Global Counterfactual Goals and Directions

You

August 16, 2023

1 Meeting notes

1.1 Next Meeting: 25th July, 2023

1.2 Meeting 5: 18th July, 2023

1.2.1 Meeting Notes:

1. **Correct approach for creating Embedding (Representation) of the Explanation Subgraphs:**

   (a) The process of generating the factual explanation subgraph using some existing framework like CFF or RC-Explainer is dependent on the model used in the framework (For example: GCN in the CFF framework), and hence the explanation subgraph produced may vary on using different frameworks.

   (b) Since our aim is to cluster the explanation subgraphs produced by a framework across all instances of a class, and we need to represent these subgraphs into a vector embedding, it is better to use the same GCN that is used for predictions on the input instances, since the explanation subgraphs are produced in the embedding space of the input graphs as represented by the GCN, it is better to obtain their representation using the same GCN for clustering, instead of using some standard Graph Embeddings like Graph2vec, etc.

   (c) Moreover, methods like Graph2Vec or Node2Vec are quite old and there are better GNNs like GAT, GIN or GraphSage for getting graph/subgraph representations.

   Key Takeaway: Use the same GCN for creating the embeddings of the expln subgraphs for clustering

2. To be able to get more insights on this approach of clustering the explanation subgraphs into groups to find similarity, also experiment the same with other factual explainers like RCExplainer(its results are quite stable and is a very good Factual and CF explainer) as well as some other Factual explainers.

3. A third experiment could be to extract the explanation subgraphs from the above methods (CFF or RC Explainer) and find the prediction of the model on just these subgraphs wrt how accurate are the predictions on these subgraphs: whether they have the same prediction as the entire graph most of the times, as then only they are good factual explanations.

4. **Finding each cluster's representative after clustering:** Jay suggested a great approach for finding the representative of each cluster once the clustering is done: **First, we find the mean of the embeddings of all the explanation subgraphs that belong to a cluster, and then find the subgraph embedding in that cluster closest to the mean embedding using some distance metric (L2 or L1 metric). This subgraph will be the representative of that cluster : it is kind of one of the prototype explanations for that particular class**.

5. A **Block diagram representation** of the entire process (**Clustering of explanation subgraphs with the end goal of generating global CF explanations**) we are trying to develop will really help to organize and figure out what parts are ambiguous or may be very complicated,

and need to be well-defined or simplified.
Link to Notebook containing the Block Diagram: https://iitkgpacin-my.sharepoint.com/:o:/g/personal/pranavnyati26_kgpian_iitkgp_ac_in/Es91PVpm9RBHtSV6F6360bUBYpOIkuo8LhiIodUFPFW?e=MMAIsP

6. Another interesting approach could be to first cluster all the factual subgraphs of all the input graph instances -¿ this will give a mapping of which graph instance belongs to which cluster. Now similar to the CFF, learn a common mask across all these graphs using Prof of Sufficiency and Necessity constructs.

1.3 Meeting 4: 27th June, 2023

1.3.1 Meeting Notes:

1. We need to tie our main objective of a global counterfactual explanation to this process of identifying explanatory (factual) subgraphs in all of the input graph instances, followed by clustering these subgraphs into groups based on the similarity of the explanation subgraphs. Moreover, extract the induced factual explanation subgraph from each graph rather than just the explanation subgraph identified by the model.

2. Think of a Clustering Objective with a Loss term that helps generate the CF. The clustering could be K-means or any other clustering method, such as differentiable versions of K-means. We can also use the GREED framework to find the GED between each pair of subgraphs and then cluster based on that.

3. We can also try to find a matching between a Factual cluster of class 0 to a factual cluster of class 1 (a matching such that the edit distance between the two clusters is minimum), though this may still not satisfy the minimality condition of a counterfactual explanation.

4. Another direction could be to explore whether the counterfactual explanation for a complete graph instance by some framework is related (or similar) to the counterfactual generated from the explanation subgraph (factual explanation subgraph) of that graph instance using the same framework.

1.4 Current experiments:

1. Extracting the induced explanation subgraphs from each of the graphs and then using Node2Vec to get a representation for these explanation subgraphs -¿ followed by K-means or a variant of it (that can determine the no clusters in the data by itself) to cluster these factual explanation subgraphs.

1.5 Meeting 3: 8th June, 2023

1.5.1 Meeting Notes:

1. **Exploring direction 4**: We discussed proceeding with direction 4 and identifying top-K most important prototypes (subgraphs/sub-structures) among all graphs of a particular class, where the prototypes are important in the sense that they are the main discriminative subsection contributing to the graph belonging in a particular class. There are several ways to approach this problem:

   (a) Using a local (instance-specific) GNN-explainer to identify important subgraphs of each graph of a particular class in the dataset and then grouping these subgraphs into clusters based on some similarity metrics; The end goal is to be able to tweak these subgraphs within the original graph to generate global counterfactuals. [Sourav: Formalize this as a three step process more concretely. All three steps could be algorithmic.]

   A Three-Step Procedure for Global Counterfactual Explanation:

i. First, for a graph-dataset, generate the local (instance-specific) Factual Explanations using some local Factual GNN-Explainer (For example, CFF) for each of the input graphs belonging to one class. Such a Factual Explainer will output for each input graph a set of one or more subgraphs that are most discriminative in the sense that their presence contributes most to the class membership of that graph.

ii. In the next step, we will cluster the identified subgraphs of all the graphs belonging to a class into certain groups based on similarity of the subgraphs (using some similarity metric like GED). After clustering the subgraphs into groups, we need to create a mapping of the input graphs of the class to the subgraph cluster to which they belong, i.e., each subgraph cluster can map to (possibly) multiple input graphs if one of the subgraph in that cluster was identified as most discriminative for that input graph.
In the process of the mapping, an input graph may get mapped to more than one graph cluters (Can resolve this based on some Criterion).
After creating the mapping, for each subgraph cluster, we identify the minimal common subgraph within the subgraphs of each cluster. This is required as we will simultaneously make modifications in this minimal subgraph across all input graphs belonging to that cluster to find a GLOBAL CF for the graphs mapped to that cluster. Thus, creating this minimal subgraph will allow uniformity in making modification in the input graphs.

iii. In the second step, for each subgraph cluster, we have identified the minimal subgraph common in all the subgraphs in that cluster and also the input graphs mapping to that subgraph cluster.
In the Third step, we first identify the region in each input graph (mapped to that cluster) where that minimal subgraph is present (**This step may require some new/existing neural framework that approximates the solution to subgraph isomprphism problem, though currently I am not entirely sure how this would be implemented**). Now we simultaneously do an EDIT withing the subgraph (An EDIT within the subgraph could be a node removal/edge removal/label change within that subgraph; here we may remove an edge/node that is completely within that subgraph and not cutting across it for uniformity in modification across all input graphs mapped to this cluster) in all the input graphs and find the mean change in prediction (say from class 0 to class 1) across all input graphs (belonging to this cluster). We do this for every possible edge/node removal and do the modification which has maximum average change in prediction towards the desired class. We greedily take that particular modification, and then repeat the same on the remaining minimal subgraph , until all (or greater than threshold number of) modified input graphs get the desired class membership.
Intuition behind the Greedy Approach: We need a Global Counterfactual for the set of input graphs belonging to a subgraph cluster, but with minimum modifications to the input graph, hence the Greedy approach.

This approach will give one Global CF explanation per subgraph cluster identified in the first step.

(b) We can also try to first cluster the original graphs into some groups based on the graph embeddings of the graphs and then we can try to identify the common important subgraph in each group and try to tweak that to form a global CF for that group.

(c) A third approach is to find the top-k most frequent subgraphs in the particular class and see if they are also the most important subgraphs in that class; this may simplify the problem and we can use some algorithmic approaches to solve this part (Analysing whether the most frequent graphs in a class are also the most discriminative ones)? [Sourav: Define the concrete experiments]

2. **Understanding the code of GCFExplainer**: We identified certain issues with the trained GNN of the GCFE, so we are trying to clarify them and understand the code in detail.

1.6 Catch up - Jay and Pranav: 6th June, 23

1. Spent time on going through the Psuedo code.
2. Connecting the Psuedo code with various function in the code.
3. found some irregularities with the prediction function (GNN model prediction) - Have to experiment more to confirm.
4. Need to schedule more time to decipher the code.
5. TODO: To seperately compare: GNN Model's predictions for the dataset with the stored predictions

1.7 Meeting 2: 31st May, 2023

1.7.1 Link to the OneNote notebook:

The notebook contains some limitations in the current work and some new approaches:
https://iitkgpacin-my.sharepoint.com/:o:/g/personal/pranavnyati26_kgpian_iitkgp_ac_in/Es91PVpm9RBHtSV6F6360bUBYpOIkuo8LhiIodUFPFWUtA?e=MMAIsP

1.7.2 Meeting Notes:

The following broad approaches were discussed during the meet for improving the existing GCFE framework/exploring new frameworks:

1. **Direction 1**: Generating a CF by multiple sequential edits: Thinking and coming up with certain properties and assumptions that the CF generation process must follow in each step till it arrives at a counterfactual. For example: If the CF is generated by editing the original graph 1 step at a time, we choose such an edit in each step which decreases the probability of the current class after that edit (or say decreases it maximally). Thus, incrementally decrease the probability of the current class in each step:
   Decide on the extent to which we want the problem to be sequential or combinatorial.

   Reason for the approach:
   (a) Suppose we have a budget of k-edits max per input-graph, then for some graphs if we cannot reach a CF in k-edits as they may be well off the boundary inside class 0. Still for these graphs we can give some insight on how close are we to a counterfactual after k-edits using above sequential approach
   (b) Moreover, we can easily ensure that the domain-constraints are validated at each sequential step (Say in chemical datasets, removing an edge/atom upsets the valency balance, so when we remove an edge in a step , we need to add another edge somewhere else to balance the valencies. This can easily be done in a sequential approach.

   Advantage: May make the RW process more guided and converge quickly
2. **Direction 2**: Better RW approach with some better theoretical guarantees or better CF quality, convergence time , etc.
3. **Direction 3:** Another more general and extreme approach is to Learn everything in the framework in a data/domain agnostic sense: For example the same approach may work for a molecualar to a social network dataset, etc.

   Advantage: More general framework and can be employed for different types of domain
   Disadvantage: May involve a lot of parameters and training and inference may become slow.
4. **Direction 4**: Can we devise a learning framework that identifies the top-K important subgraphs (found in one or more data points that have max contribution in making those graphs belong to class 0) from all of the N input graphs, and group the input graphs according to which of these k subgraphs is found in them? Then if we can sequentially tweek these subparts within the graphs to get the prediction to class 1, we may be able to learn effective CFs with a very limited Random Walk

1.7.3 Plan for this and the next Week

1. Complete the understanding of the major parts of the code
2. Trying to come up with some framework (algorithmic or learning-based) for the **Direction 4** stated above

1.8 Week 1: 7th May, 2023

1.8.1 Weekly Tasks

1. (8-15th May): Reproduce the GCE paper
2. check the application (can we add domain constraints)
3. write down running thoughts

1.8.2 High level goals

1. Theoretical guarantees for the Random Walk or Better Approaches/Alternatives to Random Walk than VRRW
2. Rather than a two-step solution (first pruning the search space to get good CFs and selecting k best among them in the 2nd step), if we can have a 1-step approach - either Learning based or Algorithmic
3. Generalizing to Multi-Lable Classification problem of graphs, i.e., generating multiple global CFs for graphs belonging to one predicted class, with each counterfactual explanation corresponding to one of the other classes.

References