Explainable DiGCN for Decomposition of Opaque Node Ranking Functions

Vishal Chandra
MIT Lincoln Laboratory, University of Michigan
Lexington, MA
chandrav@umich.edu

***Abstract*—From Snap Score in social media, to ResearchGate score in professional networks, to College Football Playoff ranking in sports, networks everywhere are home to opaque measurements with unknown factors. Thus far, standard techniques in explainable AI, such as Grad-CAM, have allowed the fitting and inspection of models based on black-box data. This work aims to do the same with these black-box rankings in directed networks. The proposed work uses the DiGCN architecture, a directed and multi-scale variant of the vanilla Graph Convolutional Network, for learning-to-rank tasks based on black-box input and output. Then, we extract multi-scale activation maps from the network to determine what factors at each neighborhood size contribute to the ranking of nodes. By leveraging intuitive ideas from existing centrality measures, this work allows the decomposition of black-box node influence functions in a more granular fashion than standard attention map analysis.**

Keywords—graph neural networks, explainable AI, learning-to-rank

I. Introduction

Across social, sports, and academic networks, we encounter opaque metrics that are comprised of unknown factors. Snapchat score measures a user's activity, ResearchGate score measures an academic's influence, and the College Football Playoff (CFP) ranking measures the power of college football teams. None of these metrics has a public formula, and in the case of the human CFP committee, a formula may not even exist. To begin to understand these metrics, models can be fit to their inputs and outputs and inspected further. For graph data, the model is chosen from a variety of Graph Neural Networks (GNNs) architectures. After training, their parameters are inspected using standard explainable AI techniques based on class activation mapping (CAM). Through this training and analysis process, we can begin to decompose the factors behind these opaque graph metrics and extract relevant substructures from these networks. To give an example, we might like to highlight the that the five most cited authors cite one another in a circular structure, and that this structure draws high activations in the ranking network.

To train the graph model on influence scores, such as the ones output by these opaque systems, we adjust the output layer from its initial classifier configuration to a single centrality score output. Then, we employ Grad-CAM to view the learned model activations and decompose the opaque centrality measure.

II. Related Work

A. Graph Neural Networks

A message-passing graph neural network serves as the most generalized template for a metric on graph nodes, allowing the aggregation of neighborhood data at different scales, and considering graph topology as well. Here, we compare three different variants of the GNN architecture.

First, the Graph Attention Network (GAT) [1]. This is an immediate candidate for an explainability-focused task in network science, as the attention map serves as an immediate entry for analysis. However, a single layer of the standard GAT implementation captures only 1-hop information around each node. While this receptive field can be enlarged by cascading multiple GAT layers, just as in a CNN, this provides for a less direct analysis of multi-scale activations. The GAT architecture also does not directly account for digraph information, though there are workarounds in encoding that can be employed.

Graph Convolutional Networks (GCN) [2] are another common option for representation learning on graphs, with some notable differences to GATs. This method is based on spectral convolution rather than 1-hop attention, and so can be set for a fixed neighborhood size. Similarly to GATs, however, edge directionality must be represented via a complex Hermitian adjacency matrix or another workaround.

Digraph Inception Graph Convolutional Networks (DiGCN) [3] seem to be best suited to this application, with natural support for directed graphs and with multi-scale information aggregated within a single layer. This allows us to simultaneously analyze activations at each scale, rather than the sequential layer-by-layer analysis characteristic of CNNs that would also be required in a GCN or GAT architecture. Because of its multi-scale consideration, DiGCN also encodes more topological information about the network than GAT, which is an enormous consideration for learning-to-rank.

B. Learning to Rank with GNNs

The Learning to Rank is a task that applies machine learning to ranking tasks, and this has been explored in graph contexts in [6] and [7]. These works extend machine learning ranking (MLR) theory to relational data for ranking nodes, but to the best of our knowledge, this has not been done with multi-scale DiGCNs prior to this work. The work in [7] is also tailored to information retrieval, which is inherently much more dependent on node-level information than on network topology. This is opposite to what we expect in social and academic networks.

C. *Explainability for GCNs*

[5] provides a survey for explainability methods in graph convolutional networks, comparing various activation mapping strategies, as well as other methods for understanding GCN intrinsics. To focus the experiments on probing the fit model, rather than on engineering the model inputs, we pursue only activation mapping, which has become a standard in AI explainability. This allows viewing the numerical values of a slice of the model during inference time.

III. Methods

For learning to rank nodes, first consider that the output layer must be able to output unbounded scores from the node embeddings. This leads us to avoid common activations like sigmoid and SoftMax in favor of a linear layer with no output constraint. This learnable mapping also captures the relationship between network topology and rank, somewhat isolating the multi-scale proximity maps to learn intuitive topological properties of the ranking.

Second, consider that we are training from partial information about the black-box process; often, as in sports networks, we are provided only the rank of nodes rather than their raw values. In training from rank-order statistics, we must either partially recover the lost information of raw values or be agnostic to them entirely. ListMLE loss is designed primarily for this purpose and computes a listwise comparison between positions within rankings.

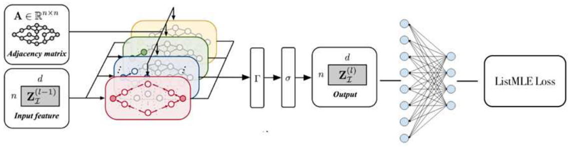
$\mathbf{A} \in \mathbb{R}^{n \times n}$
Adjacency matrix
d
n
$\mathbf{Z}_{\mathcal{I}}^{(l-1)}$
Input feature
Γ
σ
d
n
$\mathbf{Z}_{\mathcal{I}}^{(l)}$
Output
ListMLE Loss

Fig.1. Single block architecture adapted from [3], including ListMLE loss

IV. Experiments

As this work defines a new task on graphs of fitting to an unknown ranking function, and that opaque ranking data is low-volume, synthetic data was generated for training and analysis.

A. *Data Generation*

From the intuitive idea that black-box node rankings have some correlations with centrality measures, we generate a ranking derived from eigenvector, betweenness, and degree centrality. The centrality measures are computed for each node in the standard CORA dataset, then they are combined (reduced) to one measure with SVD. This hybrid centrality score is finally ordered and ranked.

B. *Activation Mappings*

We expect to see components of all three basis centrality measures in the activation mappings. Initial analysis suggests that the network fits to first-order measures like degree and betweenness centrality well, while higher-order proximity information exhibits more global eigenvector-like behavior.

V. Future Work

A. *Multi-Scale via GAT and GCN*

An open avenue for further research is extending this work to GCN and GAT models which do not directly contain multi-features in parallel within the same layer. This would require pulling out sequential activation maps from sequential layers and separating their overlapping information. This is a more difficult task than in DiGCNs and may also require some post-processing before visual analysis due to more complex differing methods necessary to encode edge directionality.

B. *Causality and Propensity*

Any method that seeks to explain the factors behind a black-box process essentially seeks to perform some kind of causal analysis on the data. This analysis has not been performed here, so the conclusions we draw can at best be correlational with the opaque process at hand. To extend this analysis, possible confounders need to be identified in the specific network context and included in the node feature set to condition the learning on those variables. An in-depth explanation of this can be found in [8].

We must also address the question of confounded analysis within the black box itself. Does the CFP committee perform this sort of analysis when considering input factors? How much should our proxy model also include those oversights to remain true to the black box? These questions can be addressed with a propensity score estimation of the opaque process, also left as a future topic here.

Acknowledgements

Edward Kao at MIT Lincoln Laboratory, for his contributions on propensity and for encouraging the initial work.
Mark Newman at the University of Michigan, for his contributions on arbitrary ranking generation.

References

[1] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *arXiv:1710.10903 [cs, stat]*, Feb. 2018, Available: https://arxiv.org/abs/1710.10903

[2] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv:1609.02907* Feb. 2017.

[3] Z. Tong, Y. Liang, C. Sun, X. Li, D. Rosenblum, and A. Lim, "Digraph Inception Convolutional Networks," *Neural Information Processing Systems*, 2020.

[4] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank," *International Conference on Machine Learning*, Jul. 2008.

[5] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability Methods for Graph Convolutional Neural Networks," *IEEE Xplore*, Jun. 01, 2019.

[6] D. Carlos and Longin Jan Latecki, "Rank-based self-training for graph convolutional networks," *Information processing & management*, vol. 58, no. 2, pp. 102443–102443, Mar. 2021.

[7] U. Ergashev, E. Dragut, and W. Meng, "Learning To Rank Resources with GNN," *Proceedings of the ACM Web Conference 2023*, Apr. 2023.

[8] E. K. Kao, "Causal Inference Under Network Interference: A Framework for Experiments on Social Networks," *arXiv.org*, Aug. 28, 2017.