[Openreview 2020] Self-supervised Graph-level Representation Learning with Local and Global Structure [paper]

Node/Graph Tasks: Graph classification

Training Type: Pretraining GNN to obtain graph embeddings and contrastive learning by maximizing the mutual information to further optimize the GNN encoder and construct the hirarchical prototypes

Pretext task data: graph topology, node features and graph labels
The pretext task is to first construct the hierarchical prototypes by rival penalized competitive learning (RPCL) and then optimize the GNN-based encoder by contrastive learning of local patch representations, local graph representations and the global prototypes representations.

Initial short summary here
Viewing that recent self-supervised graph representation learning methods are capable of modeling only the local structure between different graph instances but fail to discover the global-semantic structure, this paper proposes a Local-instance and GLobal-semantic Learning (GraphLoG) to model both the local and global structure of a given set of graphs. In this paper, the patch representations for node v through any GNN-based encoder is $\mathbf{h}_v$, the graph embedding for graph G through any permutation-invariant readout function is $\mathbf{h}_G = \text{READOUT}(\{\mathbf{h}_v | v \in \mathcal{V}\})$, the total number of graphs is N and the mutual information optimization loss function is:

$$\mathcal{L}_{\text{NCE}}(\mathbf{q}, \mathbf{z}_+, \{\mathbf{z}_i\}_{i=1}^{K}) = -\log \frac{\exp(T(\mathbf{q}, \mathbf{z}_+))}{\exp(T(\mathbf{q}, \mathbf{z}_+)) + \sum_{i=1}^{K} \exp(T(\mathbf{q}, \mathbf{z}_i))}. \tag{21}$$

where $\mathbf{q}$ is any query, its positive sample is $\mathbf{z}_+$ and the set of negative samples is $\{\mathbf{z}_i\}_{i=1}^{K}$. T is a parameterized discriminator function which measures the similarity between two representation vectors.

To preserve the local similarity between various graph instances, the embeddings of correlated graphs $\mathbf{h}_G$/patches $\mathbf{h}_v$ are aligned by maximizing their mutual information through minimizing:

$$\mathcal{L}_{\text{patch}} = \frac{1}{\sum_{j=1}^{N} |\mathcal{V}'_j|} \sum_{j=1}^{N} \sum_{v' \in \mathcal{V}'_j} \sum_{v \in \mathcal{V}_j} \mathbb{1}_{v \leftrightarrow v'} \mathcal{L}_{\text{NCE}}(\mathbf{h}_{v'}, \mathbf{h}_v, \{\mathbf{h}_{\tilde{v}} | \tilde{v} \in \mathcal{V}_j, \tilde{v} \neq v\}) \tag{22}$$

$$\mathcal{L}_{\text{graph}} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}_{\text{NCE}}(\mathbf{h}_{G'_j}, \mathbf{h}_{G_j}, \{\mathbf{h}_{G_k} | 1 \leq k \leq N, k \neq j\}) \tag{23}$$

In order to preserve the hierarchical semantic information in graph embeddings after the GNN-based encoder, the hierarchical prototypes $\{c_i^l\}_{i=1}^{M_l} (l = 1, 2, ..., L_p)$ are initialized and maintained during training where L_p denotes the depth of hierarchical prototypes, and M_l is the number of prototypes at the l-th layer. In initialization of hierarchical prototypes, the GNN model is pre-trained by minimizing $\mathcal{L}_{\text{graph}} + \mathcal{L}_{\text{patch}}$ and utilized to extract the embeddings of all graphs $\{\mathbf{h}_{G_i}\}_{i=1}^{N_D}$ in the training set. These embeddings are used to initialize the bottom layer prototypes via RPCL-based clustering algorithm []:

$$\{\mathbf{c}_i^{L_p}\}_{i=1}^{M_{L_p}} = \text{RPCL}(\{\mathbf{h}_{G_i}\}_{i=1}^{N_D}) \tag{24}$$

The prototypes of upper layers are initialized by iteratively applying RPCL-based clustering to the prototypes of the layer below. In the training process when the embeddings of each graph is dynamically changing, the updated graph embeddings are divided into M_{L_p} groups according to their most similar bottom layer prototype, and the mean graph embeddings are computed within each group as $\{\tilde{c}_i^{L_p}\}_{i=1}^{M_{L_p}}$. These mean embeddings are employed to update bottom layer prototypes via an exponential moving average scheme:

$$\mathbf{c}_i^{L_p} = \beta \mathbf{c}_i^{L_p} + (1-\beta)\tilde{\mathbf{c}}_i^{L_p}, 1 \leq i \leq M_{L_p}, \tag{25}$$

where β is an exponential decay rate. For the prototypes of upper layers, they are updated with the mean of their child prototypes in the corresponding tree. To ensure that correlated graphs are mapped to the same set of feature clusters, the mutual information between prototypes on the searching path of G_j and its correlated graph G'_j is maximized:

$$\mathcal{L}_{\text{global}} = \frac{1}{NL_p} \sum_{N}^{j=1} \sum_{L_p}^{l=1} \mathcal{L}_{\text{NCE}}(\mathbf{h}_{G'_j}, \mathbf{s}_l(G_j), \{\mathbf{c}_i^l | 1 \leq i \leq M_l, \mathbf{c}_i^l \neq \mathbf{s}_l(G_j)\}) \tag{26}$$

In order to apply the self-supervised version of GraphLoG to downstream tasks, the paper further proposes a supervised version model, named as sup-GraphLoG, which combines plain GNN and the proposed hierarchical prototypes. The GNN is first pre-trained along with a linear classifier to classify graphs. The number of bottom layer prototypes is set as the class number of the supervised task and each bottom layer prototype is the mean embedding of all the training graphs belonging to the corresponding class. For constraining the global-semantic structure in this supervised setting, we first select a matched bottom layer prototype based on the label of graph G_j and then obtain the whole searching path $s(G_j) = \{\mathbf{s}_1(G_j), \mathbf{s}_2(G_j), ..., \mathbf{s}_{L_p}(G_j)\}$ from bottom to up. Contrastive to this positive searching, we randomly sample a negative path $s^n(G_j)$ satisfying that graph G_j does not belong to the corresponding class in the negative path and each prototype on the negative path is not the same as the prototype on the same level on the positive path. The embedding of graph G_j is more similar to the prototypes on $s(G_j)$ rather than the ones on path $s^n(G_j)$, which is refined by:

$$\mathcal{L}_{\text{global}}^{\text{sup}} = \frac{1}{NL_p} \sum_{j=1}^{N} \sum_{l=1}^{L_p} \mathcal{L}_{NCE}(\mathbf{h}_{G_j}, \mathbf{s}_l(G_j), \mathbf{s}_l^n(G_j)) \tag{27}$$

The inference is realized by selecting the class of the corresponding prototype with the highest similarity to the unlabeled graph.

Bibtex: Openreview in ICLR 2021