Supplementary Material

No Author Given

No Institute Given

Hypernym-We , w

- Task Definition
- The Flow of SemPathGNN
- Experiments

1 Task Definition

Given a heterogenous graph $G = (V, E)$, with node attribute matrices $\mathbf{M}_{A_i} \in \mathbb{R}^{|V_{A_i}| \times d_{A_i}}$. Heterogenous graph embedding is the task to learn d-dimensional node embeddings $\mathbf{h}_v \in \mathbb{R}^d$ for all $v \in \mathbf{V}$ with $d \ll |V|$ that has a capacity of capturing rich semantics and structural information involved in G.

2 The Flow of SemPathGNN

The flow of our proposed SemPathGNN model is formalized as Algorithm 1.

3 Experiments

3.1 Baselines

We exploit a series of state-of-the-art graph embedding models as baselines are compared, including traditional homogeneous graph embedding models (as opposed to GNNs), traditional heterogeneous graph embedding models, GNNs for homogeneous graphs, and GNNs for heterogeneous graphs. Following [**?**], we call them as traditional homogeneous models, traditional heterogeneous models, homogeneous GNNs, and heterogeneous GNNs, respectively. The list of these baseline models are shown as follows.

3.2 Implementation Details

We perform Adaptive Moment Estimation (Adam) [**?**] to optimize our model and apply a grid search for hyper-parameters: $\{0.05, 0.01, 0.001, 0.002, 0.005\}$ and the dropout ratio is tuned amongst $\{0.1, \cdots, 0.5\}$. To void gradients vanishing or exploding, we employ batch normalization [**?**] and set LeakyReLU [**?**] as the activation function. For random walk based models, i.e., DeepWalk and Metapath2vec, we set the moving window size to 10, the walk length to 40, and the number of walks in each node to 80. For a fair comparison, we set the embedding dimension to 64 for all models. For GNNs for homogeneous graphs or heterogeneous graphs (they also can be called GNN based semi-supervised models) as baselines, i.e., GCN, GAT, HAN, and GTN, we initialize the hyper-parameters for them by following the corresponding paper and carefully tune them to ensure that they achieve the optimal performance. We conduct all the experiments in a Linux server with 1 $\times$ Intel Xeon Gold 6130 CPU (64 Cores 2.1 GHz), 96 GB of RAM, and 1 $\times$ NVIDIA Tesla P100-PICE GPU. For the acceleration of all CPUs calculations, we have adopted a parallelized processing of binding 64 cores. Our model is implemented in Pytorch 1.6.0[1].

3.3 Hyper-parameter Sensitivity Analysis

We investigate the sensitivity of a representative hyper-parameter on dimension of the final node embedding d in the SemPathGNN model and plot the results in Figure **??**. It can be shown that, no significant change in performance when d ranges from 64 to 128, and the best results is attained as d is set to 64 for all the evaluation metrics on both node classification and clustering tasks.

4 Conclusion

This paper

References

[1] https://pytorch.org/