Explainability in Graph Neural Networks: Recent Advances

Presenter: Ziyuan Ye

2022-6-10

Yuan, H., Yu, H., Gui, S., & Ji, S. (2020). Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445*.

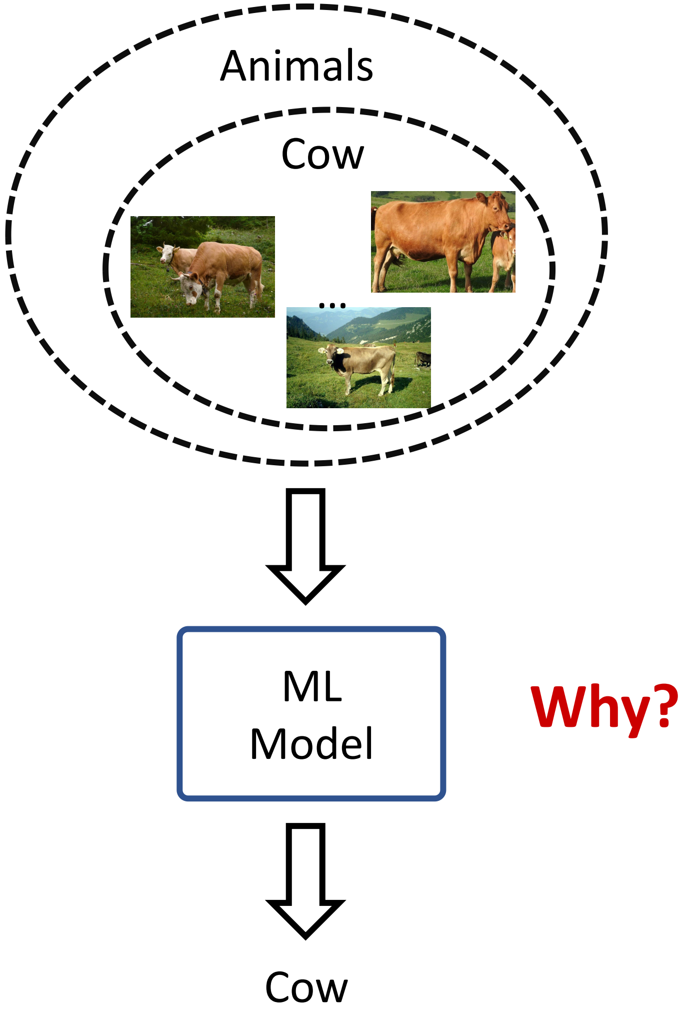
Overview

1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
    - Gradients/Features
    - Perturbations
    - Surrogate
    - Decomposition
4. Model-level Explanations
    - Generation
5. Looking forward

Overview

1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
   - Gradients/Features
   - Perturbations
   - Surrogate
   - Decomposition
4. Model-level Explanations
   - Generation
5. Looking forward

A brief intro: XAI in graph



Animals

Cow

…

ML Model

Why?

Cow

- Deep graph models becoming more widespread
- Black-box models are the mainstream
  - GCN
  - GAT
  - GIN
  - …
- Various concerns about model transparency

“Interpretable” v.s. “Explainable”

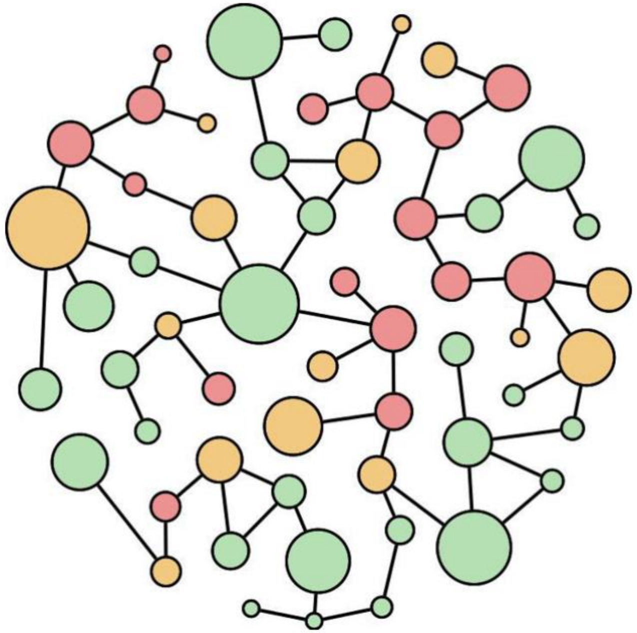
- **Interpretable:** we consider a model to be “interpretable” if the **model itself can provide humanly understandable interpretations of its predictions**. Note that such a model is **no longer a black box** to some extent. For example: decision tree

- **Explainable:** an “explainable” model implies that **the model is still a black box** whose **predictions could potentially be understood by post hoc explanation techniques**.

Overview

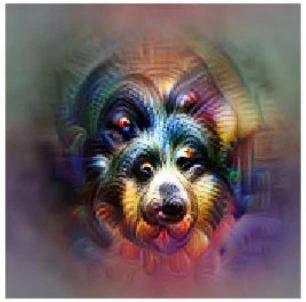
1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
    - Gradients/Features
    - Perturbations
    - Surrogate
    - Decomposition
4. Model-level Explanations
    - Generation
5. Looking forward

The challenges of XAI in graph



The challenges cause by characteristics of graph:

- Graphs are not grid-like data
  - Each node has different numbers of neighbors
  - There is no locality information
- Graph contain important topology information
  - Feature matrices
  - Adjacency matrices
- Graph data is less intuitive than images and texts
  - For explanations of images and texts, humans can easily understand them even though the explanations are highly abstract.
  - Above can't be held for graph data.



Graph classification:

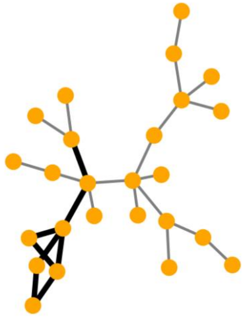
➢ Graph structures, node features

Node classification:

➢ Message passing, graph structures, node features

The challenges of transferring current methods:

- It can't be optimize via input optimization method to obtain abstract graph structure for explaining.
- Applying soft masks to the adjacency matrices will destroy the discretenss property.



Overview

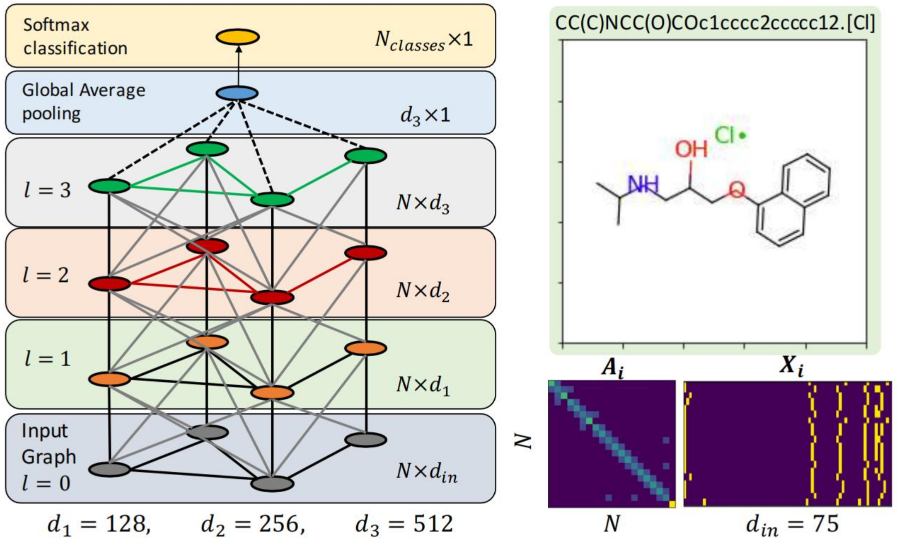
1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
   - Gradients/Features
   - Perturbations
   - Surrogate
   - Decomposition
4. Model-level Explanations
   - Generation
5. Looking forward

Gradient / Features-based methods

- **Sensitivity Analysis (SA)**

Assume x is the input, f is the graph model, S is the saliency map, G is the explanation method

$$S(x) = ||\nabla_x f||^2$$



- **Guided Backpropagation (GBP)**

Slightly, different from SA, negative gradients are clipped during backpropagation, which concentrates the explanation on the features that have an excitatory effect on the output.

Limitations:

- SA and GBP can only reflect the sensitivity between input and output, which cannot accurately show the importance.
- In addition, it also suffers from saturation problems.

F. Baldassarre and H. Azizpour, “Explainability techniques for graph convolutional networks,” in *International Conference on Machine Learning (ICML) Workshops, 2019 Workshop on Learning and Reasoning with Graph-Structured Representations*, 2019.

Gradient / Features-based methods

GCN model

$$F^l(X,A) = \sigma(\underbrace{\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}}_{V} F^{(l-1)}(X,A)W^l)$$

$$W^l \in \mathbb{R}^{d_l \times d_{l+1}}$$

Let the k'th graph convolutional feature map at layer l be defined as:

$$F_k^l(X,A) = \sigma(VF^{(l-1)}(X,A)W_k^l)$$

The graph average pooling feature after the final convolutional layer, L, is calculated as:

$$e_k = \frac{1}{N}\sum_{n=1}^{N} F_{k,n}^L(X,A)$$

The class score is calculated as:

$$y^c = \sum_k w_k^c e_k$$

- **Gradient-based heatmaps**

$$L^c_{Gradient}[n] = \|\mathrm{ReLU}\left(\frac{\partial y^c}{\partial X_n}\right)\|$$

- **Class Activation Mapping (CAM)**

$$L^c_{CAM}[n] = \mathrm{ReLU}(\sum_k w_k^c F^L_{k,n}(X,A)))$$

- **Gradient-weighted Class Activation Mapping (Grad-CAM)**

Class specific weights for class c at layer l and for feature k:

$$\alpha_k^{l,c} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial y^c}{\partial F^l_{k,n}}$$

Heatmap calculated from layer l:

$$L^c_{Grad-CAM}[l,n] = \mathrm{ReLU}(\sum_k \alpha_k^{l,c} F^l_{k,n}(X,A))$$

P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 772–10 781.

Gradient / Features-based methods

- **Gradient-based heatmaps**

$$L^c_{Gradient}[n] = \|\text{ReLU}\left(\frac{\partial y^c}{\partial X_n}\right)\|$$

- **Class Activation Mapping (CAM)**

$$L^c_{CAM}[n] = \text{ReLU}(\sum_k w^c_k F^L_{k,n}(X, A)))$$

- **Gradient-weighted Class Activation Mapping (Grad-CAM)**

  Class specific weights for class c at layer l and for feature k:

$$\alpha^{l,c}_k = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial y^c}{\partial F^l_{k,n}}$$

  Heatmap calculated from layer l:

$$L^c_{Grad-CAM}[l, n] = \text{ReLU}(\sum_k \alpha^{l,c}_k F^l_{k,n}(X, A))$$

Limitations:

- ➢ CAM has special requirements for the GNN structure, which limits its application and generalization.
- ➢ It assumes that the final node embeddings can reflect the input importance, which is heuristic and may not be true.
- ➢ It can only explain graph classification models and cannot be applied to node classification tasks.

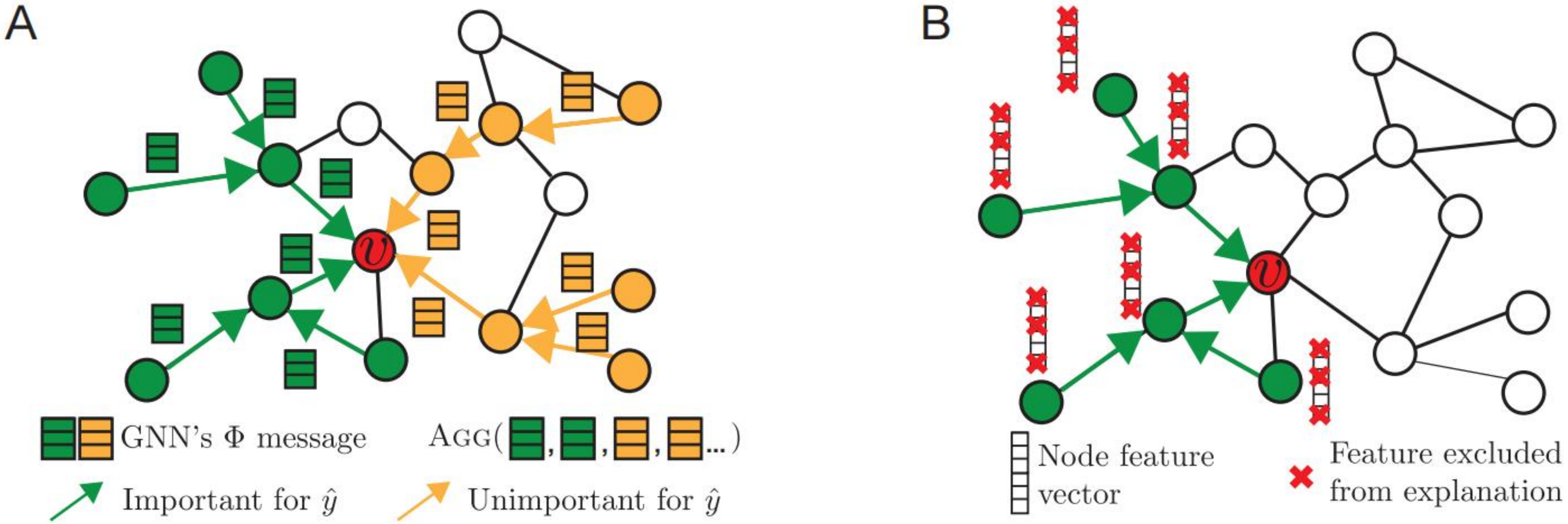
P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 772–10 781.

Overview

1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
    - Gradients/Features
    - Perturbations
    - Surrogate
    - Decomposition
4. Model-level Explanations
    - Generation
5. Looking forward

Perturbation-based methods

- **GNNExplainer**



$$\max_{G_S} MI\left(Y,(G_S,X_S)\right) = H(Y) - H(Y|G=G_S, X=X_S)$$

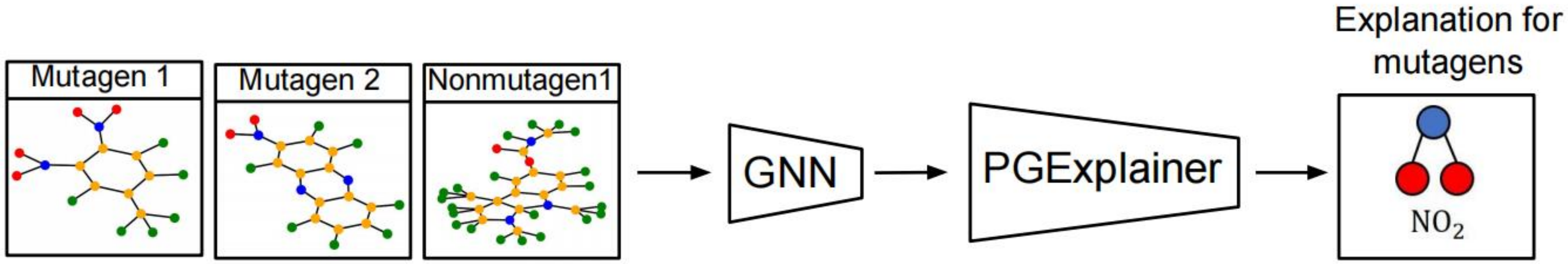
Limitations:

- “Introduce evidence” problem
- The explanations may lack a global view

Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, “Gnnexplainer: Generating explanations for graph neural networks,” *in Advances in neural information processing systems*, 2019, pp. 9244– 9255.

Perturbation-based methods

- **PGExplainer**



Key idea:

It trains a parameterized mask predictor to predict edge masks.

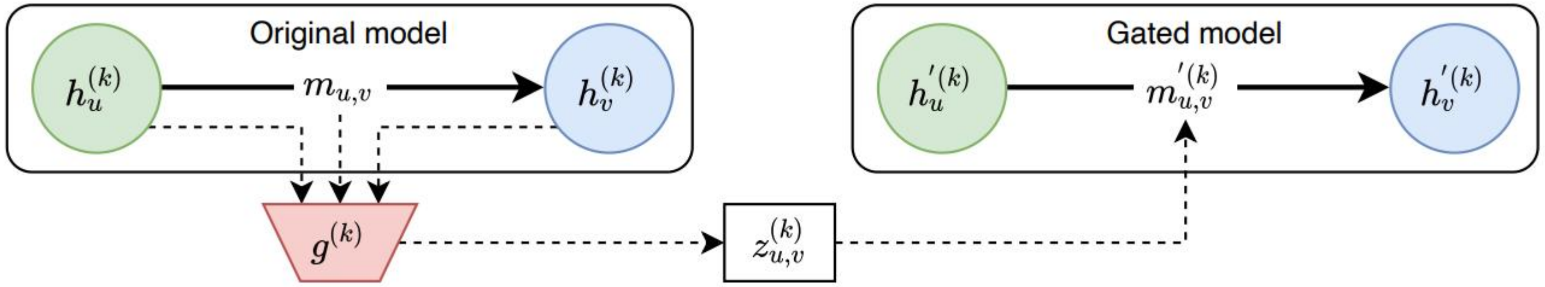
Mechanism of PGExplainer:

1. Given an input graph, it first obtains the embeddings for each edge by concatenating node embeddings.
2. Then the predictor uses the edge embeddings to predict the probability of each edge being selected, which can be treated as the importance score.
3. Next, the approximated discrete masks are sampled via the reparameterization trick.
4. Finally, the mask predictor is trained by maximizing the mutual information between the original predictions and new predictions.

D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, “Parameterized explainer for graph neural network,” in *Advances in neural information processing systems*, 2020.

Perturbation-based methods

- **GraphMask**



Compared to PGExplainer:
Similarity:
It trains a classifier to predict whether an edge can be dropped without affecting the original predictions.
Difference:

1. GraphMask obtains an edge mask for each GNN layer while PGExplainer only focuses the input space.
2. To avoiding changing graph structures, the dropped edges are replaced by learnable baseline connections, which are vectors with the same dimensions as node embeddings.

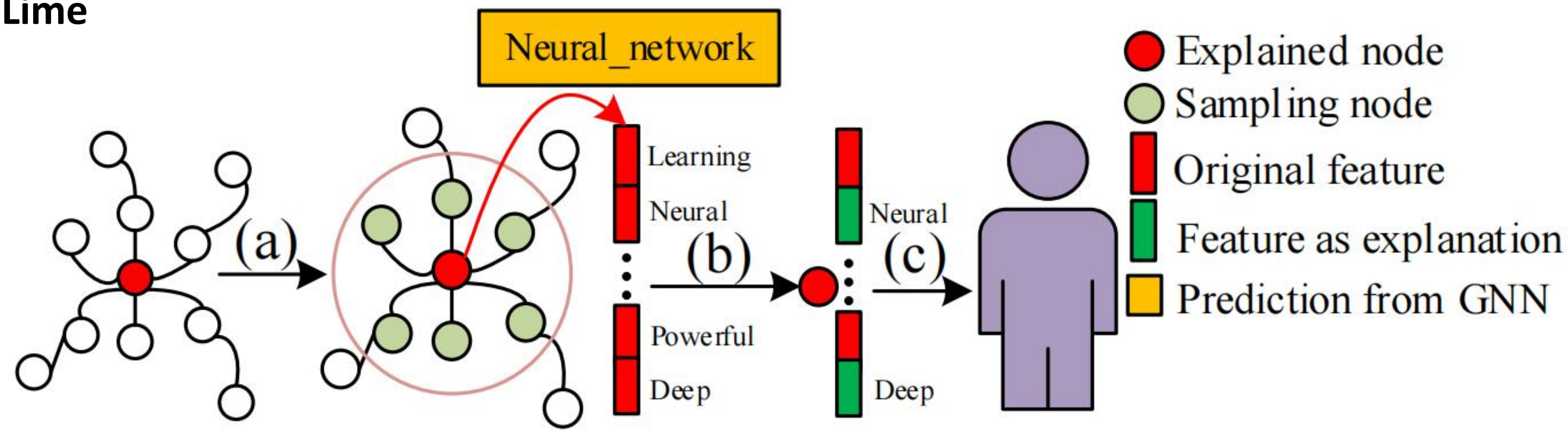
Schlichtkrull, M. S., De Cao, N., & Titov, I. (2021). Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking. In *International Conference on Learning Representations*.

Overview

1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
    - Gradients/Features
    - Perturbations
    - Surrogate
    - Decomposition
4. Model-level Explanations
    - Generation
5. Looking forward

Surrogate-based methods

- **GraphLime**



GraphLime considers its *N*-hop neighboring nodes (Determined by the trained GNNs) and their predictions as its local dataset and borrow Hilbert-Schmidt Independence Criterion (HSIC) Lasso for predictions.

$$HSIC(X,Y) = MMD(P_{XY}, P_X P_Y)$$

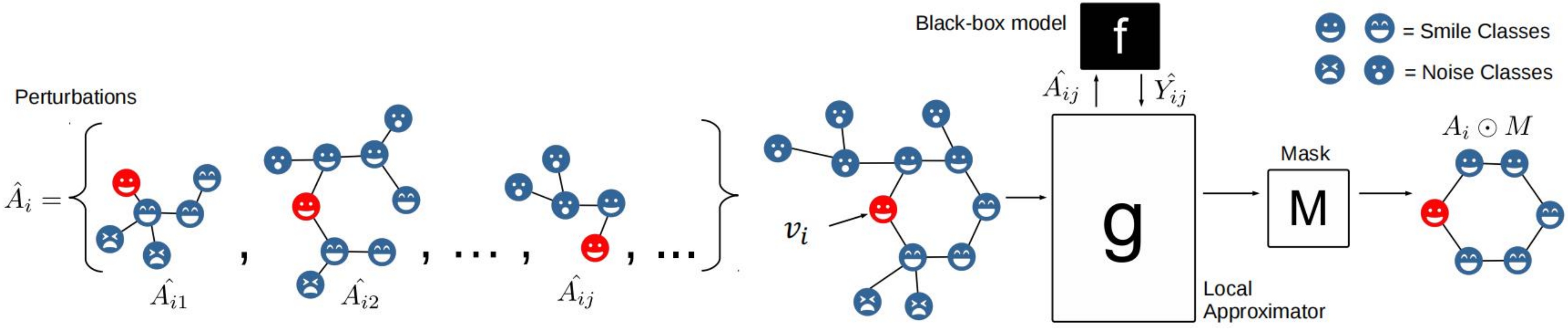
Finally, based on the weights of different features in HSIC Lasso, it can select important features to explain the HSIC Lasso predictions.

Limitation: 1. GraphLime only provide explanations for node features, ignore graph structures. 2. GraphLime is proposed to explain node classification but cannot be applied to graph classification models.

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., & Chang, Y. (2020). Graphlime: Local interpretable model explanations for graph neural networks. arXiv preprint arXiv:2001.06216.

Surrogate-based methods

- **RelEx**



1. Obtains a local dataset by randomly sampling connected subgraphs from the computational graph (BFS manner).
2. Feeding these subgraphs to train a GNNs to approximate the target node.
3. Apply perturbation method to get a mask to define the final interpretations.

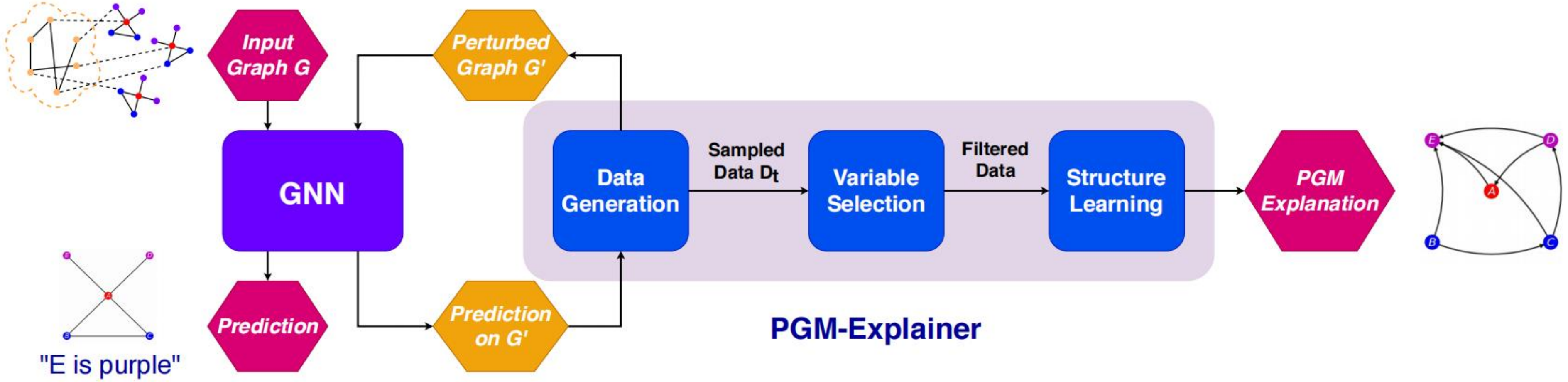
Limitations:

1. It contains multiple approximation, making the explanations less convincing and trustable.
2. It is not necessary to build another non-interpretable deep model as the surrogate model to explain.
3. It is also unknown how it can be applied for graph classification tasks.

Zhang, Y., Defazio, D., & Ramesh, A. (2021, July). Relex: A model-agnostic relational model explainer. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 1042-1049).

Surrogate-based methods

- **PGM-Explainer**



1. Given an input graph, each time PGM-Explainer randomly perturbs the node features of several random nodes within the computational graph.
2. Then for any node in the computational graph, PGM-Explainer records a random variable indicating whether its features are perturbed and its influence on the GNN predictions.
3. By repeating such procedures multiple times, a local dataset is obtained.
4. Then it selects top dependent variables to reduce the size of the local dataset via the Grow-Shrink (GS) algorithm.
5. Finally, an interpretable Bayesian network is employed to fit the local dataset and to explain the predictions of the original GNN model.

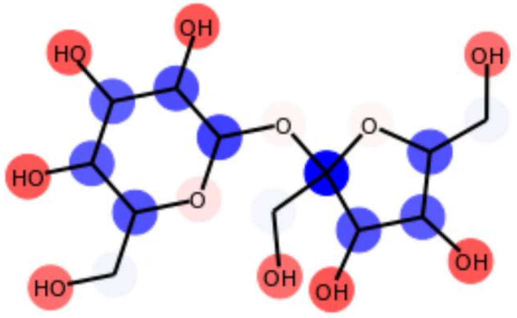
Vu, M., & Thai, M. T. (2020). Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. Advances in neural information processing systems, 33, 12225-12235.

Overview

1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
    - Gradients/Features
    - Perturbations
    - Surrogate
    - Decomposition
4. Model-level Explanations
    - Generation
5. Looking forward

Decomposition-based methods

- **LRP**



Red denotes **important** nodes
Blue denotes **unimportant** nodes

LRP decomposes the output prediction score to different node importance scores.

1. For a target neuron, its score is represented as a linear approximation of neuron scores from the previous layer.
2. Intuitively, the neuron with a higher contribution of the target neuron activation receives a larger fraction of the target neuron score.

Advantages:

LRP identifies which features of the input contribute the most to the final prediction. Furthermore, it is capable of handling positive and negative relevance, allowing for a deeper analysis of the contributing factors.

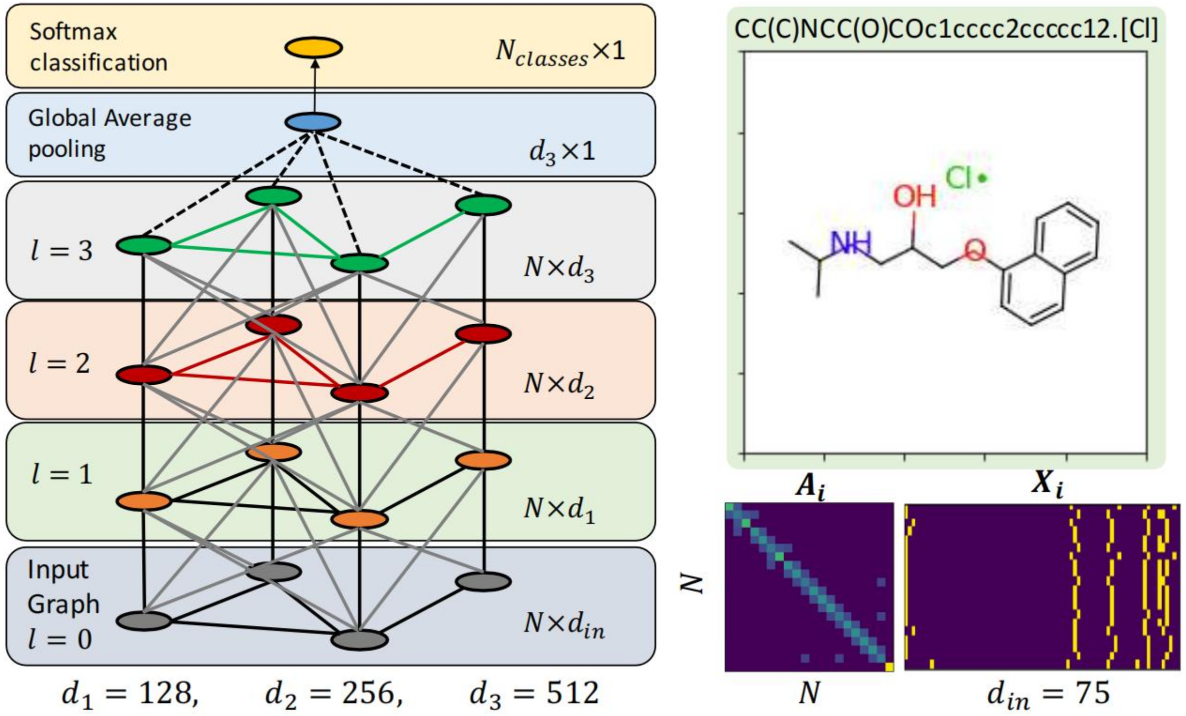
Limitations:

1. Ignore graph structure.
2. Such a algorithm requires a comprehensive understanding of the model structures.

F. Baldassarre and H. Azizpour, "Explainability techniques for graph convolutional networks," in International Conference on Machine Learning (ICML) Workshops, 2019 Workshop on Learning and Reasoning with Graph-Structured Representations, 2019.

Decomposition-based methods

- **Excitation BP**



Excitation BP shares a similar idea as the LRP algorithm but is developed based on the law of total probability.
Compared to LRP: It defines that the probability of a neuron in the current layer is equal to the total probabilities it outputs to all connected neurons in the next layer.
Share the same limitation as LRP.

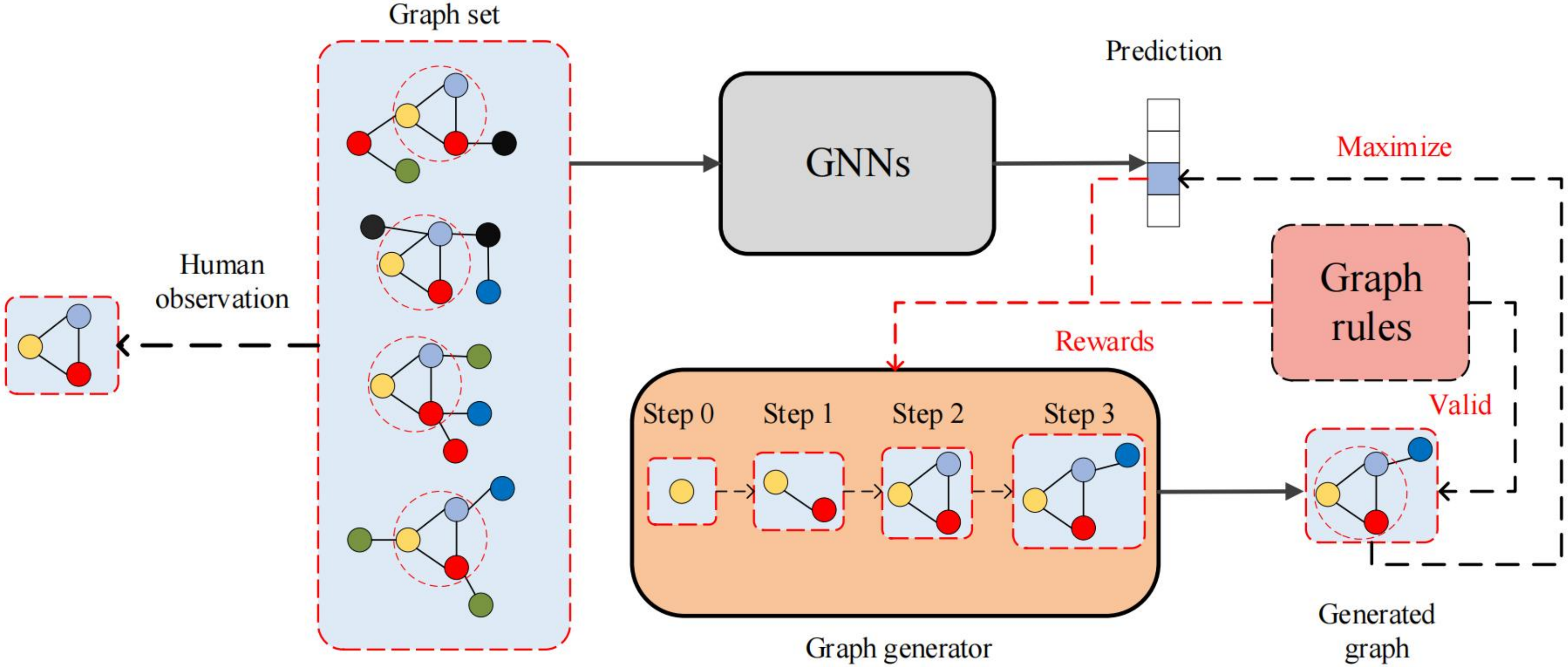
Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., & Hoffmann, H. (2019). Explainability methods for graph convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10772-10781).

Overview

1. A brief intro: XAI in graph
2. The challenges
3. Instance-level Explanations
   - Gradients/Features
   - Perturbations
   - Surrogate
   - Decomposition
4. Model-level Explanations
   - Generation
5. Looking forward

Generation-based methods

- **XGNN**



Instead of directly optimizing the input graph, it trains a graph generator so that the generated graphs can maximize a target graph prediction.

Limitation:

It is unknown whether XGNN can be applied to node classification tasks.

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., & Chang, Y. (2020). Graphlime: Local interpretable model explanations for graph neural networks. arXiv preprint arXiv:2001.06216.

Overview

1. A brief intro: XAI in graph

2. The challenges

3. Instance-level Explanations
   - Gradients/Features
   - Perturbations
   - Surrogate
   - Decomposition

4. Model-level Explanations
   - Generation

5. Looking forward

Looking forward

- Gradients/Features
- Perturbations
- Surrogate
- Decomposition
- Generation
- **What next…?**

Thanks for your attention!