[Arxiv 2020] Self-supervised Learning on Graphs: Deep Insights and New Direction. [paper] [code]

Node/Graph Tasks: Node classification

Training Type: Pre-training with fine-tuning and joint training
The purpose of self-supervised learning is to guide the GNN-based model to extract features that retain feature or topological information of the original graph. Based on the information that is to be retained, the pretext tasks are divided into follows.
Local structure information
Node property (specifically node degree here): graph topology, node features
The pretext task here is to train our GNN model to extract features which encode the information of the node degrees(could be any other node local property):

$$\mathcal{L} = \frac{1}{|\mathcal{D}_u|} \sum_{v \in \mathcal{D}_u} (f_{\theta'}(\mathcal{G})_v - d_v)^2 \tag{36}$$

where θ' denotes the parameters of a GNN model $f_{\theta'}$, $\mathcal{D}_u$ is the set of unlabeled nodes. $f_{\theta'}(\mathcal{G})_v$ is the predicted local node degree (could be any other local node property) of node v.
Edge existence: graph topology, node features
The pretext task here is to train our GNN model to extract features which encode the information of the masked graph edges. Assuming that adjacency nodes have closer GNN embeddings, the edge recovery problem can be reformalized as a binary classification problem:

$$\mathcal{L} = \frac{1}{|\mathcal{M}_e|} \sum_{(v_i,v_j) \in \mathcal{M}_e} l(f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|), 1) \tag{37}$$

$$+ \frac{1}{|\mathcal{M}_u|} \sum_{(v_i,v_j) \in \mathcal{M}_u} l(f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|), 0) \tag{38}$$

where l is the cross entropy loss, f_w linearly maps the distance between two node embeddings to 1-dimension, and $\mathcal{M}_e, \mathcal{M}_u$ represent node pairs with/without edges. Note that $\mathcal{G}$ here is the modified graph by randomly masking edges.
Global structure information
Pairwise distance: graph topology, node features
The pretext task is to train our GNN model to extract features which encode the information of distance between nodes. The distance between nodes is measured by the shortest path distance and is further grouped into multiple categories. By randomly sampling a certain amount of node pairs $\mathcal{S}$ and recovering the distance between nodes can be formulated as a multi-classification problem with the cross entropy loss:

$$\mathcal{L} = \frac{1}{|\mathcal{S}|} \sum_{(u,v) \in \mathcal{S}} l(f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|), C_{p_{ij}}) \tag{39}$$

where $C_{p_{ij}}$ is the corresponding distance category of the shortest path p_{ij} between node i, j. Note that the pairwise shortest distance could be replaced by other distance measures.

Distance to clusters: graph topology, node features Since calculating the pairwise shortest path distance for all node pairs even after the sampling is time-consuming, distance to predefined graph clusters can be utilized instead. A fixed set of anchor/center nodes associated with graph clusters is established and for each node, we calculate its distance to this set of of anchor nodes associated with graph clusters. The pretext task is to extract node features that encode the information of this node2cluster distance. The clusters $\{C_1, C_2, ..., C_k\}$ are obtained by applying the METIS graph partitioning algorithm. Suppose that we select the node with the highest degree to be the center of the corresponding cluster C_j, denoted as c_j, then each node v_i will have a cluster distance vector $\mathbf{d}_i \in \mathbb{R}^k$ and the distance2cluster pretext task is completed by optimizing:

$$\mathcal{L} = \frac{1}{|\mathcal{D}_u|} \sum_{v_i \in \mathcal{D}_u} ||f_{\theta'}(\mathcal{G})_{v_i} - \mathbf{d}_i||^2 \tag{40}$$

Attribute information

Node attribute: graph topology, node features The pretext task here is to train our GNN model to extract features which encode the information of node features that have been masked. Similarly to edge existance pretext task, we first randomly mask features of nodes in the set $\mathcal{M}_a$, and then ask the self-supversied component to reconstruct these features by:

$$\mathcal{L} = \frac{1}{|\mathcal{M}_a|} \sum_{v \in \mathcal{M}_a} ||f_{\theta'}(\mathcal{G})_{v_i} - \mathbf{x}_i||^2, \tag{41}$$

where $\mathbf{x}_i$ is the original unmasked features of node i. To handle the complxity of the node attributes, PCA is applied first to reduce the dimension.

Pairwise Attribute Similarity: graph topology, node features The pretext task here is to train our GNN model to extract features which encode the information of the pairwise similarity of nodes. Suppose $\mathcal{T}_s, \mathcal{T}_d$ denote the sets of node pairs having the highest similarity and dissimilarity, which is formally defined as:

$$\mathcal{T}_s = \{(v_i, v_j) | s_{ij} \text{ in top-K of } \{s_{ik}\}_{k=1}^{K} \backslash s_{ii}, \forall v_i \in \mathcal{V}_u\} \tag{42}$$

$$\mathcal{T}_d = \{(v_i, v_j) | s_{ij} \text{ in bottom-K of } \{s_{ik}\}_{k=1}^{K} \backslash s_{ii}, \forall v_i \in \mathcal{V}_u\} \tag{43}$$

where s_{ij} measures the cosine similarity of features between v_i, v_j and K is the number of top/bottom pairs selected for each node. The pretext task is realized by optimizing the following regression loss:

$$\mathcal{L} = \frac{1}{|\mathcal{T}_s \cup \mathcal{T}_d|} \sum_{(v_i, v_j) \in \mathcal{T}_s \cup \mathcal{T}_d} ||f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|) - s_{ij}||^2 \tag{44}$$

Distance to labeled training data: graph topology, node features, training node labels The pretext task here is to train our GNN model to extract features which encode the information of the topological distance between nodes to training nodes. For class $c_j \in \{1,...,K\}$ and unlabeled node $v_i \in \mathcal{V}_u$, the average, minimum and maximum shortest path length from v_i to all labeled nodes in class c_i is calculated and the distance vector of v_i is $\mathbf{d}_i \in \mathbb{R}^{3K}$, then the objective is to optimize the following regression loss:

$$\mathcal{L} = \frac{1}{v_i \in |\mathcal{D}_u|} ||f_{\theta'}(\mathcal{G})_{v_i} - \mathbf{d}_i||^2 \tag{45}$$

Context label: graph topology, node features, training node labels The pretext task here is to train our GNN model to extract features that encode the information of the context of nodes. The context is defined as:

$$\mathbf{y}_{ic} = \frac{|\mathcal{N}_{\mathcal{V}_l}(v_i,c)| + |\mathcal{N}_{\mathcal{V}_u}(v_i,c)|}{|\mathcal{N}_{\mathcal{V}_l}(v_i)| + |\mathcal{N}_{\mathcal{V}_u}(v_i)|}, c = 1,...,K, \tag{46}$$

where $\mathcal{N}_{\mathcal{V}_u}(v_i)$ denotes the neighborhood set from $\mathcal{V}_u$ of node v_i, $N_{\mathcal{V}_u}(v_i,c)$ denotes nodes that have been assigned class c among the neighborhood set. Label propagation (LP) or the iterative classification algorithm (ICA) could be used to construct psudo labels for unlabeled nodes in $\mathcal{V}_u$. Then the pretext task is approached by optimizing the following loss function:

$$\mathcal{L} = \frac{1}{|\mathcal{D}_u|} \sum_{v_i \in \mathcal{D}_u} ||f_{\theta'}(\mathcal{G})_{v_i} - \mathbf{y}_i||^2 \tag{47}$$

Since much noise exist in the psudo-labels produced by LP or ICA, the paper further proposes two improvements on this pretext task.

First we can combine the psudo labels produced by LP and ICA to reduce the label noise by:

$$\text{label}_i = \text{argmax}_c \sigma_{\text{LP}(v_i)} + \sigma_{\text{ICA}}(v_i), c = 1,...,K \tag{48}$$

After get this combined label for each node, we can construct the context as above and resolve the pretext task.

By performing node classification on citation networks, several discoveries are disclosed by this work. First, joint training outperforms two state training in most settings. The best performance is obtained by self-supervised pretext tasks from global structure information. The pretexts tasks of encoding node degrees, edge existence and node attributes fail to boost the original GCN. The pretext tasks of encoding pairwise distance, distance2cluster and pairwise similarity successfully improve the performance.

Initial short summary here

The content is pretty much included in the pretest task part.

Bibtex:
@articlejin2020self, title=Self-supervised learning on graphs: Deep insights and new direction, author=Jin, Wei and Derr, Tyler and Liu, Haochen and Wang, Yiqi and Wang, Suhang and Liu, Zitao and Tang, Jiliang, journal=arXiv preprint arXiv:2006.10141, year=2020