[ICLR 2019 workshop] Pre-Training Graph Neural Networks for Generic Structural Feature Extraction. [paper]

Node/Graph Tasks: Node, link and graph classifications

Training Type: multi-task pre-training and adaptive fine-tuning. The pre-training procedure is the same while in the fine-tuning procedure, we choose a fix-tune boundary in the middle of GNNs. The GNN blocks below this boundary are fixed while the ones above the boundary are fine-tuned.

Pretext task data:

Denoising Link Reconstruction: graph topology

The pretext task here is to first randomly removing a fraction of existing edges and asks our GNN model to take the noised graph, learn the representation and then utilized the learned representation to predict whether two nodes are connected or not as: $\tilde{A}_{u,v} = \mathcal{D}^{rec}(\mathcal{F}^{rec}(\mathcal{G}^*)[u], \mathcal{G}^*)[v])$ where $\mathcal{D}^{rec}$ is the neural tensor network pairwise decoder and $\mathcal{F}^{rec}(\mathcal{G}^*)[u]$ is the node representation of u from GNN models. The objective is:

$$\mathcal{L} = -\sum_{u,v \in \mathcal{V}} (\mathbf{A}_{u,v} \log(\tilde{A}_{u,v}) + (1 - \mathbf{A}_{u,v}) \log(1 - \tilde{A}_{u,v})) \tag{59}$$

Centrality Score Ranking: graph topology

The pretest here is to ask the GNNs to estimate the rank scores of node centrality, which includes eigencentrality, betweenness, closeness, and subgraph centrality. For a node pair (u,v) and a centrality score s, with relative order $R^s_{u,v} = (s_u > s_v)$, a decoder $\mathcal{D}^{rank}_s()$ for centrality score s estimates its rank score by $S_v = \mathcal{D}^{rank}_s(\mathcal{F}^{rank}(\mathcal{G})[v])$. The probability of estimated rank order is defined by $\tilde{R}^s_{u,v} = \frac{\exp(S_u, S_v)}{1+\exp(S_u, S_v)}$. Then the objective is to optimize $\mathcal{D}^{rank}_s$ and $\mathcal{F}^{rank}$ for each centrality score s by:

$$\mathcal{L}_{rank} = -\sum_{s} \sum_{u,v \in \mathcal{V}} (R^s_{u,v} \log \tilde{R}^s_{u,v} + (1 - R^s_{u,v}) \log(1 - \tilde{R}^s_{u,v})) \tag{60}$$

Cluster Preserving: node features

The pretext task here is to ask the GNNs to extract node feature preserving the cluster infromation of each node. Supposing that the graph is grouped into K different non-overlapping clusters $C = C_{i i=1}^{K}$ and $\mathbf{I}$ denotes the assignment vector telling which cluster C a given node v belongs to. First, we use an attention-based aggregator $\mathcal{A}$ to get a cluster representation. Then a neural tensor network decoder $\mathcal{D}^{cluster}$ destimates the similarity of node v with cluster C. Then we can formalize this as a multi-class problem, which optimizes the following objective:

$$\mathcal{L} = -\sum_{v \in \mathcal{V}} \mathbf{I}(v) \log(P(v \in \mathbf{I}(\mathbf{v}))) \tag{61}$$

Beides, this paper consider the node embeddings used for downstream tasks and pretext tasks as a linear combination of node representations from different GNNs layers so that different tasks can utilize different perspective of structural information. Also in the fine-tuning procedure, the paper proposes to choose a fix-tune boundary in the middle of GNNs. Layers beyond this boundary are tuned while below this boundary are fixed.

Another major difference between this paper and any other peer work is that node features of this work only consist of topology feature, which includes degree, core number, collective influence and the local clustering coefficient. All the above features are min-max normalized to ensure the same scale in graphs with different sizes. Furthermore, these four features are concatenated and cascaded with a nonlinear transformation $\mathbf{E}$ which composes of a linear transformation followed by a nonlinear transformation with tanh to get the embedding vector, which serves as the input to GNNs.

Initial short summary here

The most content of the methodological part can be seen in the previous part. Here we only attach some major discoveries. First pre-training outperforms the baseline models on all downstream tasks by 7.7% micro-F1 in average. All models benefit from the additional node attributes. Different pre-training tasks are beneficial to different downstream tasks. Node classification task benefits the most from the cluster preserving pre-training tasks. Link classification benefits more from the denoising link reconstruction task. Also the paper shows that the improvement caused by pre-training is substantial when training data is extremely scarce.

Bibtex:

@articlehu2019pre, title=Pre-training graph neural networks for generic structural feature extraction, author=Hu, Ziniu and Fan, Changjun and Chen, Ting and Chang, Kai-Wei and Sun, Yizhou, journal=arXiv preprint arXiv:1905.13728, year=2019