[ICLR 2019] Deep Graph Informax. [paper]

Node/Graph Tasks: Node classification

Training Type: Training the GNN encoder by contastive learning between local patch representations and global graph representations under two graph views generated by feature and topology corruption.

Pretext task data: Structure, node features The pretext task here is to optimize a GNN-based encoder so that the mutual information between positive pairs is maximized and between the negative pairs is minimized. The positive pairs come from the local patch representations and the original global graph representations. The negative pairs come from the local patch representations in the corrupted graph and the original global graph representations.

Initial short summary here
The dominant unsupervised representation learning with graph-structured data relies heavily on random walk-based objectives, which suffers from over-emphasis on proximity information rather than structural information, highly dependence on hyperparameter choice, and non-inductivity. This paper proposes an alternative unsupervised learning model, DGI(Deep Graoh InfoMax), with the objective to maximize the mutual information between representations of high-level graphs and low-level patches.

In each iteration, a negative sample is generated by corrupting the graph through shuffling node features and removing edges $(\widetilde{X}, \widetilde{A}) \sim \mathcal{T}(X, A)$. Then a GNN-based encoder f is applied to extract latent features $H = f(X, A) = \{\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_N}\}, \widetilde{H} = f(X, A) = \{\widetilde{\mathbf{h}}_1, \widetilde{\mathbf{h}}_2, ..., \widetilde{\mathbf{h}_N}\}$ (local patch representations) for each node in both of the original and the corrupted graphs. The local patch representations are further feed into an injective readout function R to get the global graph representations $\mathbf{s} = R(H)$. Then we maximize the mutual information between $\mathbf{h}_i, \mathbf{s}$ by minimizing the following loss function:

$$\mathcal{L} = \frac{1}{N+M}(\sum_{i=1}^{N} \mathbb{E}_{(X,A)}[\log D(\mathbf{h}_i, \mathbf{s})] + \sum_{j=1}^{M} \mathbb{E}_{(\widetilde{\mathbf{X}}, \widetilde{\mathbf{A}})}[\log(1 - D(\widetilde{\mathbf{h}_j, \mathbf{s}}))])) \tag{18}$$

The experiments are performed under two learning settings. In transductive learning setting, the GNN-based encoder is a one-layer GCN model and the corruption function is to shuffle node features. In inductive learning setting, the GraphSAGE-GCN is applied and the corruption function is to shuffle node features in the subgraph. The readout function is to simply average over all nodes' features $\mathcal{R}(\mathbf{H}) = \sigma(\frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i)$. The discriminator is to calculate the weighted similarity as $D(\mathbf{h}_i, \mathbf{s}) = \sigma(\mathbf{h}^\top W \mathbf{s})$.

Bibtex:

@articlevelivckovic2018deep, title=Deep graph infomax, author=Veličković, Petar and Fedus, William and Hamilton, William L and Liò, Pietro and Bengio, Yoshua and Hjelm, R Devon, journal=arXiv preprint arXiv:1809.10341, year=2018 *copy/page the bibtex here*