PGM-Explainer

1. Introduction

PGM-Explainer expands to Probabilistic Graphical Model model-agnostic explainer for GNNs, introduced by “PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks” paper in 2020. Explanation for a prediction is a Bayesian network approximating that prediction. Authors state that if a perfect map for the sampled data generated by perturbing the GNN`s input exists, explanation will always include the Markov-blanket of the target prediction, so all statistical information on the explained prediction will be preserved.

Bayesian network is a probabilistic graphical model for representing knowledge about an uncertain domain, where each node corresponds to a random variable, and each edge represents the conditional probability for the corresponding random variables.

A Markov blanket in simple words is a subset (in this example - a subgraph) that contains all the useful information. In a Bayesian network, the Markov blanket of a node A includes its parents, children, and the other parents of all of its children.

2. Simple explanation

Architecture of PGM-Explainer, meaning the process of creating an explanation by this method can be described by below figure (found in original paper):

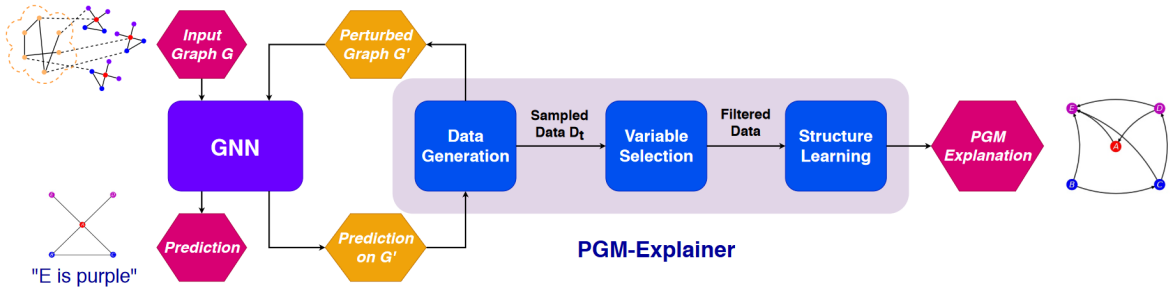

Fig 1: PGM-Explainer architecture.

It consists of three major steps:

1. Data generation: D_t consists of subgraphs with perturbed features of the original graph, and model’s prediction for that subgraph.
2. Variable selection: at this point D_t can contain perturbations of thousands of features, and finding optimal PGM might be very expensive. We need to trim the set of variables before finding an explanation. This is done by using a Markov blanket, we find a small subset that is guaranteed to contain a Markov blanket..
3. Structure learning: the final step is to learn the explanation Bayesian network.

3. Formulation

To find the explanation in structure learning step we utilize BIC score:

$$R_{\Phi,t}(\mathcal{B}) = \text{score}_{BIC}(\mathcal{B} : \mathcal{D}_t[\boldsymbol{U}(t)]) = l(\hat{\theta}_{\mathcal{B}} : \mathcal{D}_t[\boldsymbol{U}(t)]) - \frac{\log n}{2}\text{Dim}[\mathcal{B}]$$

where $D_t[U(t)]$ is data D_t on variables $U(t)$ and $Dim[B]$ is the dimension of model B. θ_B are the parameters of B and function $l(\theta_B: D_t[U(t)])$ is the log-likelihood between $D_t[U(t)]$ and θ_B. Given this objective, PGM-Explainer can use exhaustive-search to solve for an optimal PGM.

4. Example

In pytorch_geometric, this explainer is still in contrib module, meaning that it is still in progress. Unfortunately, it also means that it is not implemented yet completely, and for now can only give explanations of influential features.
Please note that this explainer doesn't calculate individual score for each feature, it returns either 1 or 0, meaning that this feature was or was not influential.

In the original paper, we can find examples of explanations on the MNIST SuperPixel-Graph dataset. Here, we get the most influential nodes for that prediction obtained by explainers (PGM, SHAP and GRAD).

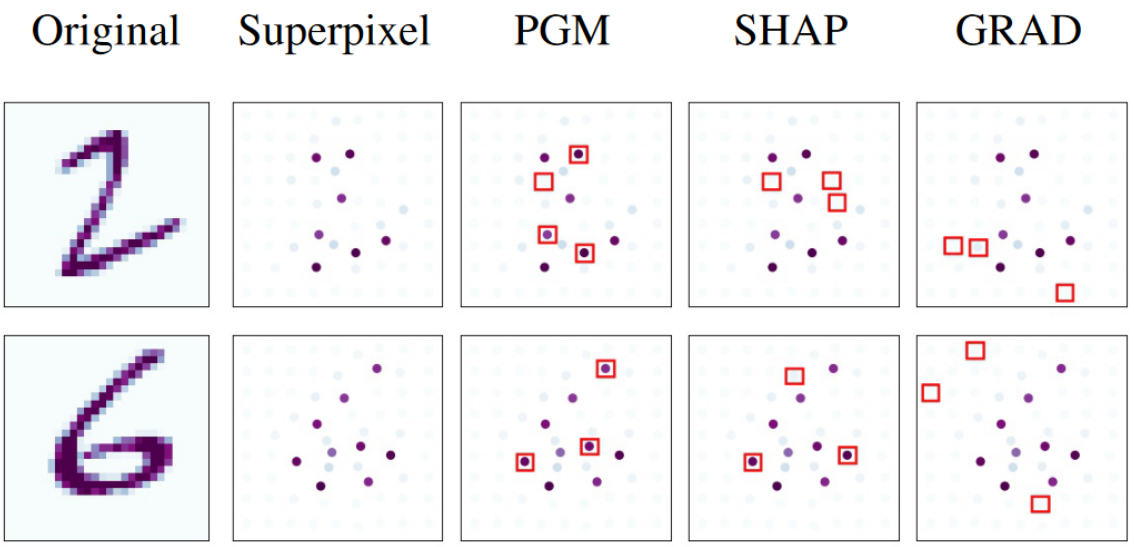

Fig 2: PGM, SHAP and GRAD explanations.

As we can observe in the above examples, influential nodes found by PGM are probably closer to what the human eye would classify as influential, which we cannot say about nodes returned by SHAP or GRAD.

5. References

- Minh N. Vu, My T. Thai: PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks, 2020
- PGMExplainer github repository
- Pytorch geometric - PGMExplainer
- P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad: Collective Classification in Network Data, 2008