Generating Explanations for Graph Neural Networks

Jinze Cui
Hanze Meng
Introduction to GNN
What is graph data

**Definition:** Data representing objects and relationships between them using nodes and edges of graphs.

**Possible sources** of graph data: social network, images, molecules, …

**Example:** Molecules modeled as graphs.
What we want to achieve using these graph data

Solve Prediction Problems!

Three levels of prediction based on graph structure:

- **Graph Level**: For a new molecule, predict whether it is toxic or not based on information from the graph data.

- **Node Level**: In a social network graph, predict the likelihood that a person will develop new friendship.

- **Edge Level**: In a map, predict the time it will take to the destination.
How do we predict

Neural network models are helpful!

A refresh on (convolutional) neural network:

- For classification and pattern recognition.
- Consists of multiple layers of neurons.
- Neurons process and transmit information.
How do we predict (continued)

Particularly on convolutional neural network:

Limitations:

- For fixed-structure data, ex. 2-dimension matrix of pixels.

- CNN does not guarantee invariance on node ordering.

Graph data are not structured well.

Need a new framework to process graph data.
How does GNN work

A graph-in, graph-out structure:

- Embed feature info into nodes, edges and global context of the input graph.
- Progressively transform the embedded information through multiple layers.
- No changes on the connectivity of the input graph.
What does GNN do at each layer

- Three key computations at each layer $l$: message, aggregation and update.

- **Message:** A function $MSG(h_{l-1}^{i}, h_{l-1}^{j}, r_{ij})$,
  $h_{l-1}^{i}$ and $h_{l-1}^{j}$ are representations of nodes $v_{i}$ and $v_{j}$ in layer $l-1$,
  $r_{ij}$ is the relation information between nodes $v_{i}$ and $v_{j}$.

- **Aggregation:** A function $AGG(\{m_{ij} \mid v_{j} \in N_{vi}\})$,
  $v_{i}$ is the node we are aggregating on,
  $N_{vi}$ is the neighborhood of $v_{i}$,
  $m_{ij}$ is the message from the $MSG$ function.
What does GNN do at each layer (continued)

- **Aggregation**: Some possible AGG functions are mean, max, and min.

- **Update**: A nonlinear function $UPDATE(M_{ij}^l, h_i^{l-1})$,
  - $M_{ij}^l$ is the aggregated message from the MSG function,
  - $h_i^{l-1}$ is the representation of node $v_i$ in the previous layer.
Review on Paper:

GNNExplainer: Generating Explanations for Graph Neural Networks
Key insights into the paper

- Transparency is important for the following reasons:

- Increases the trust in models themselves.

- Avoids fairness or privacy issues.

- Helps detect incorrect patterns before deployment.

Image Source:
Key insights into the paper (continued)

- Hard for GNN models to generate understandable explanations

- Approaches to explain other neural network models:
  - Probe surrogate models for local approximation,
  - Identify influential input instances,

Generally fail to capture relation information in graph data

Needs a way to generate explanations in GNN.
- Gives explanations for predictions made by any GNN models.

- Input: a trained GNN model, the input graph and its predictions.
- Output explanation: a subgraph of the input graph and a subset of the node features that influence the prediction the most.

- Handles both single and multi instances explanations, meaning that it can explain for either a single node or a class of nodes.
GNNEexplainer: Mathematical terminologies

Goal: Explain a node classification task

Notations: A graph $G$ on edges $E$ and nodes $V$,

An associated set of node features $X = \{x_1, x_2, \ldots, x_n\}, x_i \in \mathbb{R}^d$,

A set of classes $\{1, \ldots, C\}$ to be classified into,

A computation graph $G_c(v)$ for node $v$,

The associated adjacency matrix $A_c(v) \in \{0, 1\}^{n \times n}$,

The associated node feature set $X_c(v) = \{x_j \mid v_j \in G_c(v)\}$. 
- GNN is learning a conditional distribution $P_{\Phi}(Y \mid G_c, X_c)$, where $Y$ is a random variable representing the label in $\{1, \ldots, C\}$.

- $G_s$ is a subgraph of the computation graph $G_c$. 
- $X_s$ is the associated feature set with $G_s$, and further $X_s^F$ denotes a subset of $X_c$ used in the final explanation generated.

- GNNExplainer generates explanation for prediction $\hat{y}$ as $(G_s, X_s^F)$. 
Methodology:
Single-instance explanations
GNNEexplainer: Single-instance explanations

For a node $v$, we have:

$$\hat{y} = \Phi(G_c(v), X_c(v))$$

predicted class \hspace{3cm} trained GNN model \hspace{3cm} computation graph \hspace{3cm} node feature information

By GNNEexplainer, we want explain/identify:

$$G_S \subseteq G_c(v) \hspace{3cm} X_S = \{x_j | v_j \in G_S \}$$

important subgraph for the prediction \hspace{3cm} associated node features
GNNEExplainer: Single-instance explanations

How to define the importance? → Mutual information $MI$

What is mutual information?

- reduction in uncertainty about one random variable given knowledge of another

\[
MI(X, Y) = H(X) - H(X|Y)
\]

entropy - conditional entropy
**GNNEexplainer: Single-instance explanations**

**Entropy:** a measure of uncertainty on $X$

- The higher the entropy, the more uncertain

- Be maximal when $P_X(x)$ is uniform

- Defined as $H(X) = - \sum_x P_X(x) \log P_X(x)$

---

**Conditional Entropy:** $H(X|Y)$

- Average uncertainty about $X$ after observing $Y$

- Defined as $H(X|Y) = \sum_y P_Y(y) \left[ - \sum_x P_{X|Y}(x|y) \log \left( P_{X|Y}(x|y) \right) \right]$
Example on entropy: guessing the color

$$H(X) = - \sum_x P_X(x) \log P_X(x)$$

0.477 0.439
Example on MI: guessing the color with help

Choose a hint:
- # of
- # of and # of

\[ MI(X, Y) = H(X) - H(X|Y) \]
GNNE育人: 单实例解释

如何定义重要性？

\[
\max_{G_S} MI(Y, (G_S, X_S)) = H(Y) - H(Y | G = G_S, X = X_S)
\]

最大熵对整个图；常数熵对减小的图

这相当于最小化

\[
H(Y | G = G_S, X = X_S) = -\mathbb{E}_{Y | G_S, X_S} [\log P_{\Phi}(Y | G = G_S, X = X_S)]
\]

- \( G_S \) 最小化了 \( \Phi \) 的不确定性，当计算限制到 \( G_S \)
- 因此，\( G_S \) 最大化了概率的 \( \hat{y} \)
GNNEexplainer: Single-instance explanations

$G_c$ has exponentially many $G_S$ → How to solve the optimal $G_S$?

- **Step 1:** Approximate the distribution of $G_S$ as $\mathcal{G}$
  - Fractional adjacency matrix $A_S \in [0, 1]^{n \times n}$, and enforce $A_S[j, k] \leq A_c[j, k]$

- **Step 2:** To minimize $H(Y|G = G_S, X = X_S)$ now becomes
  $$\min_{\mathcal{G}} \mathbb{E}_{G_S \sim \mathcal{G}} H(Y|G = G_S, X = X_S)$$

- **Step 3:** By Jensen’s inequality with convexity assumption, consider the upper bound
  $$\min_{\mathcal{G}} H(Y|G = \mathbb{E}_{\mathcal{G}}[G_S], X = X_S)$$
GNNEExplainer: Single-instance explanations

$G_c$ has exponentially many $G_S$ \rightarrow How to solve the optimal $G_S$?

- **Step 4:** Approximate $G$ as $P_G(G_S) = \prod_{(j,k) \in G_c} A_S[j,k]$

- **Step 5:** With a regularizer for promoting discreteness, we replace $\mathbb{E}_G[G_S]$ by

$$A_c \odot \sigma(M)$$

original adjacency matrix sigmoid function we only need to learn
Experiments:
Single-instance explanations
Experiments: Datasets

Synthetic datasets:

- **BA-Shapes**
  - a base Barabási-Albert (BA) graph on 300 nodes
  - $\oplus$: connect randomly selected nodes to a set of 80 “house” shaped motifs
  - $\oplus$: connect randomly selected nodes to 0.1 $N$ random edges
  - 4 classes
Experiments: Datasets

Synthetic datasets:

- **BA-Community**
  - Union of 2 BA-Shape graphs

- **Tree-Cycles**
  - 8-level binary tree + six-node cycle motifs

- **Tree-Grid**
  - 8-level binary tree + 3-by-3 grid motifs
Experiments: Datasets

Real-world datasets:

- **MUTAG**
  - 4337 molecule graphs labeled according to their mutagenic effect on the Gram-negative bacterium S

- **Reddit-Binary**
  - 2000 graphs, each representing an online discussion thread on Reddit
  - nodes are users participating in a thread
  - edges indicate that one user replied to another user’s comment
  - labeled according to the type of user interactions: Question-Answer or Online-Discussion
Experiments: Methods

Baselines:

- **GRAD**
  - Gradient of the GNN’s loss function with respect to the adjacency matrix and the associated node features

- **ATT**
  - Graph attention GNN (GAT)
  - Learns attention weights for edges in the computation graph
Experiments: Results

Quantitative analyses (only on synthetic datasets)

<table>
<thead>
<tr>
<th>Explanation accuracy</th>
<th>BA-Shapes</th>
<th>BA-Community</th>
<th>Tree-Cycles</th>
<th>Tree-Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Att</td>
<td>0.815</td>
<td>0.739</td>
<td>0.824</td>
<td>0.612</td>
</tr>
<tr>
<td>Grad</td>
<td>0.882</td>
<td>0.750</td>
<td>0.905</td>
<td>0.667</td>
</tr>
<tr>
<td>GNNExplainer</td>
<td>0.925</td>
<td>0.836</td>
<td>0.948</td>
<td>0.875</td>
</tr>
</tbody>
</table>
Experiments: Results

Qualitative analyses: synthetic datasets
Qualitative analyses: synthetic datasets
Qualitative analyses: real-world datasets
Experiments: Results

Qualitative analyses: real-world datasets

B

<table>
<thead>
<tr>
<th></th>
<th>Computation graph</th>
<th>GNNEexplainer</th>
<th>Grad</th>
<th>Att</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutag</td>
<td><img src="image" alt="Mutag graph" /></td>
<td><img src="image" alt="Mutag GNNE explainer" /></td>
<td><img src="image" alt="Mutag Grad" /></td>
<td><img src="image" alt="Mutag Att" /></td>
<td><img src="image" alt="Mutag Ground Truth" /></td>
</tr>
<tr>
<td>Reddit-Binary</td>
<td><img src="image" alt="Reddit-Binary graph" /></td>
<td><img src="image" alt="Reddit-Binary GNNE explainer" /></td>
<td><img src="image" alt="Reddit-Binary Grad" /></td>
<td><img src="image" alt="Reddit-Binary Att" /></td>
<td><img src="image" alt="Reddit-Binary Ground Truth" /></td>
</tr>
</tbody>
</table>

Experts answering multiple questions
Methodology:
Joint learning of graph structural and node feature information
GNNExplainer: Joint learning of graph structural and node feature information

In the optimal solution, we have

\[ G_S \subseteq G_c(v) \quad X_S = \{ x_j | v_j \in G_S \} \]

important subgraph for the prediction

\[ X^F_S = \{ x^F_j | v_j \in G_S \}, \quad x^F_j = [x_{j,t_1}, \ldots, x_{j,t_k}] \text{ for } F_{t_i} = 1 \]

associated node features

important node features in \( G_S \) by a binary feature selector \( F \in \{0, 1\}^d \)

However, not all node features are equally important. We want to jointly learn
GNNEexplainer: Joint learning of graph structural and node feature information

How to define important features $\rightarrow$ Again, mutual information $MI$

$$\max_{G_S, F} MI \left( Y, (G_S, F) \right) = H(Y) - H(Y | G = G_S, X = X^F_S)$$
GNNExplainer: Joint learning of graph structural and node feature information

How to solve $F$?

Backpropagate gradients in

$$\max_{G_S,F} \text{MI} (Y, (G_S, F)) = H(Y) - H(Y|G = G_S, X = X^F_S)$$

- Step 1: Marginalize all feature subsets with a probability distribution
  - Use Monte Carlo to sample from empirical marginal distribution for nodes in $X_S$

- Step 2: backpropagate through a $d$-dimensional random variable $X$ as

$$X = Z + (X_S - Z) \odot F \quad \text{s.t.} \quad \sum_j F_j \leq K_F$$
Experiments:
Joint learning of graph structural and node feature information
Qualitative analyses: real-world datasets; node feature importance

A. Graph classification

- Input to GNN
- GNN’s Prediction
- Molecule’s mutagenicity: Ground Truth Feature Importance
- GNNExplainer
- Grad
- Not applicable

B. Node classification

- Computation graph of red node with node features
- Node’s structural role
- Not applicable
Extensions
GNNExplainer: Multi-instance explanations through graph prototypes

How did a GNN predict that a set of nodes all have the label?

→ a global explanation of each class
→ how identified subgraph relates to a graph structure that explains an entire class

● Step 1: Graph alignments
  ■ Choose a reference node $v_c$, as the mean embeddings of all nodes assigned to $c$
  ■ Take explanation for $G_S(v_c)$
  ■ Align it to explanations of other nodes assigned to class $c$

● Step 2: Prototypes
  ■ Aggregate aligned adjacency matrices into a graph prototype $A_{proto}$
  ■ $A_{proto}$: graph pattern shared between nodes in the same class
GNNExplainer: Extensions

GNNExplainer can be used

- in any machine learning task on graphs
- by any GNN models
- with a low computational complexity
Conclusions
Experiments: **Datasets**

**Pros:**
- Define the framework for GNN-based models’ explanation
- Propose a general, model-agnostic approach for graph structures and node features
- The performance is quite impressive

**Cons:**
- The optimization is not fully mathematically proved
- More quantitative results should be included