.

Review : Generative Causal Explanations for Graph Neural Networks

Original paper : https://proceedings.mlr.press/v139/lin21d/lin21d.pdf

Lucas Biéchy

Department of Mathematics, University of Paris-Saclay, Orsay Mathematics Laboratory
Email address: lucas.biechy@universite-paris-saclay.fr

GitHub repository : https://github.com/Biechy/Gem

Abstract

This review delves into the paper 'Generative Causal Explanations for Graph Neural Networks' written by Wanyu Lin et al., published in 2021. In order to provide interpretable explanations for any GNNs on various graph learning tasks, this paper introduces Gem, a model-agnostic approach. It achieves a relative increase of the explanation accuracy by up to 30% and speeds up the explanation process by up to 110× as compared to its state-of-the-art alternatives.

This review is divided into two main parts: a summary of the methodology developed in the paper and an implementation by myself. In the summary section, I will elaborate on the causal contribution of Granger, upon which the distillation algorithm is based, as well as the selection of a meta-model known as an explainer. Following this, my implemantation replicates the experimental setup described in the paper with a graph neural network and a graph autoencoder, providing a comparison. But unfortunately the results are rather mixed and do not fully validate Gem.

Keywords. GNN explanations • model-agnostic • graphical models

Contents

1. Introduction

Many challenges in scientific domains, ranging from social networks to biology and chemistry, can often be formulated as problems of learning properties on graphs. In this context, supervised learning on graphs, particularly with Graph Neural Networks (GNNs) [1], has a significant impact on inductive learning. However, GNNs are often perceived as black boxes, lacking transparency and explicit representations of knowledge. To increase confidence in these algorithms, facilitate error analysis, and ensure their proper functioning, enhancing interpretability is essential. With this goal in mind, Lin, W., Lan, H., Li, B. introduces *Gem* in 2021, inspired by the state-of-the-art alternative *PGExplainer* [3], yet surpassing it now, as it no longer requires any prior knowledge of the internal structure of GNNs or graph learning tasks. *Gem* proposes a two-step approach. The first step involves create a new dataset from the original that retain only nodes and edges significantly relevant for prediction. The second step involves using this truncated version as targets to learn a new model that predicts these subgraphs. This additional model, called the `explainer`, enables to find an explanatory subgraph from the original graph. With this, the precision of explanations is improved on average by 30%, with a speed increase of 110 times.

2. From The Paper

2.1. Notation

The notations adhere to the usual conventions of graph theory. Let G denote a graph composed of a set of nodes V and edges E, such that $G = (V, E)$. Associated with this graph is a node feature matrix $X = (x_1, ..., x_V) \in \mathbb{R}^{d\times|V|}$. The paper extends the notion of graphs by introducing and using the *computational graph* $G^c = (V^c, A^c, X^c)$, where V^c represents the set of nodes, X^c is the feature matrix, and A^c is the adjacency matrix. The set of computational graphs is denoted $\mathcal{G}^c = (G_i^c)_{i=1}^N$ with $|\mathcal{G}^c| = N$. The paper operates within a classification framework and thus defines the target as $\mathcal{Y} = \{y_i\}_{i=1}^N$ with $y_i \in \mathbb{N}$, and consequently, the dataset $\mathcal{D} = \{(G_i^c, y_i)\}_{i=1}^N$ for graph classification, and $\mathcal{D} = \{(v_i^c, y_i)\}_{i=1}^{|V^c|}$ for node classification. Moreover, pretrained GNNs are defined in a general way by the function $f : \mathcal{G}^c$ or $V^c \to \tilde{\mathcal{Y}}$, aimed first at minimizing a loss function $\mathcal{L} : \mathcal{Y} \times \tilde{\mathcal{Y}} \to \mathbb{R}$, where $\tilde{\mathcal{Y}}$ represents the set of predictions, often assessed by cross-entropy in the context of GNNs.

2.2. Contexte

The goal of *Gem* is to provide an explanation for a graph, i.e. to find the most significant subgraph for prediction[1]. Mathematically, GNNs have yet learned $f(G^c) = \mathbb{P}(Y|G^c)$, and the aim is to find $G^s \subset G^c$ such that $\mathbb{P}(Y|G^s) = \mathbb{P}(Y|G^c)$. The paper denotes $f_{\text{exp}} : G^c \mapsto G^s$ the meta-model that explains the GNNs, commonly referred to as the *target GNN* or the `explainer`. *Gem* is inspired by *PGExplainer* [3], which also constructs a meta-model on top of that of the GNNs. However, the latter is composed of a multilayer perceptron (MLP) and requires a different structure that necessitates prior knowledge

[1]The assumption that the explainability of a graph is localized within the graph is inspired by neuroscience [4].

of the internal structure and parameters of the target GNN. Contrary to this, *Gem* is independent of any knowledge of the target GNN which is referred to as *model-agnostic* and is a significant advance.

2.3. Causal Contribution

The first step of *Gem* is to create a dataset that contains only the nodes and edges significant for prediction. To achieve this, the paper defines causal contribution by drawing inspiration from *Granger causality* [5], which quantifies causal relationships among multiple variables. The paper extends this notion to graphs[2], stating that G^s Granger-causes G^c if the prediction of $\tilde{y}$ is improved by utilizing information with G^s compared to without.

Definition 2.3.1. *Assume δ_G the loss of a graph G, i.e. $\mathcal{L}(y, f(G))$. The causal contribution of the edge e_j is defined as :*

$$\Delta_{\delta,e_j} = \delta_{G^c \setminus \{e_j\}} - \delta_{G^c}$$

with $G^c \setminus \{e_j\}$ to denote the graph G^c with out the edge e_j.

This concept enables the creation of the truncated dataset, which the paper refers to as the *distillation process.*

Definition 2.3.2. *The distillation process consists of finding the top-K causal contribution edges that create $\tilde{G}^s = \left(\tilde{V}^s, \tilde{A}^s, \widetilde{X}^s\right)$. Mathematically,*

$$\tilde{G}^s = G^c \setminus \arg(|E| - K) \max_{e_j \in E} \Delta_{\delta,e_j}$$

In a more formal manner, the algorithm is presented in Appendix 1. It consists of three for loops: the first one orders the edges by causal contribution, the second one considers constraints that can be imposed on the dataset (for example, here ensuring that the subgraph is not in two distinct parts), and the third one truncates the original graph. It should be noted that the edges may not be independent of each other. This paper leaves this issue open by suggesting that the distillation process should be modified with new graph rules depending on the dataset.

2.4. The Explainer

The second step of *Gem* involves to train an additional model, called the `explainer`, to predict $\tilde{G}^s$ from the original dataset. The particularity of this step is that any supervised learning model capable of outputting graphs can serve as an explainer. Once trained, it can be used to construct explanations using the generative mapping for the *target GNN* with little time and no need to retrain or adapt the predictive model. This flexibility to choose the predictive model is commonly referred to as *model-agnosticism* [6]. It is worth noting that in situations where ground-truth labels of the instances are readily available, the distillation process itself can function as an explainer. The paper restricts itself to the use of Graph Generative Models (GGMs) as explainers. Specifically, the model consists of a graph convolution network (GCNs) with an inner product decoder [7]. In essence, it applies various layers of graph convolutions to aggregate neighbor

[2]In principle, the notion of Granger causality would hold if $\mathbb{P}(Y|G^c) = \mathbb{P}(Y|G^s)$ holds.

information and learn node features. Then, it utilizes the inner product decoder to generate a matrix $\hat{A}^c$ as an explanation mask, which leads in constructing the explanation. Mathematically,

$$Z = \text{GCNs}(A^c, X^c) \text{ and } \hat{A}^c = \sigma(ZZ^T)$$

In particular, the paper uses the root mean square error between the reconstructed weighted matrices and the true causal contributions distilled based on the distillation process to supervise the learning process. Specifically, the model parameter complexity of *Gem* is independent of the input graph size as it naturally inherits the advantages of GCNs. With this setup the computational complexity is $\mathcal{O}(|E|)$, where $|E|$ is the number of edges of the instance to be explained. In the specific case of node classification, each node is labeled with the minimal number of edges connecting it to the node to be predicted. The intuition underlying this node labeling technique is that nodes with different relative positions to the target node may have different structural importance to its prediction label $\tilde{y}$. By incorporating the relative role features, *Gem* would capture which node to explain in a computation graph.

2.5. Paper Results

	MUTAG				NCI1			
K	15	20	25	30	15	20	25	30
Gem–0[3]	**64.0**	**78.1**	**81.0**	**85.0**	–	–	–	–
GNNExplainer-0	60.0	67.6	68.9	75.8	–	–	–	–
PGExplainer-0	22.5	38.5	57.6	72.3	–	–	–	–
Gem	66.3	**78.0**	**82.1**	**83.4**	56.9	**65.3**	68.9	**72.8**
GNNExplainer	**67.1**	74.9	75.8	80.9	**59.3**	61.8	**69.6**	72.0

Table 1. Explanation Accuracy on Real-World Datasets (%)

	BA-SHAPES					TREE-CYCLES				
K	5	6	7	8	9	6	7	8	9	10
Gem	**93.4**	**97.1**	**97.1**	**97.1**	**99.3**	86.1	**87.5**	**92.5**	**93.9**	95.4
GNNExplainer	82.4	88.2	91.2	91.2	94.1	14.3	46.8	74.6	91.4	**96.1**
PGExplainer	71.9	90.7	92.0	93.3	94.1	**94.4**	80.6	77.0	82.4	89.4

Table 2. Explanation Accuracy on Synthetic Datasets (%)

The results of the paper are quite unanimous. Using synthetic datasets, like Ba-shapes and Tree-Cycles, or real-world datasets, like Mutag [8] or NCI1 [9], *Gem* has a better accuracy of 30% on average in predicting the most explanatory subgraph than the two other state-of-the-art methods, *PGExplainer* [3] and *GNNExplainer* [10]. Moreover, it is empirically demonstrated in the appendices of the paper that Gem is on average 10 times faster than *PGExplainer* and 500 times faster than *GNNExplainer*[4].

3. Implementation

A source code is attached to this paper[5]. However, two significant flaws are apparent. Firstly, it is not clear and easily readable, and secondly, it does not correspond, unless I am mistaken, precisely to the configuration outlined in the article. That is why we will implement this code[6] as indicated in the paper and not in the code itself. Due to time constraints, I will only apply *Gem* to the Mutag [8] dataset with binary classifica-

[3]–0 indicate the explaination process is similar to PGExplainer.

[4]I will not develop this part of the studies. Check the original paper for more information.

[5]`https://github.com/wanyu-lin/ICML2021-Gem`

[6]`https://github.com/Biechy/Gem`

tion. For the pre-trained GNN, the implementation consists of three graph convolution networks (GCN) with an output dimension of 20 each, followed by a ReLU activation function. This is followed by concatenation of the three outputs, a global max pooling, and finally an MLP. It is trained for 1000 epochs using the Adam optimizer with a learning rate of 0.001 and a cross-entropy loss criterion. The test accuracy averages 78%, compared to 88% reported in the paper. As for the explainer, the implementation is a Variational Graph Autoencoder (VGAE) trained for 100 epochs using Adam optimizer with a learning rate of 0.01 and a criterion specific to variational autoencoders. Unlike the paper, which uses a simple mean square error, which is debatable in the context of VGAEs, a different criterion is employed here. The encoder consists of three layers of GCN with output dimensions 32, 32, and 12, followed by a unique reparametrization for VGAEs. The decoder is a sigmoid output applied to ZZ^T. Below are my results:

	MUTAG			
K	15	20	25	30
Gem	59.3	61.2	61.8	61.7

Table 3. Average Accuracy on Mutag with my implementation (%)

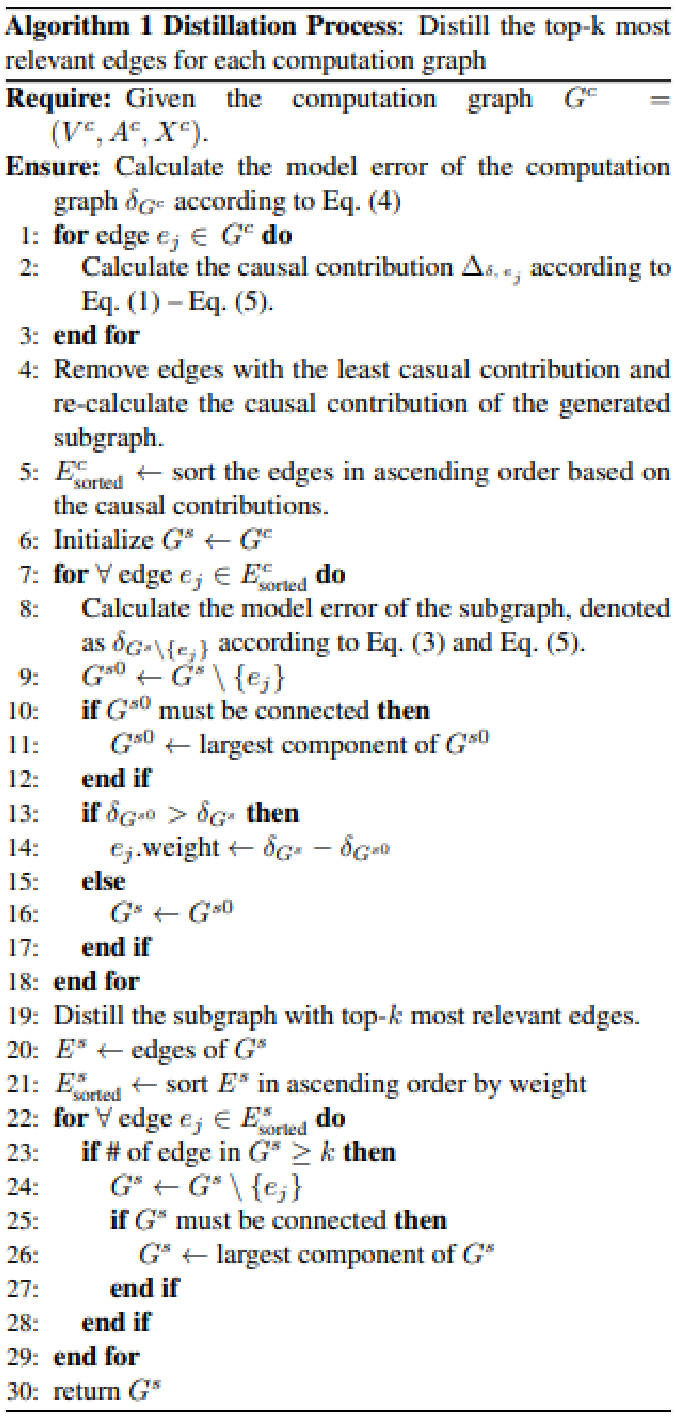
My results are rather disappointing because, compared to the paper, there is not a significant improvement with K. Furthermore, the difference between the model's accuracy and that of *Gem* is around 19% here, compared to 10% in the paper, which is once again disappointing. Given the shared code, I therefore question the accuracy of the paper's results.

4. Discussion

In sum, the paper devised a new framework *Gem* for explaining graph neural networks that use the first principles of Granger causality. *Gem* has several advantages over existing work: it is model-agnostic, compatible with any graph neural network models without any prior assumptions on the graph learning tasks, can generate compact subgraphs, causing theoutputs of the pre-trained GNNs very quickly after training. The paper is limited to variational graph autoencoders, but leaves open the possibility of using other models.

From a more personal perspective, I found the paper to be very well-written and it addresses the issue of model explainability quite naturally. I also found the implementation section to be very educational as it allowed me to code a graphical neural network, the distillation algorithm, and a graph autoencoder, even though the results leave scope for improvement.

A. Distillation Process Algorithm



Algorithm 1 Distillation Process: Distill the top-k most relevant edges for each computation graph

```
Require: Given the computation graph $G^c = (V^c, A^c, X^c)$.
Ensure: Calculate the model error of the computation graph $\delta_{G^c}$ according to Eq. (4)
 1: for edge $e_j \in G^c$ do
 2:     Calculate the causal contribution $\Delta_{\delta, e_j}$ according to Eq. (1) – Eq. (5).
 3: end for
 4: Remove edges with the least casual contribution and re-calculate the causal contribution of the generated subgraph.
 5: $E^c_{\text{sorted}} \leftarrow$ sort the edges in ascending order based on the causal contributions.
 6: Initialize $G^s \leftarrow G^c$
 7: for $\forall$ edge $e_j \in E^c_{\text{sorted}}$ do
 8:     Calculate the model error of the subgraph, denoted as $\delta_{G^s \setminus \{e_j\}}$ according to Eq. (3) and Eq. (5).
 9:     $G^{s0} \leftarrow G^s \setminus \{e_j\}$
10:     if $G^{s0}$ must be connected then
11:         $G^{s0} \leftarrow$ largest component of $G^{s0}$
12:     end if
13:     if $\delta_{G^{s0}} > \delta_{G^s}$ then
14:         $e_j$.weight $\leftarrow \delta_{G^s} - \delta_{G^{s0}}$
15:     else
16:         $G^s \leftarrow G^{s0}$
17:     end if
18: end for
19: Distill the subgraph with top-$k$ most relevant edges.
20: $E^s \leftarrow$ edges of $G^s$
21: $E^s_{\text{sorted}} \leftarrow$ sort $E^s$ in ascending order by weight
22: for $\forall$ edge $e_j \in E^s_{\text{sorted}}$ do
23:     if # of edge in $G^s \geq k$ then
24:         $G^s \leftarrow G^s \setminus \{e_j\}$
25:         if $G^s$ must be connected then
26:             $G^s \leftarrow$ largest component of $G^s$
27:         end if
28:     end if
29: end for
30: return $G^s$
```

1. Distillation Process Algorithm

References

1. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems. 30, (2017)
2. Lin, W., Lan, H., Li, B.: Generative causal explanations for graph neural networks. In: International Conference on Machine Learning. pp. 6666–6679 (2021)
3. Vu, M., Thai, M. T.: Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. Advances in neural information processing systems. 33, 12225–12235 (2020)
4. Biswal, B. B., Kylen, J. V., Hyde, J. S.: Simultaneous assessment of flow and BOLD signals in resting-state functional connectivity maps. NMR in Biomedicine. 10, 165–170 (1997)
5. Granger, C. W.: Testing for causality: A personal viewpoint. Journal of Economic Dynamics and control. 2, 329–352 (1980)
6. Ribeiro, M. T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. arXiv preprint arXiv:1606.05386. (2016)
7. Kipf, T. N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308. (2016)
8. Debnath, A. K., Compadre, R. L. Lopez de, Debnath, G., Shusterman, A. J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. Journal of medicinal chemistry. 34, 786–797 (1991)
9. Wale, N., Watson, I. A., Karypis, G.: Comparison of descriptor spaces for chemical compound retrieval and classification. Knowledge and Information Systems. 14, 347–375 (2008)
10. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems. 32, (2019)