CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks

AISTATS 2022

Ana Lucic [1] **Maartje ter Hoeve** [1] **Gabriele Tolomei** [2] **Maarten de Rijke** [1] **Fabrizio Silvestri** [2]
[1] University of Amsterdam [2] Sapienza University of Rome

Background

近年来机器学习在计算机视觉，自然语言处理，会话助理等领域取得了惊人的成绩。随着机器学习的发展，人们对于机器学习的可解释性的需求也在随之增加。XAI（Explainable AI）指的就是一系列用于解释机器学习模型，使其能够被人类所理解的方法。

反事实解释是用来解释形如如下的问题的：“如果X被改变，Y就不会发生”

目前，反事实解释已经广泛应用于表格、图像、文本数据。一些文章将模型看作黑盒模型，通过对输入改动，来形成反事实解释，另一些方法从模型的内部入手，通过改变模型的工作机理，来形成反事实的解释。

但是上述方法存在以下问题：
①上述反事实解释方法仅针对于特征扰动，无法处理图数据
②而图数据的解释性（如Gnnexplainer）方法尚未有反事实解释
③图数据的解释性方法都无法自动识别最小的子图

Contribution

①形式化 GNN的反事实解释问题
②提出了CF-GNNExplainer，一种用于为GNN提供反事实解释的方法
③提出了一个整体评估GNN的CF解释的实验设置

Related problems

■ CF examples and Adversarial Attacks:

相似点:

- 它们都代表了能够引起模型预测发生变化的对输入样例进行最小扰动的实例

不同点:

- 目的不同：对抗性示例旨在欺骗模型，而 CF 示例旨在解释预测
- 方法不同：在图数据的上下文中，对抗攻击方法通常对整个图进行最小的扰动，目的是降低整体模型性能，而不是攻击单个节点。相反，反事实方法是为单个节点生成CF示例，而不是识别整个图的扰动

■ Matrix Sparsification

给定一个权重矩阵 W，学习一个二元稀疏矩阵，该矩阵与 W 逐元素相乘，使得 W 中的一些条目被清零。
在CF-GNNExplainer中，对邻接矩阵进行矩阵稀疏化处理，使得一些边被删除

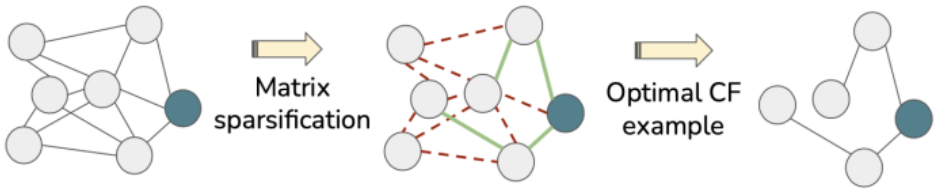

Figure 1: Intuition of counterfactual example generation by CF-GNNExplainer.

Problem Formulation

通常情况下:

$$f(x) \neq f(\overline{x})$$

$$\Delta_x^* = \overline{x}^* - x$$

图数据下:

$$\overline{v} = (\overline{A_v}, x)$$

$$f(v) \neq f(\overline{v})$$

$$\mathcal{L} = \mathcal{L}_{pred}(v, \bar{v} \mid f, g) + \beta \mathcal{L}_{dist}(v, \bar{v} \mid d)$$

GNN model: $f(A_v, X_v; W) = \text{softmax}\left[(D_v + I)^{-1/2}(A_v + I)(D_v + I)^{-1/2} X_v W\right]$

update

CF generation function: $g(A_v, X_v . W; P) = \text{softmax}\left[\bar{D}_v{}^{-1/2}(P \odot A_v + I)\bar{D}_v{}^{-1/2} X_v W\right]$

update fix

$$CF = P \odot A_v$$

$$\mathcal{L} = \mathcal{L}_{pred}(v, \bar{v} \mid f, g) + \beta \mathcal{L}_{dist}(v, \bar{v} \mid d)$$

NLL loss

$$\mathcal{L}_{pred}(v, \bar{v} | f, g) = - \mathbb{1}\left[f(v) = f(\bar{v})\right] \cdot \mathcal{L}_{NLL}(f(v), g(\bar{v}))$$

Algorithm 1 CF-GNNEXPLAINER: given a node $v = (A_v, x)$ where $f(v) = y$, generate the minimal perturbation, $\bar{v} = (\bar{A}_v, x)$, such that $f(\bar{v}) \neq y$.

Input: node $v = (A_v, x)$, trained GNN model f, CF model g, loss function $\mathcal{L}$, learning rate α, number of iterations K, distance function d.

$f(v) = y$ *# Get GNN prediction*
$\hat{P} \leftarrow J_n$ *# Initialization*
$\bar{v}^* = [\,]$

for K iterations **do**
  $\bar{v} = \text{GET\_CF\_EXAMPLE}()$
  $\mathcal{L} \leftarrow \mathcal{L}(v, \bar{v}, f, g)$ *# Eq 1 & Eq 5*
  $\hat{P} \leftarrow \hat{P} + \alpha \nabla_{\hat{P}} \mathcal{L}$ *# Update $\hat{P}$*
end for

Function GET_CF_EXAMPLE()
  $P \leftarrow \text{threshold}(\sigma(\hat{P}))$
  $\bar{A}_v \leftarrow P \odot A_v$
  $\bar{v}_{cand} \leftarrow (\bar{A}_v, x)$
  if $f(v) \neq f(\bar{v}_{cand})$ **then**
    $\bar{v} \leftarrow \bar{v}_{cand}$
    if not $\bar{v}^*$ **then**
      $\bar{v}^* \leftarrow \bar{v}$ *# First CF*
    else if $d(v, \bar{v}) \leq d(v, \bar{v}^*)$ **then**
      $\bar{v}^* \leftarrow \bar{v}$ *# Keep track of best CF*
    end if
  end if
  return $\bar{v}^*$

Table 2: Results comparing our method to RANDOM, 1HOP, and RM-1HOP. Below each metric, ▼ indicates a low value is desirable, while ▲ indicates a high value is desirable.

	TREE-CYCLES				TREE-GRID				BA-SHAPES			
Method	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲
RANDOM	**0.00**	4.70	0.79	0.63	**0.00**	9.06	0.75	0.77	**0.00**	503.31	0.58	0.17
1HOP	0.32	15.64	0.13	0.45	0.32	29.30	0.09	0.72	0.60	504.18	0.05	0.18
RM-1HOP	0.46	2.11	0.89	—	0.61	2.27	0.92	—	0.21	10.56	0.97	**0.99**
CF-GNNEXPLAINER	0.21	**2.09**	**0.90**	**0.94**	0.07	**1.47**	**0.94**	**0.96**	0.39	**2.39**	**0.99**	0.96

Fidelity:有效性（文中指原始预测结果与更改后的预测结果，越小越好）

Size:尺寸（删除的边数量，越小越好）

Sparsity:稀疏性（删除的边占所有边的比例，越接近1越好）

Acc:准确度（生成的解释子图是不是处于motif内部）

Method	TREE-CYCLES				TREE-GRID				BA-SHAPES			
	Fid. ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲
GNNEXP ($S = 1$)	0.65	1.00	0.92	0.61	0.69	1.00	0.96	0.79	0.90	1.00	0.94	0.52
GNNEXP ($S = 2$)	0.59	2.00	0.85	0.54	0.51	2.00	0.92	0.78	0.85	2.00	0.91	0.40
GNNEXP ($S = 3$)	0.56	3.00	0.79	0.51	0.46	3.00	0.88	0.79	0.83	3.00	0.87	0.34
GNNEXP ($S = 4$)	0.58	4.00	0.72	0.48	0.42	4.00	0.84	0.79	0.83	4.00	0.83	0.31
GNNEXP ($S = 5$)	0.57	5.00	0.66	0.46	0.40	5.00	0.80	0.79	0.81	5.00	0.81	0.27
GNNEXP ($S = \mathrm{GT}$)	0.55	6.00	0.57	0.46	0.35	11.83	0.53	0.74	0.82	6.00	0.79	0.24
CF-GNNEXPLAINER	**0.21**	2.09	0.90	**0.94**	**0.07**	1.47	0.94	**0.96**	**0.39**	2.39	0.99	**0.96**

S:保留的边数

Conclusion

- 本文提出了一种用于图神经网络的反事实解释方法：CF-GNNExplainer。该方法可以保证在删除最少边的前提下，为示例提供反事实的解释。
- 作者在三个通用的GNN解释数据集上进行实验，验证了CF-GNNExplainer的有效性
- 目前该方法仅适用于图结构的分类问题，并没有考虑节点特征问题。后续会考虑节点的分类问题，也会考虑特征的扰动问题。