DEEPLEX: A GNN FOR LINK PREDICTION IN MULTIPLEX NETWORKS

Vamsi K. Potluru *Robert E. Tillman* *Prashant Reddy* *Manuela Veloso*

SIAM Workshop on Network Science 2020
July 9–10 · Toronto

Abstract

Graph neural networks have recently gained attention due to their performance in many classification and prediction tasks. In particular, they are used for node classification and link prediction with a wide range of applications in social networks, biomedical datasets and financial transaction graphs. Most existing work is primarily focused on the monoplex setting where we have access to a network with only a single type of connection between entities. However, in the multiplex setting, where there are multiple types of connections between entities, tasks such as link prediction have been shown to have stronger performance when information from other connection types is taken into account. We notice a critical issue with a previous approach, MNE, based on embedding models related to the way information is aggregated across layers and propose a novel method DeePlex which aggregates layer information using a neural network. Experimental results on real-world multiplex networks show that our approach outperforms and overcomes the limitations of state of the art approaches.

Introduction One of the fundamental problems in network analysis is link prediction. Quite a few approaches have been proposed to solve this problem which can be broadly classified into two categories: (A) Supervised methods (B) (unsupervised) Heuristic methods [5]. Heuristic approaches while relatively simple and easy to compute in most cases are surprisingly good in performance over a wide-range of network domains. Examples in this class are common neighbors (CN), Adamic-Adar [1], and Katz. In many real-world settings, two nodes may be connected over multiple facets. For example, consider the trading patterns across the countries of the world where two countries are connected if they trade in a particular commodity like say oil. When we consider all commodities, a rich pattern of connectivity between the nodes arises; this notion is called *multiplexity* and results in a multiplex network. A straightforward approach to link prediction would be to treat each layer of the multiplex network separately and apply either supervised or heuristic methods. However, this may not be ideal because we are not utilizing the rich structure of correlations across the various layers that may exist in the network.

We propose to use the graph neural network (GNN) framework to embed the sub-graphs corresponding to the various layers of the networks and jointly learn a classifier which weighs these embeddings to predict the existence of links in the corresponding layer. To the best of our knowledge, this is the first work which utilizes a GNN for the multiplex link prediction problem. Additionally, we utilize label information from the other layers.

SEAL We use an existing approach for obtaining node embeddings for a given network namely SEAL [8]. It learns the node structural features based on the local enclosing subgraphs. SEAL is flexible in terms of the particular GNN architecture that it uses and we retain DGCNN [9] as our default. Experimental results show this approach to be significantly better than other link prediction algorithms such as node2vec (N2V) [2], spectral clustering (SPC) and variational graph auto-encoders (VGAE) [4].

MNE A popular approach for link prediction in multiplex networks is the recently proposed embedding approach MNE [7]. The idea is to learn a common "base" embedding which utilizes all of the links across all the layers as well as individual node embeddings for each of the layers. These two types of embeddings are combined via a linear transformation which is learned from the training data. Concretely, given a network of layers $G_1, \ldots, G_L$ where we have L layers and $G_i = (N_i, E_i)$ corresponding to sets of N_i nodes and E_i edges, MNE learns a node embedding: $v_n^i = b_n + w^i \dot{X}^{i^T} u_n^i$ where $X^i \in R^{s \times d}$, b_n correspond to the base node embedding and u_n^i to the individual node embedding for the layer.

DeePlex We set out to take the best elements of the previous two approaches while potentially avoiding the limitations of either approach to propose a novel **Dee**p multi**Plex** GNN model (**DeePlex**). Our model consists of the embeddings from SEAL utilizing the DGCNN ar-

Model	Vickers	CKM	LAZEGA	C.ELEGANS	EUAIR
DeepWalk	0.821 (0.030)	0.781 (0.008)	0.780 (0.007)	0.821 (0.006)	0.430 (0.020)
LINE	0.676 (0.011)	0.637 (0.012)	0.695 (0.006)	0.732 (0.006)	0.429 (0.020)
Node2Vec	0.821 (0.030)	0.781 (0.008)	0.780 (0.007)	0.820 (0.006)	0.430 (0.020)
PMNE (n)	0.810 (0.032)	0.917 (0.008)	0.792 (0.009)	0.843 (0.003)	0.448 (0.011)
PMNE (r)	0.844 (0.025)	0.904 (0.008)	0.813 (0.007)	0.835 (0.007)	0.838 (0.009)
PMNE (c)	0.837 (0.029)	0.847 (0.016)	0.797 (0.011)	0.824 (0.009)	0.500 (0.005)
Common Neighbor (CN)	0.799 (0.011)	0.877 (0.006)	0.809 (0.007)	0.869 (0.002)	0.913 (0.006)
Jaccard Coeficient (JC)	0.778 (0.007)	0.873 (0.006)	0.826 (0.007)	0.833 (0.001)	0.882 (0.004)
Adamic/Adar (AA)	0.803 (0.019)	0.875 (0.013)	0.814 (0.008)	0.881 (0.001)	0.918 (0.006)
MNE	0.871 (0.014)	0.900 (0.010)	0.839 (0.013)	**0.910** (0.006)	0.400 (0.020)
DeePlex	**0.911** (0.014)	**0.935** (0.011)	**0.847** (0.004)	0.907 (0.009)	**0.978** (0.008)

Table 1: Link prediction accuracy for DeePlex using architecture choice (A) and the other methods. All numbers are averaged AUC score based on five-fold cross validation. Standard deviations are reported in the parentheses.

chitecture while also proposing to combine them similar to the MNE model with the caveat that it is a non-linear combination which could help it identify a general combination function and also prune out the irrelevant layers. In this work, we explored individual feed-forward layers for the deep learning models.

Experiments We used the following datasets which are typically used for multiplex networks: Vickers (3 layers, 29 nodes, 740 edges), CKM (3 layers, 246 nodes, 1551 edges), Lazega (3 layers, 71 nodes, 2223 edges), C.Elegans (3 layers, 249 nodes, 5863 edges) and EUAIR (37 layers, 450 nodes, 3588 edges). Notice that we perform significiantly better than MNE and the rest of the competing methods or on par as shown in table 1. In particular, for EUAIR where there are dozens of layers, MNE struggles to learn to utilize the layer information. This is probably due to the base layer getting overwhelmed by the conflicting information when aggregating across all the layers. Our models are able to circumvent the issue by learning either layer dependent classifiers and/or filtering out the layers which are not relevant for a given layer.

Conclusions We proposed a novel method for link prediction in multiplex networks. It improves on the previous approach based on embeddings learned from random walks on the network (MNE) and highlighted a critical issue with the base embedding across all the layers used in this approach. Promising work includes incorporating the recently developed differentiable k-NN objects [3] to automatically select the relevant layers for a given layer's prediction task. Alternatively, attention based mechanisms [6] can be utilized to select the relevant layers. Extensions include tackling the node classification and heterogenous network settings.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

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

[3] A. Grover, E. Wang, A. Zweig, and S. Ermon. Stochastic optimization of sorting networks via continuous relaxations. *arXiv preprint arXiv:1903.08850*, 2019.

[4] T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[5] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[7] H. Zhang, L. Qiu, L. Yi, and Y. Song. Scalable multiplex network embedding. In *IJCAI*, volume 18, pages 3082–3088, 2018.

[8] M. Zhang and Y. Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, pages 5165–5175, 2018.

[9] M. Zhang, Z. Cui, M. Neumann, and Y. Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.