Learning from Counterfactual Links for Link Prediction

Aditya Choudhary and Sooraj A P

Course Project : CS768

September 14, 2023

Introduction

Widely used methods for link prediction utilise node representation learning and use that to learn associations and further predict existence of links. However this approach overlooks the causal relationship between links and nodes.

This paper attempts to incorporate the causal relationship between nodes and links in the learning procedure to identify underlying factors that dictate these relations and thereby improve link-prediction models.

Causal Relationship - An Example

Take a specific social network as an example. Suppose Alice and Adam live in the same neighborhood and they are close friends. The association between neighborhood belonging and friendship could be too strong to discover the essential factors of friendship such as common interests or family relationships. Such factors could also be the cause of them living in the same neighborhood

Association based methods may fail to identify these underlying factors.

Causal Relationship - An Example

The approach of the paper is to ask the counter-factual question *Would Alice and Adam still be close friends if they were not living in the same neighborhood?*.

This helps us to investigate the causal relation between being friends and being in the same neighborhood.

Causal Inference

A more general approach would be to ask the question *Would the link exist or not if the graph structure became different from observation ?*

A counterfactual question is usually framed with three factors: context (as a data point), manipulation (e.g., treatment, intervention, action, strategy), and outcome. In our example we use the node representations of Adam and Alice as context, whether they have the same neighborhood as treatment and their friendship (existence of link) as outcome.

Given a pair of nodes and their treatment we look for another pair of nodes that is most similar to our pair but has the opposite treatment. This is a counter-factual link.

Treatment

Notation

Let $\mathbf{A}$ denote the adjacency matrix, $\mathbf{A^{CF}}$ denote adjacency matrix of counter-factual links, $\mathbf{T}$ denote treatment matrix and $\mathbf{T^{CF}_{i,j}} = 1 - \mathbf{T_{i,j}}$.

The paper uses the global structural role of each node pair as its treatment. The causal model does not limit the treatment to be structural roles i.e $T_{i,j}$ can be any binary property of node pair (v_i, v_j). The paper proceeds with Louvain an unsupervised approach used for community detection.
The treatment is then taken to be whether or not a pair of node belong to same community or note. Formally,

$$\mathbf{T_{i,j}} = 1 \text{ if } c(v_i) = c(v_j), \text{ and } \mathbf{T_{i,j}} = 0 \text{ otherwise}$$

where c is any clustering method

Causal Model

Notation

Z denotes the context, T $\in \{0, 1\}$ denotes treatment and Y denotes outcome.

The paper uses Individual Treatment Effect (ITE) defined as difference of outcome given context with opposite treatments. Formally,

$$ITE(z) = g(z, 1) - g(z, 0)$$

$$ITE_{(v_i, v_j)} = g((z_i, z_j), 1) - g((z_i, z_j), 0)$$

where $z \in Z$ and g(z,T) is the outcome of z given treatment T.
The definition uses same context with opposite treatments however this is (practically) impossible to procure. Hence we look for pairs of nodes with similar context but opposite treatment

That is, we want to find the nearest neighbor with the opposite treatment for each observed node pairs and use the nearest neighbor's outcome as a counterfactual link.
Formally, $\forall(v_i, v_j) \in V \times V$, we define its counterfactual link (v_a, v_b) as

$$(v_a, v_b) = \underset{v_a, v_b \in V}{\arg\min}\{d(\widetilde{x}_i, \widetilde{x}_a) + d(\widetilde{x}_j, \widetilde{x}_b) \mid T_{a,b} = 1 - T_{i,j}, d(\widetilde{x}_i, \widetilde{x}_a) + d(\widetilde{x}_j, \widetilde{x}_b) < 2\gamma\} \tag{1}$$

where d(·, ·) is specified as the Euclidean distance on the embedding space of X, and γ is a hyperparameter that defines the maximum distance that two nodes are considered as similar

Learning

Unlike traditional link-prediction algorithms the CFLP model takes the following as inputs

- The observed graph data $\mathbf{A}$ and raw feature matrix $\mathbf{X}$
- The factual treatments $\mathbf{T}$ and counterfactual treatments $\mathbf{T}^{\mathbf{CF}}$
- The counterfactual links data $\mathbf{A}^{\mathbf{CF}}$

and outputs the link-prediction logits for factual and counter-factual links.

Learning Model

The model consist of two trainable components: a graph encoder f and a link decoder g. The paper uses GCN as its graph encoder For the link decoder, the paper uses a Multi-Layer Perceptron (MLP) that acts on the context (given by Hadamard Product of node representations of (v_i, v_j)) and the treatment $T_{i,j}$. Formally,

$$\hat{A}_{i,j} = MLP([z_i \odot z_j, T_{i,j}]),$$

$$\hat{A}^{CF}_{i,j} = MLP([z_i \odot z_j, T^{CF}_{i,j}]),$$

where [.,.] is concatenation

Learning Process

During the training process, data samples from the factual distribution and the counterfactual distribution are put into decoder g and optimized towards $\mathbf{A}$ and $\mathbf{A^{CF}}$.

The loss function is the standard Binary Cross Entropy loss for both factual and counter-factual distributions given by

$$\mathcal{L}_F = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{i,j} \cdot log(\hat{A}_{i,j}) + (1 - A_{i,j}) \cdot log(1 - \hat{A}_{i,j})$$

$$\mathcal{L}_{CF} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} A^{CF}_{i,j} \cdot log(\hat{A}^{CF}_{i,j}) + (1 - A^{CF}_{i,j}) \cdot log(1 - \hat{A}^{CF}_{i,j})$$

Learning Process

During the training process, data samples from the factual distribution and the counterfactual distribution are put into decoder g and optimized towards $\mathbf{A}$ and $\mathbf{A^{CF}}$.

The loss function is the standard Binary Cross Entropy loss for both factual and counter-factual distributions given by

$$\mathcal{L}_F = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{i,j} \cdot log(\hat{A}_{i,j}) + (1 - A_{i,j}) \cdot log(1 - \hat{A}_{i,j})$$

$$\mathcal{L}_{CF} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} A^{CF}_{i,j} \cdot log(\hat{A}^{CF}_{i,j}) + (1 - A^{CF}_{i,j}) \cdot log(1 - \hat{A}^{CF}_{i,j})$$

Learning Process

The model is trained on both factual and counter-factual distributions however it is tested on just factual distribution. This shift in input distribution has been seen as an issue in counter-factual representation learning. We need the training and test distributions to be similar, hence we force the counter-factual distribution to be similar to the factual distribution, we add regularization parameter to our loss.

$$\mathcal{L}_{disc} = \text{disc}(\hat{P}_f, \hat{P}_f^{CF})$$

where disc(P,Q) $= \|P - Q\|_F$ where $\|.\|_F$ is Frobenius norm
$\hat{P}_{f_{i,j}} = [z_i \odot z_j, T_{i,j}] \quad \hat{P}^{CF}_{f_{i,j}} = [z_i \odot z_j, T^{CF}_{i,j}]$
The overall loss function is given by,

$$\mathcal{L} = \mathcal{L}_F + \alpha \cdot \mathcal{L}_{CF} + \beta \cdot \mathcal{L}_{disc}$$

Learning Process

The discrepancy loss is computed on the representations of node pairs learned by the graph encoder f, so the decoder g is trained with data from both $\hat{P}_F$ and $\hat{P}_{CF}$ without balancing the constraints. Therefore, after the model is sufficiently trained, we freeze the graph encoder f and fine-tune g with only the factual data wrt $\mathcal{L}_F$

Plans of Further Work

- Ablation Study to study the effect of $\mathcal{L}_{disc}$ in loss function and experiment with alternatives to Frobenius Norm like Spectral Norm
- Investigate how ATE can play a role in Graph Classification problems

More Complex Treatment

We propose an alternative of evaluating the treatment variable that would allow us to move from binary values to a larger scope.

Between each pair of communities we compute a distance metric between them and linearly assign scores to each distance between 0 and the maximum distance. The treatment of (v_i, v_j) is given to be the distance score between the communities v_i and v_j belong to. Higher the treatment the closer the nodes are.

The counter-factual link (v_a, v_b) would require its treatment to be less than $1 - \mathbf{T_{i,j}}$ i.e

$$(v_a, v_b) = \underset{v_a, v_b \in V}{\arg\min} \{ d(\widetilde{x_i}, \widetilde{x_a}) + d(\widetilde{x_j}, \widetilde{x_b}) \mid T_{a,b} \leq 1 - T_{i,j}, d(\widetilde{x_i}, \widetilde{x_a}) + d(\widetilde{x_j}, \widetilde{x_b}) < 2\gamma \} \tag{2}$$

Community to Cliques : A Theoretical Improvement

Cliques being stronger than community allow us to investigate more in-depth.

We identify all the maximal cliques in the graph. Note that the vertex may belong to several maximal cliques, in this case we would assign its clique to be the maximum clique among the maximal cliques it is a part of. Now, we would have a mapping from vertices to maximal cliques instead of a mapping from vertices to communities.

Further analysis follows as for community mappings. However the problem of enumerating maximal cliques of a graph is NP Complete and hence we propose a theoretical analysis and possible implementation on graph of small size.

Possible adaptation to multiple Treatment variables

We suggest an idea to possibly adapt the ideas in the paper to accommodate more than one treatment variables.

Suppose that $T^{(1)}, T^{(2)}, \ldots, T^{(k)}$ are k distinct Boolean treatment variables for node-pairs. Define by $(\hat{T})$, a new treatment vector obtained by concatenating the $T^{(i)}$s (This is a $k-$bit $(0,1)$-vector).
We repeat the process for finding counterfactual links k times as in equation 1, with the constraints as $|ham(T_{i,j}, T_{a,b})| = i$ starting with $i = k$ and moving down to 1. Here, $ham(\cdot, \cdot)$ represents Hamming Distance.

We could mimic the same procedure but with $k+1$ different probability distributions and train over the $k+1$ adjacency matrices obtained from the decoder g.

We are however unsure whether or not this is feasible.