Important Counterfactual Paper Summaries and Takeaways:

You

October 12, 2023

1 Meeting notes

1.1 New Ideas:

1.1.1 Generative Approaches to Global CFs:

1. I feel if we could use probabilistic and generative approaches for combining the info of multiple instances with same prediction and in some way, generate CFs for them using some generative scenario with the counterfactual objective, it may be a good way to generate Global CFs

1.1.2 Possibility of Improvement in GCFE:

1. Currently, GCFE explores every possible edge deletion/addition from an input graph: We can try to minimize the search space by first using a factual explainer to identify the factual nodes and edges, and only explore changes to these nodes and edges. This may help in faster convergence (just a thought experiment)

1.1.3 Most Common/Desired Characteristics of Counterfactual Explanations (or Global ones):

1. Preferably close to the original instance (for global, close to the original instance, on average)
2. Should ideally be generated without access to the training data or the inner workings of the models that they explain (model-agnostic is a very strong requirement); only the input and the output to the model for a set of instances should be sufficient to generate the explanations
3. Specially for Global CFs: less no of them should be able to explain most (all) of the input instances in a set of instances
4. They should ideally be model-agnostic and should not reveal any information about the underlying model and data that is not accessible through the model's predictions.
5. For global explanations that aim to explain a group of similar input instances (and of course with similar predictions); there will always be a trade-off b/w how global the explanation is it and how minimal (on average) are the changes required to flip the prediction of any graph in that group. If we have few global explanations for all examples, we will need to sacrifice minimality of change to some extent, and if we have large no of global CFs, we can ensure the minimality of changes to a reasonable extent.

1.2 Broad Summary of Some Important Counterfactual Papers and Ideas that can be taken from them:

1.2.1 Robust Counterfactual Explainer:

1. **Problem Formulation:**
    - Focus on **GNNs** that adopt piecewise linear activation functions, such as **MaxOut** and the family of **ReLU**.

- **Robust Counterfactual Explanation Problem:** Given a GNN model ϕ trained on a set of graphs D, for an input graph $G = \{V, E\}$, the goal is to explain why G is predicted by the GNN model as $\phi(G)$ by identifying a small subset of edges $S \subseteq E$, such that (1) removing the set of edges in S from G that causes the maximum drop in the confidence of the original prediction; and (2) S is stable and doesn't change when the edges and the feature representations of the nodes of G are perturbed by random noise.

2. Method:

- **Modelling and Extracting Decision Regions:**
  - Extracting common decision logic of GNNs on a large set of graphs with same class prediction.
  - ϕ_{gc}: the mapping function of GCN that maps an input graph G to its graph embedding $\phi_{gc}(G) \in O^d$
  - ϕ_{fc}: the mapping function (realized by the fully connected layers) that maps the graph embedding $\phi_{gc}(G)$ to a predicted distribution over the classes in C. Thus, $\phi(G) = \phi_{fc}(\phi_{gc}(G))$.
  - $\mathcal{H}$: The set of **Linear Decision Boundaries** induced by ϕ_{fc}. The set of LDBs in $\mathcal{H}$ partitions the space O^d into a large number of convex polytopes.
  - A single convex polytope is formed by a **subset of LDBs** in $\mathcal{H}$. All the graphs whose graph embeddings are contained in the same convex polytope are predicted as the **same class**
  - Therefore, the LDBs of a convex polytope encode the common decision logic of ϕ_{fc} on all the graphs whose graph embeddings lie within the convex polytope. Here, a graph G is **covered** by a convex polytope if the graph embedding $\phi_{gc}(G)$ is contained in the convex polytope.
  - The key idea is to find a convex polytope covering a large set of graph instances in D that are predicted as the same class $c \in C$.
  - Definitions:
    (a) $D_c \subseteq D$: the set of graphs in D predicted as a class $c \in C$
    (b) $\mathcal{P} \subseteq \mathcal{H}$: a set of LDBs that partition the space O^d into a set of convex polytopes
    (c) $r(\mathcal{P}, c)$: the convex polytope induced by $\mathcal{P}$ that covers the largest number of graphs in D_c
    (d) $g(\mathcal{P}, c)$: the number of graphs in D_c covered by $r(\mathcal{P}, c)$
    (e) $h(\mathcal{P}, c)$: the number of graphs in D that are covered by $r(\mathcal{P}, c)$ but are not predicted as class c
  - Extract a decision region covering a large set of graph instances in D_c by solving the following constrained optimization problem.

$$\max_{\mathcal{P} \subseteq \mathcal{H}} g(\mathcal{P}, c), \text{ s.t. } h(\mathcal{P}, c) = 0 \tag{1}$$

  - This formulation realizes the two properties of decision regions because $\mathcal{P} \subseteq \mathcal{H}$ ensures that the decision region is induced by a subset of LDBs in $\mathcal{H}$, maximizing $g(\mathcal{P}, c)$ requires that $r(\mathcal{P}, c)$ covers a large number of graphs in D_c, and the constraint $h(\mathcal{P}, c) = 0$ ensures that all the graphs covered by $r(\mathcal{P}, c)$ are predicted as the same class c.
  - The optimization problem in Equation (1) is intractable for standard GNNs, mainly because it is impractical to compute $\mathcal{H}$, all the LDBs of a GNN. The number of LDBs in $\mathcal{H}$ of a GNN is exponential with respect to the number of neurons in the worst-case.
  - **Key Takeaways for our Problem:**
    (a) In the context of Global CF, the coverage of a CF can be defined based on the Convex Polytope that it explains.
    (b) We can even try to group a set of nearby convex polytopes having same prediction to get a larger set of similar points if the individual polytopes do not have a large number of points.

(c) As suggested by Peyman, if some convex polytopes have very few graphs, then they could be outliers in the dataset, and we may refrain from outputting global CFs for such graphs.

(d) Instead of taking only a subset of LDBs from $\mathcal{H}$, we can try to determine the entire set of LDBs of $\mathcal{H}$, that are obtained based on traning data, and train a GVAE to produce a counterfactual for each such convex polytope (i.e. all graphs in it).

(e) Once such a GVAE has been trained, the decision logic based on all the LDBs is in someway distilled into the GVAE, so for a new test data point, we need not again find the convex polytope to which it belongs.

(f) This way, finding the entire set of convex polytopes for the trained GNN, is just a one-time process, even if it may be computationally complex.

(g) Issue with sampling only subset of LDBs and determining the convex polytopes based on them, is that since it contains only a subset of the LDBs, some of the polytopes formed by them may contain graphs of different class predictions within the same polytope. So, then all the points in that convex polytope may no longer be similar and may no longer be predicted the same class.

- **Generating Counterfactual Explanations:**
  - **Key idea:** to use the LDBs of decision regions to train a neural network that produces a robust counterfactual explanation as a small subset of edges of an input graph.
  - f_θ: the neural network to generate a subset of edges of an input graph G as the robust counterfactual explanation on the prediction $\phi(G)$. θ represents the set of parameters of the neural network.
  - The neural network f_θ takes $\mathbf{z}_i$ and $\mathbf{z}_j$ (embeddings of the connected vertices v_i and v_j as the input and outputs the probability for the edge between v_i and v_j to be part of the explanation. This can be written as

$$\mathbf{M}_{ij} = f_\theta(\mathbf{z}_i, \mathbf{z}_j), \tag{2}$$

  - $\mathbf{M}$ is an n-by-n matrix that carries the complete information to generate a robust counterfactual explanation as a subset of edges, denoted by $S \subseteq E$, where S is obtained by selecting the edges in E whose corresponding entries in $\mathbf{M}$ greater than 0.5.
  - **Training f_θ:**
    (a) Two proxy graphs G_S (denoted by G_θ and uses M instead of A) and $G_{E\setminus S}$ (denoted by G'_θ, and uses

$$\mathbf{M}'_{ij} = \begin{cases} 1 - \mathbf{M}_{ij} & \text{if } \mathbf{A}_{ij} = 1 \\ 0 & \text{if } \mathbf{A}_{ij} = 0 \end{cases} \tag{3}$$

)

    (b) **Loss function:**

$$\mathcal{L}(\theta) = \sum_{G \in D} \left\{ \lambda \mathcal{L}_{same}(\theta, G) + (1-\lambda)\mathcal{L}_{opp}(\theta, G) + \beta \mathcal{R}_{sparse}(\theta, G) + \mu \mathcal{R}_{discrete}(\theta, G) \right\}, \tag{4}$$

References