} \tag{13}$$

where $\mathcal{G}_1$ and $\mathcal{G}_2$ refer to the two GCN encoders, respectively. And finally, we use a weighted sum of the above losses:

$$\mathcal{L}_1 = \mathcal{L}_{mrl} + \gamma \mathcal{L}_{adv} + \mathcal{L}_{reg} \tag{14}$$

where $\mathcal{L}_{reg}$ is a regularization term and we have $\mathcal{L}_{reg} = \mathbb{E}_{\mathbf{Z_1}\sim q(\mathbf{X},\hat{\mathbf{A}}_\mathbf{1})}[-\log \mathcal{D}_n(\mathbf{Z_1})]$. We then jointly train the model by minimizing $\mathcal{L}_1$, and finally take the encoded representations from the main view, i.e., $\mathbf{Z_1}$, as the output representations.

3 Experiments

In this section, we evaluate our proposed method based on three tasks. First, we conduct the experiment of link prediction on the benchmark dataset of three citation networks. We also report the experiment of node clustering on these networks. Finally, we apply the proposed method to a real-world medical application, i.e., 30-day unplanned ICU readmission prediction.[1]

[1] All datasets are freely available and the code is available at https://github.com/qiuhaolu/MRGAE.

Method	Cora		Citeseer		Pubmed	
	AUC	AP	AUC	AP	AUC	AP
SC	84.6 ± 0.01	88.5 ± 0.00	80.5 ± 0.01	85.0 ± 0.01	84.2 ± 0.02	87.8 ± 0.01
DW	83.1 ± 0.01	85.0 ± 0.00	80.5 ± 0.02	83.6 ± 0.01	84.2 ± 0.00	84.1 ± 0.00
GAE	91.0 ± 0.02	92.0 ± 0.03	89.5 ± 0.04	89.9 ± 0.05	96.4 ± 0.00	96.5 ± 0.00
VGAE	91.4 ± 0.01	92.6 ± 0.01	90.8 ± 0.02	92.0 ± 0.02	94.4 ± 0.02	94.7 ± 0.02
ARGA	92.4 ± 0.003	93.2 ± 0.003	91.9 ± 0.003	93.0 ± 0.003	96.8 ± 0.001	97.1 ± 0.001
ARVGA	92.4 ± 0.004	92.6 ± 0.004	92.4 ± 0.003	93.0 ± 0.003	96.5 ± 0.001	96.8 ± 0.001
MRGAE	94.0 ± 0.7	94.1 ± 0.6	94.3 ± 0.4	94.9 ± 0.8	97.2 ± 0.2	97.4 ± 0.3
DBGAN	94.5 ± 0.01	95.1 ± 0.05	94.5 ± 0.04	95.8 ± 0.01	96.8 ± 0.01	97.3 ± 0.02
MRGAE*	$\mathbf{95.0 \pm 0.3}$	$\mathbf{95.2 \pm 0.4}$	$\mathbf{95.7 \pm 0.5}$	$\mathbf{96.4 \pm 0.4}$	$\mathbf{97.8 \pm 0.1}$	$\mathbf{97.8 \pm 0.2}$

Table 1: Performance comparison on link prediction.

3.1 Link Prediction

Link prediction is a popular task in evaluating network embedding methods. Essentially, a small portion of the edges are removed for generating the validation and test sets, and the same number of pairs of unconnected nodes are randomly picked as negative samples. The goal of the task is to predict whether or not there exists an edge between two nodes.

3.1.1 Dataset and Second View Construction

We conduct the experiment on three popular citations networks, i.e., Cora, Citeseer and Pubmed (Sen et al., 2008). The nodes represent scientific publications from different areas, and the edges represent the citation links between them. The nodes are represented with feature vectors, which are described by 0/1-valued word vectors indicating the absence/presence of the corresponding word (Cora and Citeseer) or tf-idf weighted word vectors (Pubmed). Each node has a corresponding class label.

In this experiment, we take the original edge set of the input network, i.e., the citation links, as the first and main view. We construct the second view based on textual similarities. Essentially, if the cosine similarity between two publications is greater than the empirical threshold 0.7, then we create a link between them in the second view.

3.1.2 Baselines

We compare the proposed method with several baseline methods: Spectral Clustering (SC) (Tang and Liu, 2011), Deepwalk (DW), Graph Autoencoder (GAE), Variational Graph Autoencoder (VGAE), Adversarially Regularized Graph Autoencoder (ARGA), Adversarially Regularized Variational Graph Autoencoder (ARVGA) and DBGAN (Zheng et al., 2020).

3.1.3 Experiment Settings

For all the experiments, we split each of the datasets into the training set (85%), the validation set (5%), and the test set (10%). To reduce the influence of randomness, we average the results over five randomly selected splits as in (Zheng et al., 2020).

We use the same set of hyperparameters for the GCN encoder with the baselines (Kipf and Welling, 2016b; Pan et al., 2018; Zheng et al., 2020). More specifically, we use a 32-dim hidden layer and 16-dim latent representations for the GCN encoder in the link prediction task. We also use two multi-layer perceptrons (MLP) as the discriminators, each of which consists of two 128-dim hidden layers. We set the balancing factors $\alpha_1, \alpha_2, \gamma$ to 1.0, and set β to 0.8 in all experiments. The performance of our method is recorded as MRGAE in Table 1.

Note that DBGAN uses a larger embedding size in their experiments. For a fair comparison, we also set the representation size to 32-dim (Cora) and 64-dim (Citeseer and Pubmed), the results of which are

Method	Acc	NMI	F1	Prec	ARI	Method	Acc	NMI	F1	Prec	ARI
SC	0.367	0.127	0.318	0.193	0.031	SC	0.239	0.056	0.299	0.179	0.010
DW	0.484	0.327	0.392	0.361	0.243	DW	0.337	0.088	0.270	0.248	0.092
RTM	0.440	0.230	0.307	0.332	0.169	RTM	0.451	0.239	0.342	0.349	0.203
RMSC	0.407	0.255	0.331	0.227	0.090	RMSC	0.295	0.139	0.320	0.204	0.049
TADW	0.560	0.441	0.481	0.396	0.332	TADW	0.455	0.291	0.414	0.312	0.228
GAE	0.596	0.429	0.595	0.596	0.347	GAE	0.408	0.176	0.372	0.418	0.124
VGAE	0.609	0.436	0.609	0.609	0.346	VGAE	0.344	0.156	0.308	0.349	0.093
ARGA	0.640	0.449	0.619	0.646	0.352	ARGA	0.573	0.350	0.546	0.573	0.341
ARVGA	0.638	0.450	0.627	0.624	0.374	ARVGA	0.544	0.261	0.529	0.549	0.245
MRGAE	0.703	0.523	0.681	0.716	0.476	MRGAE	0.627	0.361	0.587	0.601	0.363
GALA	0.745	**0.576**	–	–	0.531	GALA	**0.693**	**0.441**	–	–	**0.446**
DBGAN	0.748	0.560	–	–	0.540	DBGAN	0.670	0.407	–	–	0.414
MRGAE*	**0.764**	0.559	**0.740**	**0.742**	**0.570**	MRGAE*	0.671	0.403	**0.620**	**0.620**	0.418

Table 2: Node clustering performance on Cora (left) and Citeseer (right).

recorded as MRGAE*.

3.1.4 Results

We use the same evaluation metrics with the previous work, i.e., *area under the Receiver Operating Characteristics curve* (AUC) and *average precision* (AP) scores.

As shown in Table 1, the proposed method (MRGAE) achieves the best performance on all three citation networks, outperforming the state-of-the-art method, i.e., DBGAN, indicating the effectiveness of exploiting node content by incorporating multiview adversarial regularization.

3.2 Node Clustering

In this experiment, we consider another unsupervised task of clustering nodes in the network. We first compute the embeddings of Cora and Citeseer and perform K-means clustering on them, where K is set to be the number of node classes in each network. Then we follow the same procedure of previous work (Shi et al., 2019; Xia et al., 2014; Pan et al., 2018) and match the predicted class labels with the ground-true labels using the Munkres assignment algorithm (Munkres, 1957). The results are evaluated based on accuracy (Acc), normalized mutual information (NMI), precision (Prec), F-score (F1) and average rand index (ARI).

3.2.1 Baselines

Except for the baselines we use in the link prediction task, we include four more baseline algorithms which are designed for clustering: RTM (Chang and Blei, 2009), RMSC (Xia et al., 2014), TADW (Yang et al., 2015), and GALA (Park et al., 2019).

3.2.2 Results

For a fair comparison, we first set the size of the output representations to 16-dim and record the results as MRGAE, and since GALA and DBGAN only report high dimensional performance, we then report the performance of our method with the same dimensions with DBGAN (i.e., 128-dim for Cora, 64-dim for Citeseer) and record the result as MRGAE*.

As shown in Table 2, our proposed method MRGAE outperforms the other methods on both datasets across all metrics. For the Cora dataset, the proposed method MRGAE* shows superior performance to GALA and DBGAN in almost all metrics except NMI. For the Citeseer dataset, MRGAE* and DBGAN perform similarly well while GALA gives the best results. It is mainly because GALA uses a 500-dim node representation which is much larger than DBGAN and MRGAE*.

Method	Link Prediction		Node Clustering				
	AUC	AP	Acc	NMI	F1	Prec	ARI
w/o $\mathcal{D}_v$	93.8	94.4	0.748	0.540	0.732	0.744	0.526
w/o MRL	94.0	94.1	0.706	0.527	0.686	0.712	0.496
w/o both	92.9	93.2	0.643	0.482	0.645	0.664	0.397
MRGAE	94.4	94.7	0.764	0.559	0.740	0.742	0.570

Table 3: Effectiveness evaluation of $\mathcal{D}_v$ and MRL.

3.3 Ablation Study

In this section, we validate the effectiveness of the multiview adversarial regularization module in our proposed method. We conduct the ablation experiments on both link prediction and node clustering tasks with the Cora dataset.

We first remove the view discriminator $\mathcal{D}_v$. By removing this part, the proposed method loses the ability to preserve the distribution consistency across the two specific views. We then remove the multiview reconstruction loss (MRL) and replace it with a simple GAE-based reconstruction loss. By removing this, the method losses rich information from the generated second view. Finally we remove both parts. The three ablated methods are recorded as "w/o $\mathcal{D}_v$", "w/o MRL" and "w/o both", respectively.

As shown in Table 3, removing either part would cause performance decrease on both link prediction and node clustering tasks, indicating the effectiveness and necessity of $\mathcal{D}_v$ and MRL. The ablated method "w/o both" shows the biggest performance decrease, which consistently validates the claim.

3.4 30-day Unplanned ICU Patient Readmission Prediction

In real-world networks, node content usually carries rich and important information for downstream applications, which highlights the practical value of the proposed method. Therefore, to better evaluate, we apply our method to a real-world application, i.e., unplanned ICU patient readmission prediction, to test if any performance gain can be achieved, compared with several baseline embedding methods.

We conduct this experiment based on Lin *et al.*'s work (Lin et al., 2018), which leverages the embeddings of medical concepts (in the form of ICD-9 codes) in their method and achieves the state-of-the-art performance. According to their claim, incorporating embeddings of medical concepts can benefit the prediction performance greatly. In this experiment, we test the 30-day unplanned ICU patient readmission prediction performance with different network embeddings for the ICD-9 ontology.

3.4.1 Dataset and Second View Construction

In this experiment, we follow the data preprocessing procedure of previous work (Harutyunyan et al., 2017; Lin et al., 2018; Lu et al., 2019), and generate a dataset of $48,410$ ICU stay records out of the freely available MIMIC-III database (Johnson et al., 2016). The task is to predict whether or not a patient in an ICU stay will be readmitted within 30 days after discharge.

We take the transitive closure of ICD-9 as the first and main view. We first transform the short textual descriptions of nodes into one-hot representations, and compute the cosine similarities between them. If the cosine similarity between two nodes is greater than an empirical threshold 0.7, we create a link between them in the second view of ICD-9.

3.4.2 Baselines

Apart from the baselines used in the link prediction and node clustering task, we add one more strong baseline method, i.e., Poincaré (Nickel and Kiela, 2017), as the Poincaré method proves to be particularly effective in embedding hierarchical data, such as the ICD-9 ontology.

Method	Acc	Pre-0	Pre-1	Re-0	**Re-1**	**A.R**	A.P
Poincaré	0.7223	0.9035	0.3740	0.7361	0.6655	0.7786	0.4827
GAE	0.7052	0.9061	0.3597	0.7089	0.6901	0.7712	0.4588
VGAE	0.7042	0.9007	0.3554	0.7127	0.6684	0.7653	0.4444
ARGA	0.7075	0.9126	0.3654	0.7057	0.7150	0.7757	0.4593
ARVGA	0.6966	0.9056	0.3518	0.6973	0.6934	0.7693	0.4519
MRGAE	0.7094	0.9157	0.3687	0.7055	**0.7259**	**0.7807**	0.4770

*Acc: Accuracy, Pre: Precision, Re: Recall, A.R: AUC under ROC, A.P: AUC under PRC

Table 4: Performance on 30-day unplanned ICU patient readmission prediction.

3.4.3 Experiment Settings

We use the same metrics with Lin *et al.*'s work (Lin et al., 2018). The *area under the Receiver Operating Characteristics curve* (AUC or A.R) is the main metric for evaluation. The recall rate of positive cases (Re-1), i.e., sensitivity, is also important in screening real patients. Additional metrics are reported, but they can be unstable and better be used for additional evaluation. Lin *et al.* use the embeddings for ICD-9 codes as part of their input. We replace the embeddings for ICD-9 with different methods.

3.4.4 Results

As shown in Table 4, our proposed method achieves the best AUC score of 0.7807 with the highest sensitivity score of 0.7259. It is worth mentioning that the best reported AUC of Lin *et al.* is 0.791, but this is unfair to compare with since all the embeddings in the table are trained from the ICD-9 only, while the Claims embeddings (Choi et al., 2016) they use are trained from millions of textual data.

4 Related Work

Recently, researchers use specifically designed encoders to aggregate the local neighborhood information of nodes, to generate low-dimensional embeddings. For example, GAE and VGAE (Kipf and Welling, 2016b) are two methods that use graph convolution networks (GCN) (Kipf and Welling, 2016a) as the encoder. VGAE uses the Gaussian distribution as a prior and pushes the learned representations close to this prior by incorporating a KL divergence penalty. ARGA and ARVGA incorporate an adversarial regularization framework for the same purpose, which is essentially similar in spirit to VGAE. Actually, incorporating adversarial regularization terms and matching the latent representations to a prior distribution are particularly useful for generating robust and meaningful representations when dealing with real-world complex graph data, which is first proposed by Adversarial Autoencoder (AAE) (Makhzani et al., 2015). We extend the adversarial regularization framework to a multiview scenario where the distribution consistency across graph space and node content space are to be preserved. But unlike previous methods like DBGAN (Zheng et al., 2020) and DANE (Gao and Huang, 2018) which aim to reconstruct the node content directly, we focus on the semantic relatedness among them.

5 Conclusion

In this study, we propose a novel network embedding method, i.e., MRGAE, which models the consistency of node representations across two specific views of networks. To achieve this, we incorporate a multiview adversarial regularization module and specifically design a loss function for joint optimization. We conduct extensive and diverse experiments for evaluation, and the results demonstrate the superb performance of the proposed method.

Acknowledgements

This research is partly supported by NSF grant CNS-1747798 to the IUCRC Center for Big Learning.

References

Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 891–900. ACM.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep neural networks for learning graph representations. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jonathan Chang and David Blei. 2009. Relational topic models for document networks. In *Artificial Intelligence and Statistics*, pages 81–88.

Youngduck Choi, Chill Yi-I Chiu, and David Sontag. 2016. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41.

Quanyu Dai, Qiang Li, Jian Tang, and Dan Wang. 2018. Adversarial network embedding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hongchang Gao and Heng Huang. 2018. Deep attributed network embedding. In *IJCAI*, volume 18, pages 3364–3370.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

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

William L Hamilton, Rex Ying, and Jure Leskovec. 2017b. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, and Aram Galstyan. 2017. Multitask learning and benchmarking with clinical time series data. *CoRR*, abs/1703.07771.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035.

Thomas N Kipf and Max Welling. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Thomas N Kipf and Max Welling. 2016b. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Yu-Wei Lin, Yuqian Zhou, Faraz Faghri, Michael J Shaw, and Roy H Campbell. 2018. Analysis and prediction of unplanned intensive care unit readmission using recurrent neural networks with long short-term memory. *bioRxiv*, page 385518.

Qiuhao Lu, Nisansa de Silva, Sabin Kafle, Jiazhen Cao, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Brent Hailpern, Berthold Reinwald, and Yunyao Li. 2019. Learning electronic health records through hyperbolic embedding of medical ontologies. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 338–346.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347.

Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM.

Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. 2018. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2609–2615. AAAI Press.

Jiwoong Park, Minsik Lee, Hyung Jin Chang, Kyuewang Lee, and Jin Young Choi. 2019. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6519–6528.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine*, 29(3):93–93.

Han Shi, Haozheng Fan, and James T Kwok. 2019. Effective decoding in graph auto-encoder using triadic closure. *arXiv preprint arXiv:1911.11322*.

Lei Tang and Huan Liu. 2011. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234. ACM.

Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. 2015. Network representation learning with rich text information. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Shuai Zheng, Zhenfeng Zhu, Xingxing Zhang, Zhizhe Liu, Jian Cheng, and Yao Zhao. 2020. Distribution-induced bidirectional generative adversarial network for graph representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7224–7233.