CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks

CHADAL, Marion
marion.chadal@polytechnique.edu

WINTER, Alexis
alexis.winter@polytechnique.edu

Structured Data: Learning and Prediction
April 7, 2024

1 Paper analysis

1.1 Task description and evaluation metrics

Task Description in CF-GNNExplainer The paper focuses on generating counterfactual explanations (CF explanations) for predictions made by graph neural networks (GNNs). A counterfactual explanation essentially answers the question: "What is the smallest change we can make to the input graph that would cause the GNN to make a different prediction?"

The aim of the method (CF-GNNExplainer) is finding the optimal CF explanation according to some distance function d. If we were to note the instance x and the perturbed instance $\bar{x}$. The resulting optimal CF explanation is $\bar{x}^*$ such that $f(x) \neq f(\bar{x}^*)$ and $\Delta_x^* = \bar{x}^* - x$

We can summarize the tasks in these two elements

- **Input:** The input consists of a graph and a GNN model that has already been trained on some data.
- **Target:** The target is to find a minimal modification to the graph such that the GNN's prediction for that specific graph changes compared to the original prediction. This modification is ideally achieved by removing the smallest number of edges possible.

Evaluation Metrics in CF-GNNExplainer The paper proposes four metrics to evaluate the performance of CF-GNNExplainer:

- **Fidelity:** The fidelity measures the proportion of nodes where the original prediction from the GNN model still holds true after applying the counterfactual explanation. Here for good counterfactuals, we want the original prediction to change, so ideally fidelity should be low.
- **Explanation Size:** This metric directly measures the number of edges removed from the original graph to create the counterfactual explanation. Since the goal is to have minimal explanations, a small value for explanation size is desirable.
- **Sparsity:** This metric complements explanation size by measuring the proportion of edges removed relative to the total number of edges in the original graph (represented by Av). A value of 0 indicates all edges were removed, while a value of 1 signifies only minimal removals. Similar to explanation size, sparsity should be close to 1 for ideal counterfactual explanations.
- **Accuracy:** This metric focuses on how well the explanations correspond to actual changes in the prediction, but only for specific nodes. Accuracy is calculated for nodes originally predicted as belonging to certain motifs. Here, a "correct" explanation is one that exclusively removes edges within the motif, leading to a different prediction. Given the desire for minimal explanations, achieving high accuracy translates to explanations that modify only the relevant edges within the motifs.

A good explanation method should have low fidelity and explanation size and a high sparsity and accuracy.

1.2 Type of method

Method for model-agnostic counterfactual explanation The paper proposes a method for model-agnostic counterfactual explanation.

- **Model-agnostic:** The CF-GNNExplainer method itself is not specific to a particular GNN model. It can be applied to explain predictions made by any GNN model as long as the model takes graphs as input and produces a prediction for each node in the graph.
- **CF explanation:** The method focuses on generating counterfactual explanations. This means it aims to identify the minimal changes (edge removals) to the input graph that would cause the GNN model to make a different prediction for a particular node.

Therefore, CF-GNNExplainer bridges the gap between the GNN model and human interpretability by providing counterfactual explanations in a model-agnostic way.

1.3 Hypothesis space

The Hypothesis Space of Graph Neural Networks (GNNs) GNNs can be viewed as learning a family of functions that map graph structures to node representations and then to predictions. The specific function within this family depends on the training data and model architecture.

The choice of GNN architecture (e.g., number of layers, message passing functions) influences the expressive power of the model. More complex architectures can potentially learn a wider range of functions within the hypothesis space.

An important property of GNNs is their permutation invariance. This means that the order in which nodes are processed within the graph shouldn't affect the final predictions. This property ensures that the learned representations and predictions are robust to re-arrangements of the nodes in the graph, as long as the underlying relationships remain the same. In essence, GNNs don't have a single, predefined hypothesis space.

1.4 Loss, penalties or constraints

The Loss Function in CF-GNNExplainer Here's a breakdown of the loss function:

$$\mathcal{L} = \mathcal{L}_{pred}(v, \bar{v} \mid f, g) + \beta \mathcal{L}_{dist}(v, \bar{v} \mid d), \tag{1}$$

The loss function is a combination of two different loss terms:

- **Prediction Loss:** Encourages the GNN's prediction for the original node v to differ from the prediction for the counterfactual example $\bar{v}$. We use the negative log-likelihood (NLL) loss for $\mathcal{L}_{pred}$:

  $$\mathcal{L}_{pred}(v, \bar{v} | f, g) = -\mathbb{1}\left[f(v) = f(\bar{v})\right] \cdot \mathcal{L}_{NLL}(f(v), g(\bar{v})). \tag{2}$$

  Includes an indicator function to ensure the loss is only active when the predictions for the original and counterfactual nodes are the same ($f(v) = f(\bar{v})$). This effectively penalizes explanations that don't change the prediction.
- **Distance Loss:** Encourages the counterfactual example $\bar{v}$ to be similar to the original node v in terms of structure. In the paper, it's the element-wise difference between the adjacency matrices of the original and counterfactual nodes but it can be any differentiable distance function. (For undirected graphs, it's divided by 2 to account for symmetry in the adjacency matrices)
- **Beta:** Beta (β) controls the relative importance of the distance loss compared to the prediction loss.

1.5 Learning algorithm (complexity in time /in memory)

The algorithm We have an overview of the algorithm in Algorithm 1. We can summarize each step as followed:

- **Initialization:** Given a node in the test set v, the original prediction from the GNN model f is obtained. The algorithm initializes a matrix $\hat{P}$ as a matrix of ones (denoted as $\mathbf{J}_n$), representing initially retaining all edges.

  The algorithm then execute the next three steps K times

- **CF Example Generation and Evaluation:** To find a counterfactual (CF) example, the algorithm first computes P by thresholding $\hat{P}$. Then it uses P to obtain the sparsified adjacency matrix that gives a candidate CF example, $\bar{v}_{\text{cand}}$.

  The candidate CF example $\bar{v}_{\text{cand}}$ is fed into the original GNN model f. If f predicts a different output than for the original node, a valid CF example $\bar{v}$ is found. The algorithm keeps track of the "best" CF example according to a specified distance metric d and returns this as the optimal CF example $\bar{v}^*$ after K iterations.

- **Loss Computation and Update:** The algorithm computes the loss following Eqn 1 and 2, and updates $\hat{P}$ based on the gradient of the loss.

- **Final Explanation Retrieval:** After the K iterations, the optimal CF explanation $\Delta_v^* = v - \bar{v}^*$ is retrieved, representing the difference between the original node and the optimal CF example.

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 $\alpha$, 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
  $\bar{v}$ = GET_CF_EXAMPLE()
  $\mathcal{L} \leftarrow \mathcal{L}(v, \bar{v}, f, g)$   # Eq 1 & Eq 2
  $\hat{P} \leftarrow \hat{P} + \alpha \nabla_{\hat{P}} \mathcal{L}$   # 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 $\bar{v}^*$
```

The learning algorithm Since the model is made to generate optimal CF examples based on predictions made by a GNN, the model does not really have a learning phase as usually intended for machine learning models. However the model need to use an already trained GNN so we can consider that the learning algorithm can correspond to the learning algorithm of the GNN model being used.

1.6 Inference step: definition + algorithm (complexity in time /in memory)

Inference step Since the model does not have a learning phase, the inference step here correspond to the entirety of Algorithm 1.

Complexity The method has a time complexity of $O(KN^2)$, where N is the number of nodes in the subgraph neighbourhood and K is the number of iterations.

1.7 Model selection and performance assessment methodology

We have already talked about the evaluation metrics of the method, so in this part we will elaborate on which dataset and baseline the model was evaluated and what were the best parameters for the model.

Datasets The model and the baseline are evaluated on 3 different datasets, tree-cycles, tree-grids, ba-shapes from Ying et al. (2019) [5]. They are all undirected graphs and synthetic datasets. Authors could not use real world node classification dataset, as there is none available. The classification task is to determine whether or not the nodes are part of a certain motif. Each dataset has different motifs: 6-cycle motifs for tree-cycles, tree-grids has3×3 grids as the motifs, ba-shapes has house-shaped motifs. The base graph is a binary tree for the two first dataset and a Barabasi-Albert graph for ba-shapes.

Baseline The model is compared to 4 baseline models:

- **Random:** This baseline serves as a sanity check. It randomly initializes the perturbation matrix with values between -1 and 1. It then follows the same procedure as the proposed method (CF-GNNExplainer).
- **1Hop:** This baseline focuses on the 1-hop neighborhood of the target node v (its ego-graph). It simply keeps all edges within this ego-graph, effectively not modifying the local structure around the node.
- **RM-1Hop:** Similar to the ego-graph keep baseline, this approach focuses on the 1-hop neighborhood of v. However, it removes all edges within the ego-graph, resulting in a completely isolated node.
- **GNNExplainer:** This baseline leverages the GNNExplainer method by Ying et al. (2019). GNNExplainer identifies the most relevant edges (subgraph) for the prediction of node v. However, it's important to note that GNNExplainer is not specifically designed for counterfactual explanation. Unlike CF-GNNExplainer, it requires the user to pre-define the size of the relevant subgraph, so the minimality of the CF example found cannot be evaluated.

Hyperparameter search There are 5 different hyperparameter the model can be tuned from:

- **Number of iterations** (K)**:** This hyperparameter controls the number of times the CF-GNNExplainer iterates to find a counterfactual example. The experiment tests values of $K \in \{100, 300, 500\}$. In the reported results, the best performance was achieved with $K = 500$.
- **Trade-off parameter** (β)**:** This hyperparameter balances the importance between the prediction loss (encouraging different predictions for original and counterfactual nodes) and the distance loss (encouraging minimal changes to the graph structure). The experiment tests values of $\beta \in \{0.1, 0.5\}$. The best results were achieved with $\beta = 0.5$.
- **Learning rate** (α)**:** This hyperparameter controls the step size taken during the optimization process of updating the perturbation matrix. The experiment tests values of $\alpha \in \{0.005, 0.01, 0.1, 1\}$. The best performance was achieved with $\alpha = 0.1$.
- **Nesterov momentum** (m) **(when applicable):** This hyperparameter is specific to optimizers that support momentum, influencing how much weight is given to previous gradients during the update process. The experiment tests values of $m \in \{0, 0.5, 0.7, 0.9\}$. For the tree-cycles and tree-grid datasets m was set at 0. However, for ba-shapes, $m = 0.9$ achieved better performance.
- **Optimizer:** The experiment compares three different optimizers (Adam, SGD, AdaDelta) for updating the perturbation matrix during the search for counterfactual explanations. SGD (Stochastic Gradient Descent) achieved the best overall performance for all three datasets.

2 Paper assessment

The paper focuses on a fairly recent and important area of research in Deep Learning, in line with the desire to make neural networks interpretable: explainable AI. This issue is well presented, and the usefulness of developing explainability methods is clearly understood.

CF-GNNExplainer focuses on node classification, in contrast to competing methods that focus on graph-level classification. It is based on a simple yet innovative principle, as authors propose the first method based on counterfactuals, and it is well explained in the paper. Counterfactuals are clear and concise: if a specific feature's value changes, the prediction will in turn change to the predefined output. At the same time, they do not require to have access to the model or the data behind it. However, CF-GNNExplainer takes existing concepts and puts them together to improve them, without really building anything radically new.

One drawback, although not the fault of the authors but important to note, is that the model could not be tested on real world node classification dataset. It would be worth mobilizing resources to build such a dataset, as it would enable us to give credibility (or not) to the proposed methods. Indeed, one of the objectives of explainable AI is to make the internal workings of AI models more transparent, so that humans can understand how decisions are made. Explainability metrics therefore need to be robust to real world datasets.

The Rashomon effect [1] should have been mentioned: for each instance, there are multiple counterfactual explanations. Rashomon is a Japanese movie in which the murder of a Samurai is told by different point of views. Each people provides a plausible explanation of the event, yet they contradict one another. Analogously, in the context of counterfactual explanations, there are also multiple different counterfactual explanations. Each counterfactual tells a different "story" of how a certain outcome was reached. For instance, one counterfactual might suggest altering feature A, whereas another advises keeping A unchanged but modifying feature B, presenting a clear contradiction. However, let's remind that numerical experiments were ran on synthetic datasets, so this issue did not arise.

As for reproducibility, which will be discussed in more detail later, we managed to obtain the same results as those presented in the paper. On the other hand, we had to adjust the code of the baseline scripts, and were unable to get everything to work. The same applies to the explanation size metric.

3 Numerical experiments

As mentioned earlier, one of the limitations of CF-GNNExplainer is that it has not been evaluated on a real world node classification dataset, due to the non-existence of such a dataset which includes explainability ground truth. There only exist such dataset for graph classification task. For example, GraphSVX [3] performs a graph classification task using motifs in molecules graphs from the MUTAG dataset as ground truth.

The advantage of having access to ground truth is that we can calculate the accuracy of our explainer. This is an easily interpretable metric that reflects a model's performance in the task assigned to it. Otherwise, explainability can be measured through the three other metrics: fidelity, sparsity and explanation size.

Unlike the GraphSVX methodology for annotating MUTAG graphs, which is based on the knowledge that carbon rings with NH_2 or NO_2 groups are mutagenic, MUTAG cannot be annotated at the level of molecule atoms. Or at least, we are not chemists, and we don't have the knowledge to do so.

On the other hand, with other types of data, we are able to obtain ground truth based on our intuition, as a graph representing links between individuals, such as connections on a social network. If our GNN aims to classify nodes according to their level of connection to others, we could explain its predictions by basic but meaningful notions of the level of connectivity between nodes.

The following implementation adopts this intuition. Ground truth are generated through three different node connectivity measures on a Facebook graph dataset used in [4] that represents social circles.

3.1 Methodology

Several graphs networks were available in the Facebook Social circles dataset. It consists of 10 circles (i.e. friends lists) from Facebook. Facebook data was collected from survey participants using the app. The dataset includes node features (profiles), circles, and ego networks. Information details about the features are not available. But node classification in a social network could for example be used to predict the political affiliation or the age of the user, based on its relations with the other users.

Experiments were ran only on two circles, as the others had less than 3 edges for a huge amount of nodes. The nodes where features were missing were removed. The ground truth for each node was added, following the three measures we will discuss.

The average measure of ground truth was taken as a threshold to decide whether a subgraph should be in a motif or not (allowed to get a higher average node degree in the motif). The datasets statistics with node degree as ground truth are available in Table 1, while representations of graphs and their motif can be seen in Table 2.

	Circle 1	**Circle 2**
Total nodes	1045	347
Total edges	639	2513
Nodes in motif	486	40
Edges in motif (GT)	333	530
Avg node degree	1.22	14.48
Avg node degree in motif	9.25	26.50

Table 1: Datasets statistics

The datasets have characteristics similar to the synthetic ones used in CF-GNNExplainer. Their difference remains in their explainability ground truth, as the motifs rely on specific architectures of subgraph (house shape for BA-SHAPES, tree shape for TREE-GRIDS and TREE-CYCLES).

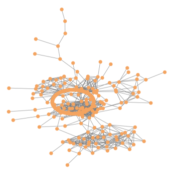
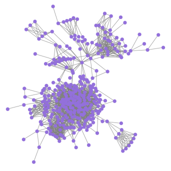
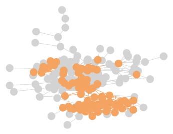
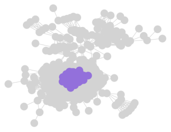
	Circle 1	Circle 2
Graph		
Motif		

Table 2: Graph and motif representations in Circles 1 and 2

In the social circles Facebook dataset, the ground truth for explainability had to be able to express the connectivity of a node to others in a graph (the connectivity of a Facebook user to others, in terms of friends list). Three metrics were chosen:

- **Degree centrality:** corresponds to the fraction of nodes a node v is connected to. This value is normalized by dividing by the maximum possible degree in a simple graph n - 1 with n the number of nodes in G.
- **Clustering coefficient:** it is the fraction of possible triangles through a node v that exist. This method is inspired by the one used in GraphSVX to label MUTAG molecules.
- **Eigenvector centrality:** The centrality for node v is the *v-th* element of a left eigenvector associated with the eigenvalue λ of maximum modulus that is positive.This measure is inspired by the one used in PageRank [2]. The difference is that they use the in-degree for a node, which corresponds to the number of edges pointing to the node, and is therefore suited for directed graph. As circles 1 and 2 are undirected graph, the simpler version is adopted.

3.2 Results

To choose hyperparameters configurations and compare connectivity measures, we used the same metrics as the ones of the CF-GNNExplainer paper: fidelity, sparsity, and accuracy. The explanation size was not available. The results are shown in Tables 3 and 4.

	Circle 1			Circle 2		
Connectivity measure	Fid.	Spars.	Acc.	Fid.	Spars.	Acc.
Degree centrality	0.87	0.99	0.92	0.86	0.99	0.92
Clustering coefficient	0.74	0.97	0.92	0.70	0.99	0.96
Eigenvector centrality	—	—	—	0.87	0.99	0.98

Table 3: Results comparing CF-GNNExplainer on different methods for ground truth generation.

It was also important to compare the performances of CF-GNNExplainer on the Facebook social circles dataset with the baselines. Unfortunately, we did not manage to make the scripts run for every baselines, so we were only able to compare CF-GNNExplainer with 1HOP.

	Circle 1			Circle 2		
Connectivity measure	Fid.	Spars.	Acc.	Fid.	Spars.	Acc.
Degree centrality	0.99	0.88	—	0.74	0.9	0.46
Clustering coefficient	0.99	—	—	0.58	0.9	0.25
Eigenvector centrality	0.9	0.95	—	0.61	0.93	0.38

Table 4: Results comparing 1HOP on different methods for ground truth generation.

The fidelity metric indicates that CF-GNNExplainer generally presents lower fidelity in Circle 1 compared to 1HOP, suggesting that it might be producing more counterfactuals that deviate from the original predictions. However, in Circle 2, CF-GNNExplainer exhibits higher fidelity, indicating a reversal in performance favorability.

Regarding sparsity, CF-GNNExplainer consistently achieves higher or comparable levels, signifying its efficiency in generating more concise explanations, which we want, as sparser explanations are often easier to comprehend.

Accuracy results, though sparse, hint at CF-GNNExplainer's superior capability in accurately identifying crucial features, as evidenced in Circle 2. The inconsistency and lack of data for some metrics, especially accuracy, underscore the challenges inherent in our ground-truth generation approach.

Our analysis reveals the intricate dependencies between the method employed and the ground-truth data's nature, leading to mixed results that resist simple generalization. It also highlights the comparative strengths of CF-GNNExplainer in generating sparser and, in some cases, more accurate counterfactual explanations. However, the observed variability across different metrics and circles cautions against drawing definitive conclusions without further investigation into the underlying causes of these disparities.

The code is available in a Github repository.

4 Conclusion

The CF-GNNExplainer method stands out as a model-agnostic approach designed to generate counterfactual explanations for graph neural networks (GNNs). It aims to identify the minimal set of modifications—specifically, edge removals—required to alter the prediction of a GNN for a given node in the input graph. This method leverages a loss function that balances the need for prediction change (fidelity) and minimal structural alteration (sparsity), thereby ensuring the generated counterfactuals are both effective in changing the outcome and easily interpretable due to their simplicity. The versatility of CF-GNNExplainer is underscored by its application across different connectivity measures and datasets, highlighting its potential as a powerful tool for enhancing the transparency and understandability of GNN predictions.

In this study, we explored the application of CF-GNNExplainer and compared it against the 1HOP baseline using the Facebook social circles dataset. Our focus was on evaluating the models' performance across different connectivity measures—degree centrality, clustering coefficient, and eigenvector centrality—using fidelity, sparsity, and accuracy as metrics.

Our results reveal that CF-GNNExplainer exhibits a versatile performance, showing higher fidelity in Circle 2 and superior sparsity across both circles. This suggests that CF-GNNExplainer is adept at generating concise and potentially more understandable counterfactual explanations. However, the variability in performance across different metrics and circles underscores the complexity of the ground-truth data and the nuanced nature of counterfactual explanation generation.

One limitation of our study is the absence of real-world node classification datasets with explainability ground truth, limiting our evaluation to synthetic and social network datasets. This gap highlights an opportunity for future research to develop datasets that can further validate and refine counterfactual explanation methods like CF-GNNExplainer.

In conclusion, while our analysis supports the potential of CF-GNNExplainer in providing insightful counterfactual explanations, it also points to the need for more comprehensive evaluations. Future work could explore the integration of additional connectivity measures, the development of real-world datasets for node classification, and the refinement of counterfactual explanation generation methods to enhance their interpretability and applicability.

References

[1] Robert Anderson. The rashomon effect and communication. *Canadian Journal of Communication*, 41(2):249–270, 2016.

[2] Sergey Brin. The pagerank citation ranking: bringing order to the web. *Proceedings of ASIS, 1998*, 98:161–172, 1998.

[3] Alexandre Duval and Fragkiskos D Malliaros. Graphsvx: Shapley value explanations for graph neural networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*, pages 302–318. Springer, 2021.

[4] Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. *Advances in neural information processing systems*, 25, 2012.

[5] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks, 2019.