Causal Inference Explanations for Graph Neural Networks

Sahil Kumar[1] **Cláudia Soares**[1] **Francisco Caldas**[1]

[1]NOVA School of Science and Technology, Lisbon, Portugal

Abstract

Explainable Artificial Intelligence has emerged, aiming to enhance the trustworthiness of black box models by devising explanation methods that clarify their inner workings. However, prevalent explanation techniques predominantly leverage correlation and association rather than employing causality, a significant aspect of human comprehension. We propose a novel explanation method grounded in causal inference tailored specifically for Graph Neural Networks. Our approach seeks to illuminate the decision-making process of Graph Neural Networks, thereby augmenting their transparency and trustworthiness. We apply our method to the medical referral problem in healthcare.

1 INTRODUCTION

The use of Deep Learning (DL) models in the medical referral system [Valdeira et al., 2023, Wee et al., 2022, Duarte et al., 2021, Han et al., 2018] as helpers could significantly improve it, but most are black box models that do not explain how they arrived at their predictions. Hence, they cannot be deployed in the actual system because of the lack of transparency necessary for DL model usage in critical environments like healthcare. Thus, explainability becomes a key factor in gaining confidence in the predictions of the model.

However, even though most explanation methods provide transparent and comprehensible explanations, the inherent nature of the explanation mechanism is based on identifying associations and correlations in the data to explain black box predictions. While associations and correlations are important, they might not provide a complete understanding of the underlying mechanisms in a model because "correlation does not imply causation".

We propose CIExplainer, an explanation method that goes beyond simple associations and delves into causality to help identify the cause-and-effect relationships driving the decisions of a black box model. We specifically focus on a Graph Neural Network (GNN) model, which takes as input a graph and returns an embedding of an element of the graph that can be used down the pipeline in other tasks like node classification or link prediction.

2 METHODOLOGY

CIExplainer is a local explanation method dedicated to generating causal inference explanations. By leveraging CIExplainer, we aspire to enhance the interpretability and transparency of GNN-based link prediction models, thereby enabling their trustful deployment in critical decision-making environments like healthcare.

Given a pair of nodes (v, w) extracted from a graph $\mathcal{G}$, a trained GNN model f, and a link prediction probability $\hat{y}_p$ generated by f for that particular pair by sampling a K-hop neighborhood $Ne_K = (\mathcal{V}_N, \mathcal{E}_N)$ of the nodes, CIExplainer yields a subgraph $\mathcal{G}_{EXP} = (\mathcal{V}_{EXP}, \mathcal{E}_{EXP})$ as an explanation for $\hat{y}$, denoting the infered link or not by analysing $\hat{y}_p$. This subgraph contains the top l nodes and the edges between those nodes that caused f to output $\hat{y}_p$. Specifically, $\mathcal{V}_{EXP} \subseteq \mathcal{V}_N$, $\mathcal{E}_{EXP} \subseteq \mathcal{E}_N$, and $|\mathcal{V}_{EXP}| = l$, where $l \in \{1, 2, ..., |\mathcal{V}_N|\}$ is a hyperparameter enforcing the returned explanation to be concise and informative. Our proposed method, CIExplainer, selects the top l nodes by evaluating the causal effect that each node in Ne_K exerts on the prediction $\hat{y}$, then prioritizing the l nodes with the highest causal effect values. This computation of causal effects leverages the potential outcome framework for causal inference at the unit level [Holland, 1986]. In order to address the Fundamental Problem of Causal Inference, we make the assumptions of Temporal Stability and Causal Transience [Holland, 1986].

Let u denote the link prediction $\hat{y}$ that says if there is a link or not between the nodes v and w. Let S be a binary variable representing the cause, t or c, to which u is exposed. We note that to generate a link prediction probability $\hat{y}_p$, a GNN model f only needs the sampled K-hop neighborhood Ne_K from the pair of nodes (v, w). As such, manipulating u consists of manipulating Ne_K. Hence, let c denote the absence of manipulation of Ne_K and let t denote the manipulation of Ne_K, specifically, manipulating the feature values of a node in Ne_K. Maintaining the original Ne_K used to generate $\hat{y}$ is denoted by $S(u) = c$ and it represents the *actual* outcome, while, manipulating the features of nodes in Ne_K is denoted by $S(u) = t$ and it represents the *counterfactual* outcome. Let Y denote the link prediction probability generated by f for some sampled K-hop neighborhood. Then, $Y_c(u)$ denotes the link prediction probability when u is exposed to c, that is, it denotes the original or *actual* link prediction probability $\hat{y}_p$. Whereas, $Y_t(u)$ denotes the link prediction probability when u is exposed to t, that is, *counterfactual* link prediction probability obtained by constructing a *counterfactual* K-hop neighborhood.

In this context, the Temporal Stability assumption holds because the output produced by the model f remains consistent regardless of when the input is provided to f. Essentially, the value $Y_c(u)$ remains unchanged irrespective of the timing of exposing c to u and measuring $Y_c(u)$. Similarly, the Causal Transience assumption is valid because exposing c to u does not alter the overall network structure of Ne_K, and computing the output of f for Ne_K does not change the weights of the f. Consequently, when computing the output for the original network configuration Ne_K, denoted as $Y_t(u)$, prior exposure of u to c does not influence the result. Therefore, the causal effect CE of altering the feature value of a node in Ne_K on the predicted outcome $\hat{y}$, as assessed by Y, can be determined using $CE = Y_t(u) - Y_c(u)$ [Holland, 1986].

Given the aforementioned input conditions, and considering each node possesses n features, our method CIExplainer generates an explanation following Algorithm 1. Concerning binary features, generating counterfactual nodes is relatively straightforward—simply switch the feature value to its opposite state. For a continuous feature with value v, as an initial approach, the counterfactual node can be obtained by changing the value v by a value contained in the set $C = \{\, c \,|\, c \in S \wedge abs(v - c) \in D\}$, where S is a set containing all the possible values from the data set for that feature, $abs()$ is a function that returns the absolute value of the argument and D is a set containing the top d furthest values of S from v as measured by $abs()$. For example, for $\{1, 2, 3\} \in S$ and $v = 0$ the top 2 furthest values are 3 and 2 because $abs(0 - 3) > abs(0 - 2) > abs(0 - 1)$, then $D = \{3, 2\}$. However, when dealing with continuous feature values, there are several challenges, necessitating careful consideration and analysis. As such, we also aim to address these challenges in our ongoing work.

3 RESULTS

We applied CIExplainer to explain the link predictions returned by a GNN model, specifically, GraphSAGE [Hamilton et al., 2017]. The GNN model was trained on a weighted bipartite referral graph connecting 294 general physicians to 839 specialty care doctors through 34 241 edges. The feature set of each node of the graph was composed of the gender and age of the doctor. The weights of the edges correspond to the number of different patients a general physician referred to a specialist.

For $l = 9$, CIExplainer generates the explanation subgraph represented in 1 when the GraphSAGE model predicted a probability of 0.9233932 for a link between the nodes colored in green.

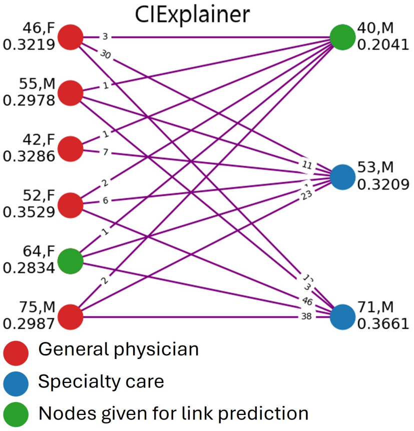

Figure 1: Node description: Age, Gender, Causal Effect.

Observing the explanation subgraph, we see that the specialty care doctor has more direct neighbors than the general physician in the pair given for link prediction. This means they have a bigger impact on the prediction, even though the node with the biggest causal effect is a direct neighbor of the general physician. Another observation is that the node with the biggest causal effect is a senior doctor with big edge weights, meaning he is highly referred to. Since he shares his gender with the specialty care doctor in the pair given for link prediction, he is expected to have a big causal effect on the link prediction.

A plot showing the effects of different values of l is shown in the supplementary material B in figure 2.

For future work, we plan to compare CIExplainer against the existing explanation techniques for GNNs to discern their relative efficacy. As such, both quantitative and qualitative evaluations will be conducted to gauge the effectiveness and comprehensibility of our approach, providing insights into its utility and potential for enhancing the interpretability of GNN-based systems. We will also apply our explanation method to space surveillance awareness.

Acknowledgements

This work was partially supported by FCT I.P., through the strategic project NOVA LINCS (UIDB/04516/2020) and was carried out under Project “Artificial Intelligence Fights Space Debris” Nº C626449889-0046305 co-funded by Recovery and Resilience Plan and NextGeneration EU Funds, www.recuperarportugal.gov.pt. This work was also conducted under a Master’s Student Grant funded through the European Union / Next Generation EU research framework program.

References

Regina Duarte, Qiwei Han, and Claudia Soares. Referral prediction in healthcare using graph neural networks; referral prediction in healthcare using graph neural networks. 2021. doi: 10.1145/nnnnnnn.nnnnnnn. URL `https://europe.naverlabs.com/wp-content/uploads/2021/09/DuarteEtAl2021.pdf`.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. volume 2017-December, 2017.

Qiwei Han, Mengxin Ji, Inigo Martinez De Rituerto De Troya, Manas Gaur, and Leid Zejnilovic. A hybrid recommender system for patient-doctor matchmaking in primary care. 2018. doi: 10.1109/DSAA.2018.00062.

Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81, 1986. ISSN 1537274X. doi: 10.1080/01621459.1986.10478354.

Filipa Valdeira, Stevo Racković, Valeria Danalachi, Qiwei Han, and Cláudia Soares. Extreme multilabel classification for specialist doctor recommendation with implicit feedback and limited patient metadata, 2023.

Chee Keong Wee, Xujuan Zhou, Ruiliang Sun, Raj Gururajan, Xiaohui Tao, Yuefeng Li, and Nathan Wee. Triaging medical referrals based on clinical prioritisation criteria using machine learning techniques. *International Journal of Environmental Research and Public Health*, 19, 2022. ISSN 16604601. doi: 10.3390/ijerph19127384.

Causal Inference Explanations for Graph Neural Networks (Supplementary Material)

Sahil Kumar[1] **Cláudia Soares**[1] **Francisco Caldas**[1]

[1]NOVA School of Science and Technology, Lisbon, Portugal

A CIEXPLAINER ALGORITHM

The algorithm used by the CIExplainer explanation method:

```
**Input:** pair of nodes $(v, w)$, GNN model $f$, link prediction $\hat{y}$ for $(v, w)$
**Output:** explanation subgraph $\mathcal{G}_{EXP}$ containing the nodes that caused $\hat{y}_p$
Sample a $K$-hop neighborhood $Ne_K$ of $(v, w)$;
Compute a link prediction probability for $(v, w)$ in accordance with $\hat{y}$ using $f$ and $Ne_K$;
for each node $v$ of $Ne_K$ do
    for each feature $x_i$ of $v$ do
        Generate a counterfactual node by changing the feature value;
        Compute the causal effect $CE_i$ of changing the feature value;
    end
    The causal effect $CE$ of the node is given by $CE = \max_{i \in \{1,2,\ldots,n\}} CE_i$;
end
Return a subgraph of $Ne_K$ containing the top $l$ nodes with the highest $CE$ and the edges between those nodes;
```

Algorithm 1: CIExplainer explanation generation algorithm

B MEAN SQUARED ERROR BY THE NUMBER OF NODES IN THE EXPLANATION SUBGRAPH

CIExplainer computes the causal effect for all the nodes of the K-hop neighborhood of the pair of nodes given for link prediction. As such, the hyperparameter l used in CIExplainer controls the number of nodes in the explanation subgraph in order to mantain the subgraph relevant and informative by choosing only the most influential nodes.

The true accuracy of the explanations should ideally be evaluated by doctors. However, we calculated the Mean Squared Error (MSE) of the explanations provided by CIExplainer by generating a link prediction using only the explanation subgraph and then comparing it to the original link prediction using the original test graph. The MSE is defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (f(G_{TEST}) - f(G_{EXP}))^2 \tag{1}$$

where n is the number of node pairs in the test set, G_{TEST} is the test graph and G_{EXP} is the explanation subgraph.

Figure 2 shows how the MSE changes according to different values of l when $n = 21$.

Observing figure 2 we note that the MSE decreases when l increases indicating that bigger explanation subgraphs are more

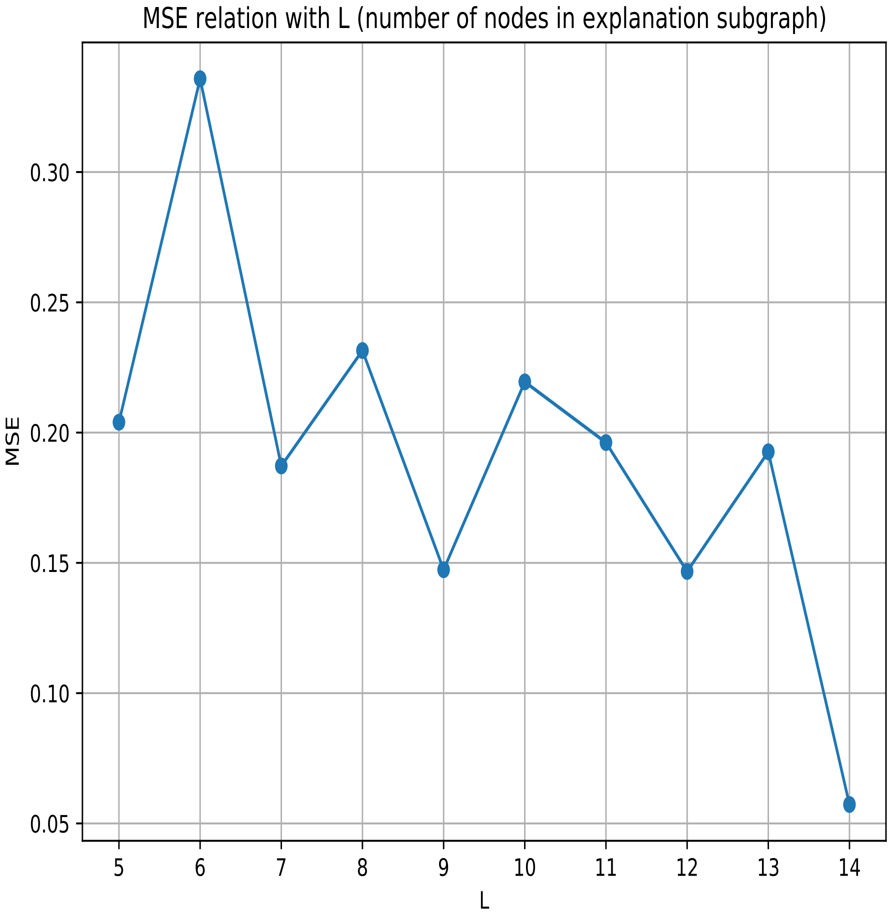

Figure 2: MSE according to he number of nodes in the explanation subgraph.

accurate than smaller explanation subgraphs. However, as the MSE is very low (smaller than 0.5) smaller explanations are preferred considering that they are more informative.