DOI: 10.53469/jtpes.2024.04(05).03

Multiplex Network Representation Learning Based on Graph Neural Network

Lianwei Li [1*], Peiyuan Yang[2], Han Lei[3], Baoming Wang[4], Zhou Chen[5]

[1]Computer Science, The University of Texas at Arlington, Arlington, USA
[2]Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[3]Computer Science Engineering, Santa Clara University, Santa Clara, USA
[4]Electrical and Computer Engineering,University of Illinois at Urbana-Champaign,Urbana, IL, USA
[5]Software Engineering, Zhejiang University, Hangzhou, China
[*]*Corresponding author: Lianwei Li [E-mail: llwlewuss@gmail.com]*

Abstract: ***Network representation learning is receiving increasing attention from scholars. Among them, methods based on graph neural networks have become particularly popular. However, most existing methods currently only focus on networks with a single type of relations. In the real world, networks contain a wealth of diverse information, with multiple types of relationships between nodes. In this paper, we propose a graph neural network-based multiplex network representation learning model (GNMRL). We model nodes within each layer of the multiplex network by aggregating neighbor information. Additionally, since nodes are influenced by different network layers, we integrate node interaction information across network layers. We conducted systematic experiments on four datasets, and the results show that GNMRL outperforms other comparison methods in both link prediction and node classification tasks.***

Keywords: Multiplex network; Network Representation Learning; Graph Neural Networks.

1. INTRODUCTION

Network representation learning [1] is a method of representing network nodes as low-dimensional vectors. It preserves both the structural characteristics and semantic characteristics of the network, allowing the learned node vectors to exhibit significant performance in many downstream network analysis tasks, such as node clustering [5], link prediction [2-3] and node classification [4]. Most network representation learning methods, including DeepWalk [6], node2vec [8], and GCN [20], focus on single-dimensional networks, where network nodes have only one type of relation. However, real-world network structures are diverse, and nodes have multiple types of relations. For example, in social networks, users can be connected through friendships, work relationships, or financial transactions. In a video network, movies can be connected based on the same director or actor, or by similar genres. These networks can be referred to as multiplex networks [9-10], where each layer represents a type of relations between nodes. Nodes within the same layer can influence each other, and there can also be interactions between layers. Due to the complexity of multiplex networks, single-layer network representation learning methods are no longer effective in representing network nodes. Therefore, how to fully analyze and explore node information in multiplex network is a significant issue.

In the paper, we propose a multiplex network embedding model based on graph neural networks (GNMRL), which can better learn and store node information in multi-relation networks. Within each layer network, nodes are modeled by aggregating neighbor information based on graph neural networks. Between different layers, considering that there are also interactions between layers, we use attention mechanisms [12] to learn the general representations of nodes. Furthermore, GNMRL organically combines the layer-specific node representations with the general representations across layers, effectively mining the information in multiplex networks. In the paper, our contributions can be concluded as follows: We propose a multiplex network representation learning model based on graph neural networks (GNMRL), which can learn both inner-layer node information and capture feature information between cross-layer nodes. We introduce graph neural networks to represent nodes in the network, modeling inner-layer and cross-layer node interactions through aggregation and attention mechanisms. We evaluate GNMRL model on several network datasets for two downstream tasks. Results indicate that GNMRL model exhibits prominent performance.

2. RELATED WORK

2.1 Network Representation Learning

Research on network representation learning (or network embedding [11]) has been rapidly evolving in recent years. The objective is to acquire low-dimensional embeddings of network nodes for subsequent analysis tasks like visualization tasks, node clustering, classification, etc. Early representation learning works mostly relied on random walk-based methods. Inspired by word embedding techniques, DeepWalk [6] generates node walk sequences through random walks, where nodes in the sequences are analogized to words in sentences, and then utilizes the skip-gram model to generate node vectors. Building upon this, LINE [7] learns first and second-order similarity, demonstrating advantages in large-scale networks. Node2vec [8] improves the strategy of random walks, preserving more neighbor information. For heterogeneous networks, Metapath2vec [13] constructs heterogeneous neighbors of nodes based on metapath-based random walks. There are also methods leveraging deep learning for network embedding. SDNE [14] introduces deep autoencoders, effectively capturing nonlinear network structure information. DANE [15] extends SDNE to support attribute networks. Inspired by Generative Adversarial Networks (GAN) [16], GraphGAN [17] designs generators and discriminators for network embedding. Recently, Graph Neural Networks (GNN) have become popular. Network representation learning based on GNN, such as GCN [20], GAT [21], GraphSAGE [22], DGI [23], etc., have shown outstanding performance compared to traditional methods.

Since the aforementioned methods can only handle single-layer networks, many scholars have been devoted to researching multiplex network representation learning for better handling the diversity of real-world networks. PMNE [24] introduces Node2vec into multiplex networks based on proximity. MNE [25] and MELL [26] propose hierarchical embedding and common embedding to preserve features within and between node layers. DMGI [10] applies the idea of DGI and introduces maximizing mutual information into multiplex networks with attributes. GATNE [9] proposes transductive and inductive learning methods to handle the analysis of heterogeneous multiplex attribute networks. SSAMN [27] obtains node embeddings of attributed multiplex networks through feature mapping and homogeneity analysis.

2.2 Graph Neural Networks

Graph Neural Networks (GNN) [18-19] aim to learn and extract useful information from graph data, enabling effective performance across diverse downstream tasks like graph classification, node classification, etc. In recent years, GNN have played a significant role in various fields. Inspired by Convolutional Neural Networks (CNN) [28], GCN [20] applies convolutional operations from grid data to graph data to aggregate neighbor information of nodes, thereby capturing the overall graph information. In GAT [21], attention mechanisms are introduced to allocate varying weights to neighborhoods, enabling the model to better capture complex relationships and structures within the graph. GraphSAGE [22] adopts an inductive learning approach, utilizing both structural and feature information of nodes, sampling neighbors randomly, and generating vector representations of target nodes. GAE [29], on the other hand, utilizes autoencoders to reconstruct graph structure information and learn latent node representations. DGI [23] draws inspiration from contrastive learning, obtaining node embeddings by maximizing mutual information.

3. METHODOLOGY

3.1 Problem Definition

Table 1: Notations

Notation	Description
$\mathcal{G}$	A multiplex network
v_i	A node in $\mathcal{G}$
R	The number of edge types
$\mathcal{V}, \mathcal{E}$	The set of nodes/edges in $\mathcal{G}$
n	The number of nodes
$\mathrm{N}_{i,r}$	The neighbor set of node in layer r
K	Aggregate depth
d	The dimension of node representation

We first introduce some notations and definition. The notations are shown in Table 1.

Given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where denotes a set of nodes and is a set of R type edges $\{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_R\}$. So $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_R\}$, where is a network with edge type r.

In a multiplex network $\mathcal{G}$, the task of multiplex network representation learning involves generating an embedding of each node v_i, where the function f: $\mathcal{V} \rightarrow R^d$, $d \ll |\mathcal{V}|$.

3.2 Inner-layer Interactions

To capture the inner-layer interactions, we apply graph neural network to learn the vector of node on layer r. Based on GraphSAGE, we generate node representations by aggregating information from neighboring nodes:

$$u_{i,r,k} = Aggregate(\{u_{j,r,k-1} | v_j \in N_{i,r}\}) \quad (1)$$

Where represents the set of neighbors of node under layer r, is the depth of aggregation, $1 < k < K$. There are various types of functions, such as max-pooling aggregator, mean aggregator, LSTM aggregator. In this paper, we use the mean aggregator:

$$u_{i,r,k} = act(W_{r,k} \cdot \text{mean}(\{u_{j,r,k-1} | v_j \in N_{i,r}\})) \quad (2)$$

Where is non-linear activation function. After rounds of iteration, represents the vector embedding of node under edge type r.

3.3 Cross-layer Interactions

In the multiplex network, the node under edge type is also influenced by other layers of the network. We denote the interaction effects brought to node by cross-layer networks as:

$$u_{i,r} = \sum_{r'=1,2,...,R} b_{r',r} \cdot u_{i,r'} \quad (3)$$

Where the weight represents the degree of influence of edge type on edge type r. Typically, networks of different relationship types are influenced differently by each other. If two network structures are similar, the connections between the two networks tend to be tighter, and the degree of mutual influence is relatively higher. Therefore, we introduce attention mechanism to learn the influence scores corresponding to different edge types:

$$p(r'|r) = tr(w_r^T X w_{r'}) \quad (4)$$

$$b_{r',r} = \frac{\exp(p(r'|r))}{\sum_r \exp(p(r'|r))} \quad (5)$$

Where is the parameter to be learned, denotes the trace of a matrix. Then, attention coefficients are obtained through softmax.

3.4 Combining Information

In this section, we organically combine inner-layer representation information with cross-layer representation information to fully preserve the characteristics of nodes in multiplex networks. When the aggregation depth is and the edge type is r, the inner-layer node is represented as $u_{i,r,k}$, and the cross-layer node representation is

$$u'_{i,r,k} = \sum_{r'=1,2,...,R} b_{r',r} \cdot u_{i,r',k} \quad (6)$$

We obtain new node representation after aggregating the two, where is a hyperparameter:

$$\tilde{u}_{i,r,k} = (1-\alpha) \cdot u_{i,r,k} + \alpha \cdot u'_{i,r,k} \quad (7)$$

The unified representation of the layer node is:

$$\tilde{u}_{i,k+1} = \sigma(\mathrm{W}^k \cdot concat(\tilde{u}_{i,r,k}, r \in R)) \tag{8}$$

Where is the activation function. Is the final node representation after rounds of aggregation. To train the gnmrl model, negative sampling are used to improve the loss function:

$$L = -\log \sigma\left(\tilde{u}_{N_i}{}^T \cdot \tilde{u}^r_{N_j}\right) - \sum_{h \in H^r_{N_j}} \log \sigma\left(-\tilde{u}_h{}^T \cdot \tilde{u}^r_{N_j}\right) \tag{9}$$

Where is neighboring nodes set of node v_i, is the negative sampling set, including nodes not belonging to the set N_j, and is the nonlinear function. The overall framework of GNMRL is depicted in Figure 1.

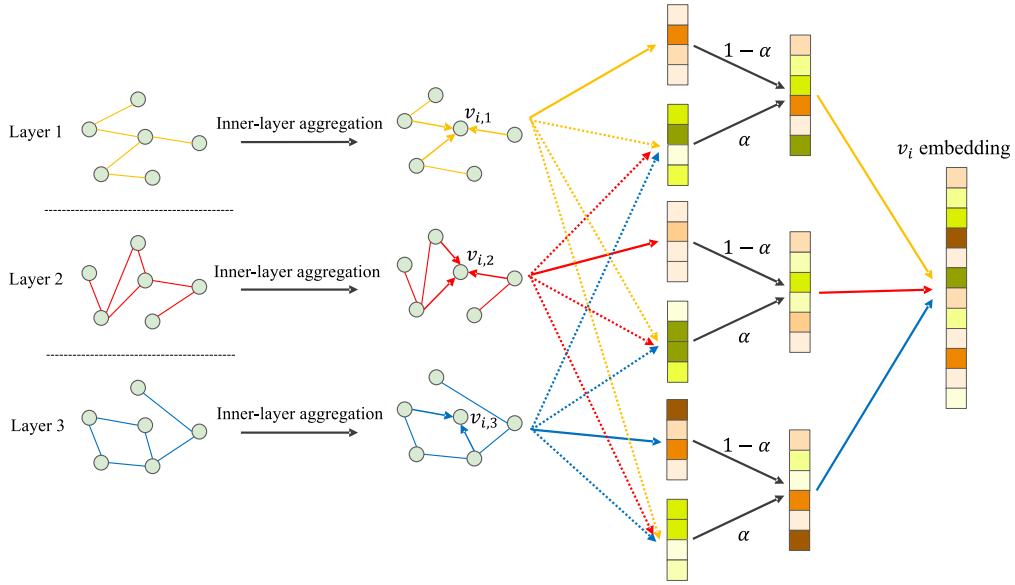

Figure 1: The overall framework of GNMRL

4. EXPERIMENTS

4.1 Datasets

We select four real datasets for experimental validation. The front networks are from Manlio De Domenico's research, covering social, medical, and biological domains, respectively. The fourth dataset is from YouTube, encompassing various types of interaction relationships among video users. Particular statistics are provided in Table 2.

Table 2: Statistics of datasets

Dataset	Layers	Nodes	Edges
Vickers	3	29	740
CKM	3	246	1,551
C.elegans	3	279	5,863
YouTube	4	2,000	689,515

Vickers: The dataset was obtained by interviewing some grade students at an Australia school. Students were asked three questions: Who do you play well with in this class? Who are bosom female friends in this class? Who do you want to work with? The three questions represent different relations.

CKM: The dataset pertained to medical advancements, and consultations were conducted with physicians in four Illinois towns: Peoria, Bloomington, Quincy, and Galesburg. The focus was on understanding how network connections influenced physicians' adoption of a new medication.

C.elegans: The dataset was collected from Biological General Repository for Interaction Datasets (BioGRID), which showed caenorhabditis elegans connectome. It included various synaptic junctions, namely polyadic ("PolySyn"), chemical monadic ("MonoSyn") and electric ("ElectrJ").

YouTube: The dataset consists of different interactions between video users from YouTube. The relation types include encompass friends, mutual friends, shared subscriber, and shared favorite videos.

4.2 Baseline Methods

Our baseline methods are mainly divided into two categories: single-layer network representation learning and multiplex network representation learning. For single-layer network representation learning methods, we primary employ the following:

DeepWalk: DeepWalk [6] is a graph data mining method that combines random walks with word2vec. It treats the node sequences generated by random walks as sentences composed of words, and employs the skip-gram model to represent nodes as vectors containing latent information.

LINE: LINE [7] is an embedding method suitable for large-scale networks. It considers both first-order and second-order proximities of nodes while preserving both local and global features of nodes.

Node2vec: Node2vec [8] is an algorithm based on DeepWalk with improvements. It introduces parameters to control the paths of random walks, better exploring properties such as node communities.

GCN: GCN [20] is a classical method in network representation learning. It is graph neural network that utilizes convolutional operations to aggregate attributes of neighboring nodes onto central nodes, thereby obtaining node embeddings. For multiplex network representation learning methods, we primarily compare with two methods: PMNE and MNE.

PMNE: PMNE [24] proposes three different methods to generate a unified node embedding. These three methods are denoted as PMNE(n), PMNE(r), and PMNE(c), respectively. We compare our method with the third method.

MNE: MNE [25] introduces high-dimensional unified embeddings and low-dimensional hierarchical embeddings, and learns them through optimization with negative sampling.

4.3 Parameter Settings

To be fair, we set the dimensionality of all method-generated vectors to 128. For methods based on random walks, we set the window size to 10, the number of negative samples to 5, the number of walks to 30, and the length of each walk to 20. For Node2vec and PMNE methods, we utilized the optimal hyperparameter settings as reported in the original paper for model training. For our model, the aggregation depth is set to 1.

4.4 Link Prediction

Link prediction is a downstream analysis task in which the future appearance of new links is predicted based on the existing topology of a graph network. We randomly delete a portion of links and on residual graph train the model to predict the presence of these links. Specifically, we randomly remove 10% of the connected edges as positive link samples and arbitrarily select the same proportion of unconnected edges as negative link samples, creating this test set to evaluate model performance. We use the usual evaluation metric ROC-AUC to measure the performance of link prediction. Table 3 displays comparative results of various methods across four datasets.

Drawing from the experimental results, we can draw the following observations:

(1) Across all four datasets, GNMRL consistently outperforms other comparative methods. This suggests that our model can proficiently learn both the structural features of nodes within the network and capture the influence of information from different networks.

(2) Single-layer network representation learning methods (such as LINE, Node2vec, GCN) only pay attention to the structural features of nodes within a single network, neglecting the existence of correlations between different networks. The learned node information is relatively one-sided, leading to poorer performance compared to GNMRL.

(3) Multiplex network representation learning methods (PMNE, MNE) consider the differences and influences between different networks. However, they simply introduce common node embeddings without effectively integrating them with hierarchical node embeddings within the network. Consequently, their performance is slightly inferior to GNMRL.

Table 3: Experimental results on link prediction

	ROC-AUC			
	Vickers	CKM	C.elegans	YouTube
DeepWalk	0.6636	0.7756	0.7630	0.7148
LINE	0.6047	0.7395	0.7128	0.6951
Node2vec	0.6658	0.7866	0.7758	0.7339
GCN	0.7593	0.8426	0.7884	0.7640
PMNE	0.5944	0.5765	0.6043	0.5583
MNE	0.7857	0.7792	0.7801	0.7814
GNMRL	**0.8009**	**0.8524**	**0.8225**	**0.7982**

4.5 Node Classification

Node classification is a downstream task where the goal is to predict the categories of unknown labels based on known node labels. We use CKM as the experimental dataset for this task and randomly hide 30% of node labels as the test set, with the remaining labeled nodes used for training. We train a logistic regression classifier on the labeled node training set, repeating the experiment 10 times and evaluating the average performance for each method. We use Macro-F1 and Micro-F1 to measure performance. Figure 2 illustrates the comparative results of each method.

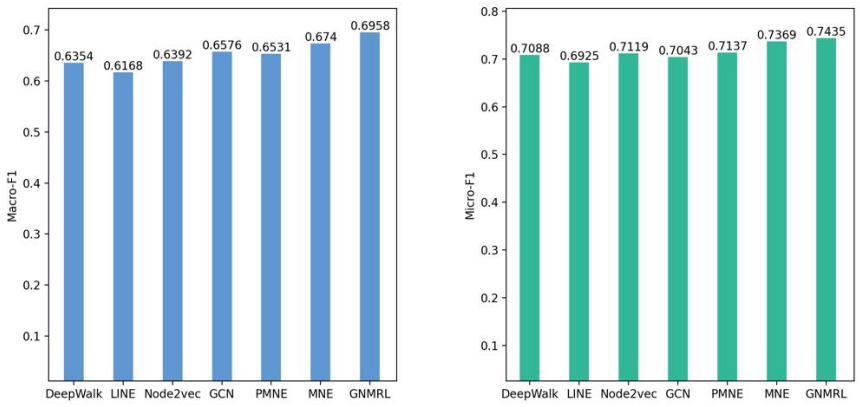

Figure 2: Experimental results of node classification on CKM dataset

From above chart results, we can conclude the following:

Our GNMRL model demonstrates superior performance compared to the comparative methods. GNMRL outperforms single-layer network embedding methods such as Node2vec and GCN significantly, indicating the necessity of retaining information across layers of networks. GNMRL outperforms multiplex network embedding

methods like MNE, suggesting that GNMRL not only effectively learns the interaction information across layers of networks but also integrates information within layers and across layers in a more optimal manner.

5. CONCLUSION

In this paper, we propose a multiplex network representation learning method based on graph neural networks GNMRL. Building upon graph neural networks, we capture the interaction information among nodes within layers while introducing attention mechanisms to learn the influence of cross-layer networks on nodes. We assess the performance of our model on two downstream tasks: link prediction and node classification. Experimental results demonstrate that GNMRL outperforms other methods. In the future, we aim to incorporate richer information such as node attributes and node types into our model.

REFERENCES

[1] Zhang D, Yin J, Zhu X, et al. Network representation learning: A survey[J]. IEEE transactions on Big Data, 2018, 6(1): 3-28.
[2] Taskar B, Wong M F, Abbeel P, et al. Link prediction in relational data[J]. Advances in neural information processing systems, 2003, 16.
[3] Butts C T. Revisiting the foundations of network analysis[J]. science, 2009, 325(5939): 414-416.
[4] Bhagat S, Cormode G, Muthukrishnan S. Node classification in social networks[J]. Social network data analytics, 2011: 115-148.
[5] Zhang Y, Wu J, Zhou C, et al. Multi-view fusion with extreme learning machine for clustering[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(5): 1-23.
[6] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 701-710.
[7] Tang J, Qu M, Wang M, et al. Line: Large-scale information network embedding[C]//Proceedings of the 24th international conference on world wide web. 2015: 1067-1077.
[8] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016: 855-864.
[9] Cen Y, Zou X, Zhang J, et al. Representation learning for attributed multiplex heterogeneous network[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 1358-1368.
[10] Park C, Kim D, Han J, et al. Unsupervised attributed multiplex network embedding[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 5371-5378.
[11] Cui P, Wang X, Pei J, et al. A survey on network embedding[J]. IEEE transactions on knowledge and data engineering, 2018, 31(5): 833-852
[12] Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning[J]. Neurocomputing, 2021, 452: 48-62.
[13] Dong Y, Chawla N V, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017: 135-144. Ma X, Z140.
[14] Wang D, Cui P, Zhu W. Structural deep network embedding[C]//Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016: 1225-1234.
[15] Gao H, Huang H. Deep attributed network embedding[C]//Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)). 2018. Ma J, Z9.
[16] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.
[17] Wang H, Wang J, Wang J, et al. Graphgan: Graph representation learning with generative adversarial nets[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).
[18] Zhou J, Cui G, Hu S, et al. Graph neural networks: A review of methods and applications[J]. AI open, 2020, 1: 57-81.
[19] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. IEEE transactions on neural networks and learning systems, 2020, 32(1): 4-24.
[20] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
[21] Casanova P, Lio A R P, Bengio Y. Graph attention networks[J]. ICLR. Petar Velickovic Guillem Cucurull Arantxa Casanova Adriana Romero Pietro Liò and Yoshua Bengio, 2018.

[22] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30.
[23] Velickovic P, Fedus W, Hamilton W L, et al. Deep graph infomax[J]. ICLR (Poster), 2019, 2(3): 4.
[24] Liu W, Chen P Y, Yeung S, et al. Principled multilayer network embedding[C]//2017 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, 2017: 134-141.
[25] Zhang H, Qiu L, Yi L, et al. Scalable multiplex network embedding[C]//IJCAI. 2018, 18: 3082-3088.
[26] Matsuno R, Murata T. Mell: effective embedding method for multiplex networks[C]//Companion Proceedings of the The Web Conference 2018. 2018: 1261-1268.
[27] Sadikaj Y, Rass J, Velaj Y, et al. Semi-Supervised Embedding of Attributed Multiplex Networks[C]//Proceedings of the ACM Web Conference 2023. 2023: 578-587.
[28] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern recognition, 2018, 77: 354-377.
[29] Kipf T N, Welling M. Variational graph auto-encoders[J]. arXiv preprint arXiv:1611.07308, 2016.