Two papers on Interpret GNN models

Presenter: Ji Gao

https://qdata.github.io/deep2Read

Outline

1. GNN Explainer: A Tool for Post-hoc Explanation of Graph Neural Networks
   - Method
     - Edge
     - Feature
     - Multi-instance explanation
     - Link prediction and Graph classification
   - Experiment result
2. Interpretable Graph Convolutional Neural Networks for Inference on Noisy Knowledge Graphs
   - Method
   - Result

GNN Explainer: A Tool for Post-hoc Explanation of Graph Neural Networks[YBY+19]

- Deep learning model is too complicated, hard to understand.
- Propose GNNEXPLAINER, an model agnostic approach for providing interpretable explanations for predictions of any GNN models
  - Generate a simplified graph model(Subgraph)
  - Maximize the mutual information between the prediction between full model and simplified graph

Explainer Problem

- Graph $G = (V, E, X)$
- Feature $X = x_1, ..x_n, x_i \in \mathcal{R}^d$
- Label function $f : V \rightarrow 1..C$
- Explanation: Edge/Feature on what model learned

GNN Explainer

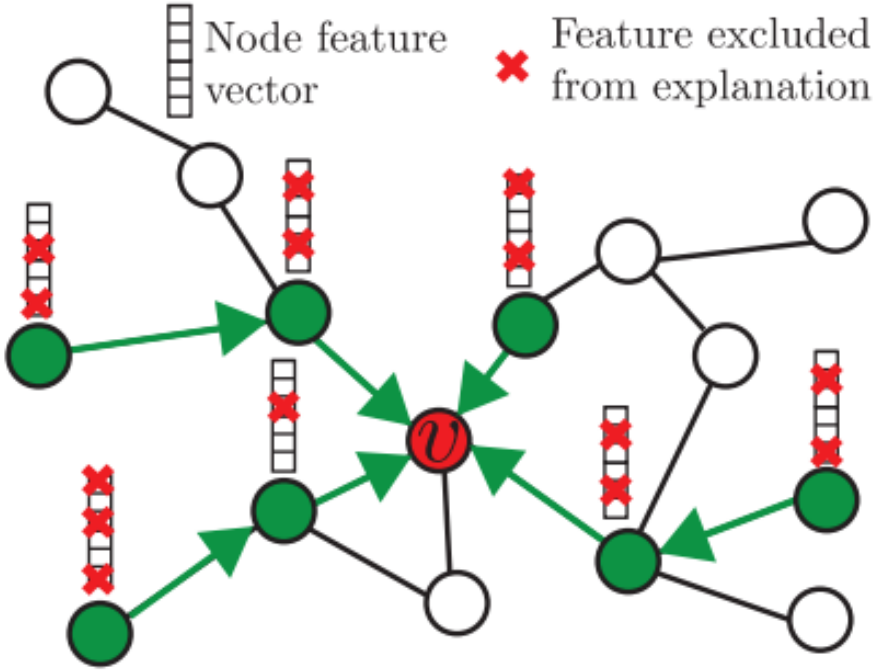

Figure 2: For v's explanation G_S (in green), GNNEXPLAINER identifies what feature dimensions of G_S's nodes are essential for prediction at v by learning a node feature mask M_T.

Single Instance Explanation - Edges

- Goal: Maximize mutual information gain

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

- Impose a size constraint $|G_S| \leq k$
- In fact, $H(Y)$ is a constant. Only need to minimize the second term.

$$H(Y|G = G_S, X = X_S) = -E_{Y|G_S, X_S}[\log P_\phi(Y|G = G_S, X = X_S)]$$

- Similar to adversarial sample.

Optimization on edges

- Treat G_S in a continuous way: $A_S \in [0,1]^{n \times n}$
- By Jensen's inequality

$$\min_{\mathcal{G}} \mathbb{E}_{G_S \sim \mathcal{G}} H(Y|G = G_S, X = X_S) \leq \min_{\mathcal{G}} H(Y|G = \mathbb{E}_{\mathcal{G}}[G_S], X = X_S)$$

- According to mean field variational approximation, decompose $\mathcal{G}$ into $P_{\mathcal{G}}(G_S) = \prod_{(i,j) \in G_c(v_i)} A_{S\,ij}$
- True objective: find a mask M, that $\sigma(M)$ is a 0,1 mask function $\in \{0,1\}^{N \times N}$

$$\min_M -\mathbb{E}_{Y|A_S \odot \sigma(M), X_S} [\log P_\Phi(Y|G = A_S \odot \sigma(M), X = X_S)]$$

$$= \min_M -\sum_{c=1}^{C} \mathbf{1}[y = c] \log P_\Phi(Y = y|G = A_S \odot \sigma(M), X = X_S).$$

Features

- A binary feature subset $T \in \{0,1\}^D$, that:

$$MI\left(Y,(G_S,T)\right) = H(Y) - H(Y|G = G_S, X = X_S^T)$$

- Use a Monte Carlo estimation:

$$P_\Phi(Y|G_S, X_S^T) = \sum_{x_i \in D \setminus T} P(x_i) P_\Phi(Y|G = G_S, X = \{X_S^T, x_i\})$$

- In real case, sample a random variable Z of dimension $n \times d$ from the marginal distribution of X_S, and then, $X = Z + (X_S - Z) \odot M_T$

Regularization

- Explanation size: Explanation size can be set with a regularization term
- Discrete mask: Discrete Mask can be encouraged by regularizing the cross entropy of M and M_T:

$$H(M) = -\sum_{1 \leq i,j \leq n} (M_{i,j} \log M_{i,j} + (1 - M_{i,j}) \log(1 - M_{i,j}))$$

- Prior knowledge: If some prior of labels exists, it can be added with in the form of Laplacian smoothing $f^T L_S f$

Multi-instance explanation

- Construct a prototype graph of the classified node
  1. Create a reference node by computing the mean of all node embeddings, and pick the nearest node in the dataset.
  2. Align every subgraphs $G_S(v)$ to the reference node v_c Matching the subgraphs lead to an optimization:

$$\min_P |P^T A_v P - A^*| + |P^T X_v - X^*|.$$

However, in this case

$$A_{\text{proto}} = \frac{1}{n} \sum_i A_i$$

Link prediction and Graph classification

- Learn masks for two nodes for link prediction

$$P_\Phi(Y|G_1 = A_{S_i} \odot M_1, G_2 = A_{S_j} \odot M_2, X = X_S),$$

- Use Y as the graph labels in graph classification

Examples

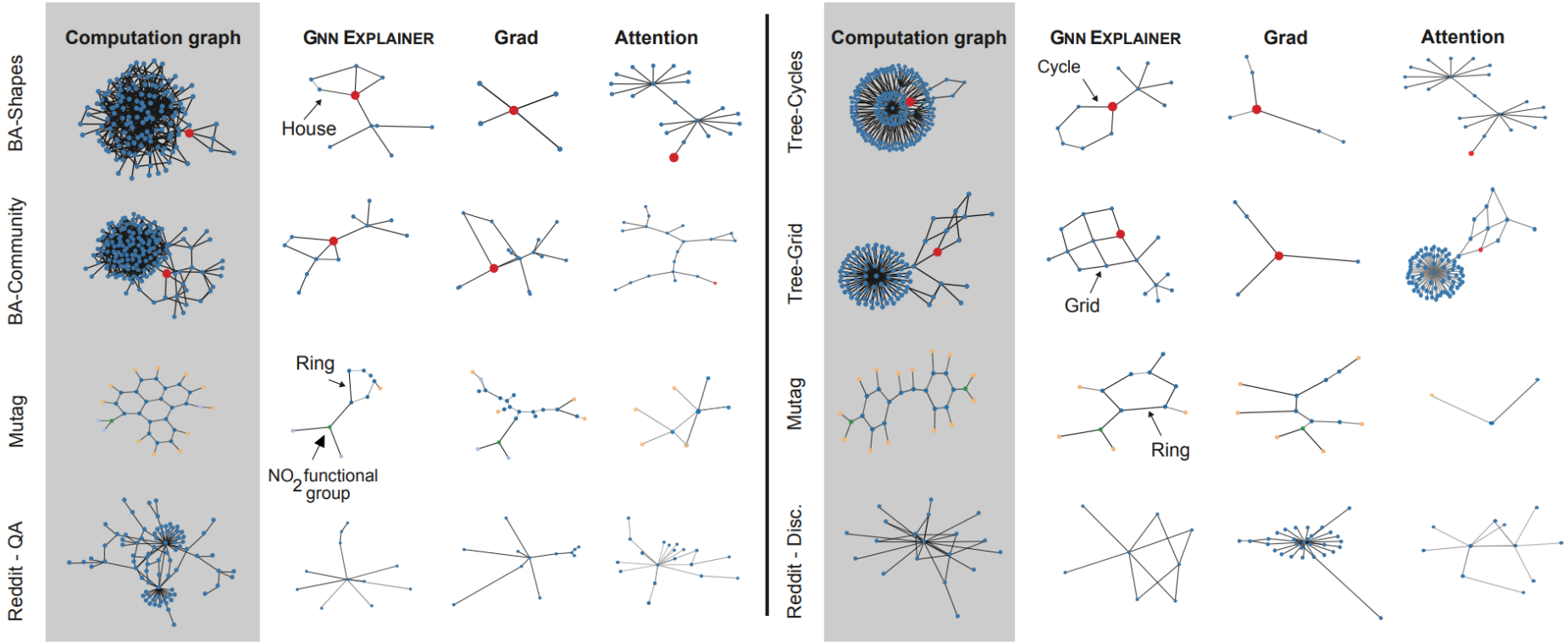

Figure 3: Examples of single-instance important subgraphs. The red node is the explained node.

Feature importance

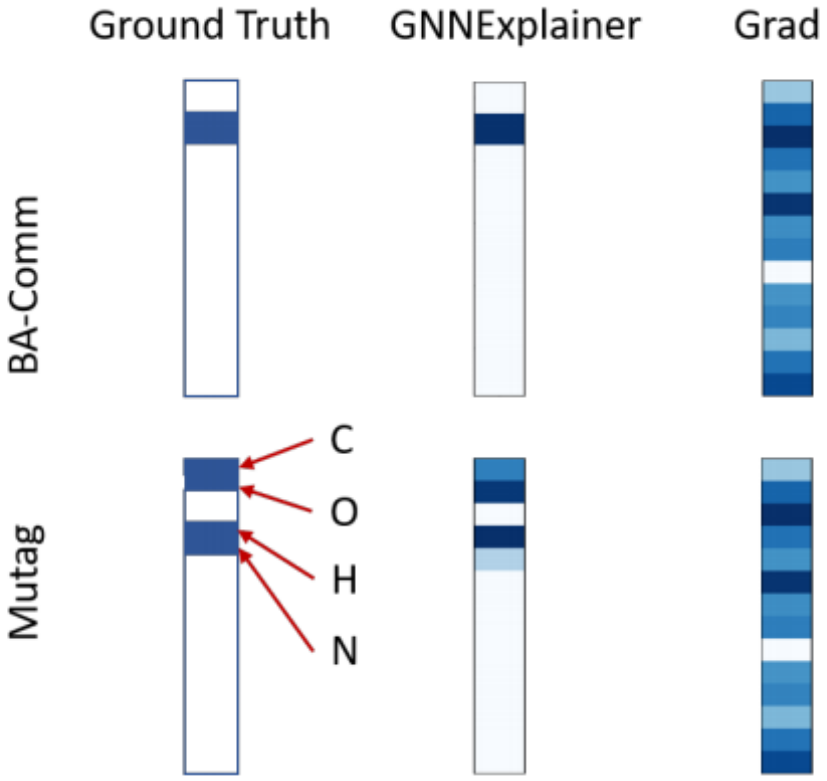

Figure 4: Feature importance visualization

Quantitative result

Baselines:

- Attention value in GAT
- Gradient

Table 1: GnnExplainer compared to baselines in identifying subgraphs using AUC.

	BA-Shapes	BA-Community	Tree-Cycles
GAT	0.957	0.974	0.914
Grad	0.936	0.883	0.885
GNN	**0.991**	**0.993**	**0.975**

Multi-instance prototype

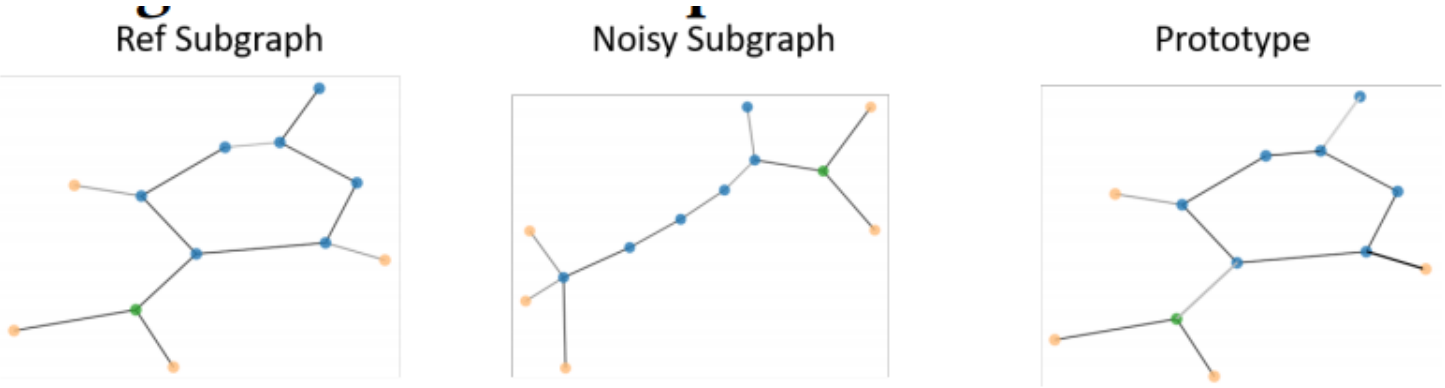

Figure 5: GnnExplainer is able to provide a prototype for a given node class, which can help identify functional subgraphs, e.g. a mutagenic compound from the Mutag dataset.

Interpretable Graph Convolutional Neural Networks for Inference on Noisy Knowledge Graphs[NBL+18]

- New GCNN on **link prediction**
- On biomedical knowledge base
- Provide experiment result and Visualization

New GCNN formulation

- Use an attention matrix $C_r \in \mathcal{R}^{N \times N}$, on different type of edge

$$H^{(l+1)} = \sigma\left(B^{(l)} + \sum_{r \in \mathcal{R}} (C_r \odot A_r)(H^{(l)} W_r^{(l)})\right) \tag{1}$$

- Also learn Relation matrix R, which is a simple embedding matrix. Link prediction:

$$f(e_s, R_r, e_o) = e_s^T R_r e_o \tag{2}$$

- Use a fixed total budget:

$$C_{r,i,j} = \frac{1}{\sum_{r' \in \mathcal{R}} \sum_{j' \in \mathcal{N}_i^r} |\hat{C}_{r',i,j'}|} |\hat{C}_{r,i,j}| \tag{3}$$

Performance

	Hits@10				MRR			
Algorithm	100%	50%	Skip	Noised	100%	50%	Skip	Noised
DistMult	43.2	20.2	N/A	20.6	23.9	8.69	N/A	8.93
ComplEx	44.1	24.1	N/A	24.3	25.9	10.9	N/A	11.0
GCNN	47.5	33.2	25.8	21.4	27.2	16.8	13.3	11.1
GCNN w/att	**48.2**	**34.7**	**34.0**	**35.6**	**28.3**	**18.5**	**18.8**	**19.1**
R-GCN+ ([21])	41.7	-	-	-	24.9	-	-	-

Table 1: Performance on the FB15k-237 Dataset. Our results compare favorably with those reported in previous GCNN studies.

Visualization

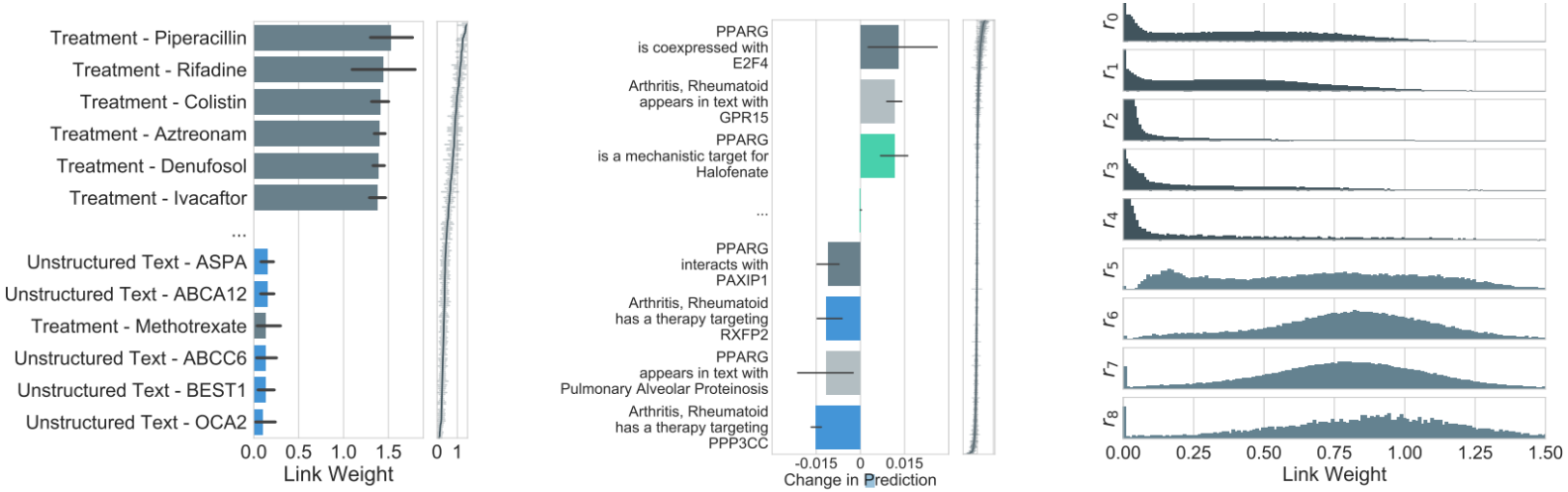

Figure 3: **Left**: Ranking of a node's influencers. The top 6 and bottom 6 known weighted-edges (+/- standard error) for cystic fibrosis are visualized as an example. **Centre**: Analyzing the drivers of link prediction, evaluating the possibility of PPARG being a drug target for Rheumatoid Arthritis. Each bar demonstrates the effect that fact has on prediction score (+/- standard error). **Right**: Distribution of edge weights across $r \in \mathcal{R}$ in the biomedical knowledge graph.

Reference

- Daniel Neil, Joss Briody, Alix Lacoste, Aaron Sim, Paidi Creed, and Amir Saffari, *Interpretable graph convolutional neural networks for inference on noisy knowledge graphs*, arXiv preprint arXiv:1812.00279 (2018).
- Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec, *Gnn explainer: A tool for post-hoc explanation of graph neural networks*, arXiv preprint arXiv:1903.03894 (2019).