Parameterized Explainer for Graph Neural Network

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen,Xiang Zhang

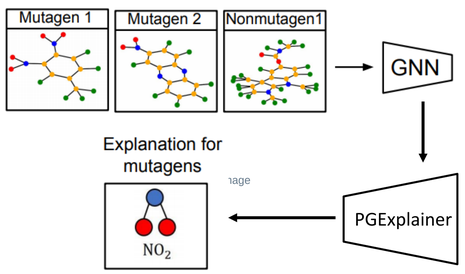
Introduction

Despite recent progress in Graph Neural Networks (GNNs), explaining predictions made by GNNs remains a challenging open problem. The leading method independently addresses the local explanations. In this study, we propose PGExplainer, a parameterized explainer for GNNs. PGExplainer adopts a deep neural network to parameterize the generation process of explanations, which enables PGExplainer a natural approach to explaining multiple instances collectively.

- Local Fidelity
- Model Agnostic
- Task Agnostic
- Global View
- Collective
- Inductive

Example

PGExplainer provides human-understandable explanations for predictions made by GNNs. The left part shows the process of applying GNNs for graph classification on the MUTAG dataset. A GNN based model is trained to predict their mutagenic effects. As a post-hoc method, PGExplainer takes the trained GNN model as input and provides consistent explanations for predictions made by the GNN model. For the mutagen molecule graphs in the example, the explanation is the NO2 group.



The PGExplainer

The learning objective

To explain predictions made by a GNN model, we divide the original input graph Go into two subgraphs: $G_o = G_s + \Delta G$, where G_s presents the underlying subgraph that makes important contributions to GNN's predictions, which is the expected explanatory graph, and ΔG consists of the remaining task-irrelevant edges for predictions made by the GNN. PGExplainer finds G_s by maximizing the mutual information between the GNN's predictions and the underlying structure G_s

$$\max_{G_s} \mathbf{MI}(Y_o, G_s) = H(Y_o) - H(Y_o|G = G_s)$$

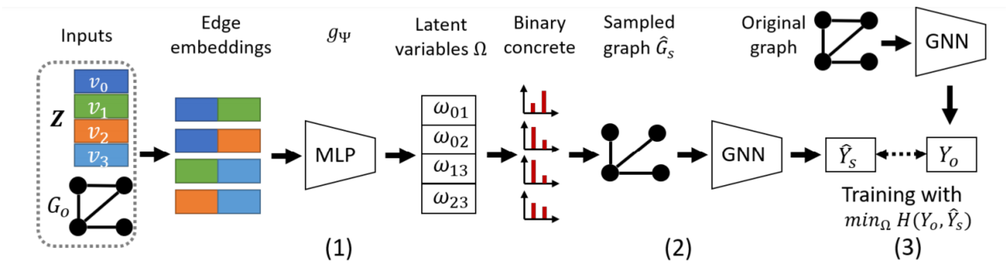
The reparameterization trick

Due to the discrete nature of G_s, we relax edge weights from binary variables to continuous variables in the range (0, 1) and adopt the reparameterization trick to efficiently optimize the objective function with gradient-based methods. We rewrite the objective function as:

$$\min_{\Omega} -\frac{1}{K}\sum_{k=1}^{K}\sum_{c=1}^{C} P_\Phi(Y = c|G = G_o) \log P_\Phi(Y = c|G = \hat{G}_s^{(k)})$$

Explanation of graph neural networks with a global view

To have a global view of a GNN model, our method collectively explains predictions made by a trained model on multiple instances. Instead of treating Ω in the above equation as independent variables, we utilize a parameterized network to learn to generate explanations from the trained GNN model, which also applies to unexplained instances.



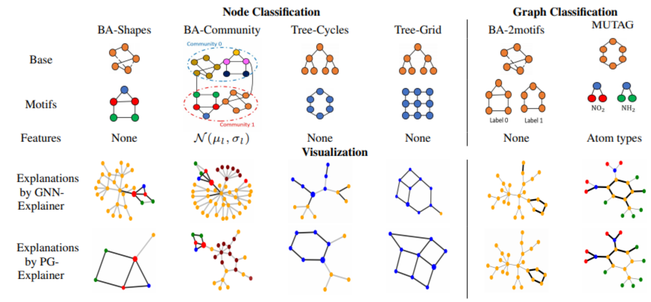
Experimental study

We follow the setting in GNNExplainer and construct four kinds of node classification datasets, BA-Shapes, BA-Community, Tree-Cycles, and Tree-Grids [1]. Furthermore, we also construct BA-2motifs and use a real-life dataset MUTAG for graph classification.

Table 1: Dataset statistics

	Node Classification				Graph Classification	
	BA-Shapes	BA-Community	Tree-Cycles	Tree-Grid	BA-2motifs	MUTAG
#graphs	1	1	1	1	1,000	4,337
#nodes	700	1,400	871	1,231	25,000	131,488
#edges	4,110	8,920	1,950	3,410	51,392	266,894
#labels	4	8	2	2	2	2

We compare with the baseline methods, GNNExplainer [1], a gradient-based method (GRAD) [1], graph attention network (ATT) [2], and Gradient [3].



Explanation AUC

GRAD	0.882	0.750	0.905	0.612	0.717	0.783
ATT	0.815	0.739	0.824	0.667	0.674	0.765
Gradient	-	-	-	-	0.773	0.653
GNNExplainer	0.925	0.836	0.948	0.875	0.742	0.727
PGExplainer	**0.963**±0.011	**0.945**±0.019	**0.987**±0.007	**0.907**±0.014	**0.926**±0.021	**0.873**±0.013
Improve	4.1%	13.0%	4.1%	3.7%	24.7%	11.5%

Inference Time (ms)

GNNExplainer	650.60	696.61	690.13	713.40	934.72	409.98
PGExplainer	10.92	24.07	6.36	6.72	80.13	9.68
Speed-up	59x	29x	108x	106x	12x	42x

References

[1] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In NeurIPS, pages 9240–9251, 2019.

[2] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In: ICLR 2018.

[3] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In: CVPR 2019.