Supplementary Files for HetBiSyn

Yulong Li[1][0000−0002−3068−2554], Hongming Zhu[1][0000−0001−5795−5279], Xiaowen Wang[1][0000−0003−1880−7921], and Qin Liu[1][0000−0002−9352−1694]*

[1]School of Software Engineering, Tongji University, Shanghai, China

1 Detailed Derivation of HGAT

We first agree on symbols to elucidate the principle of the HGAT models more clearly. For a graph $G = (V, E)$ in this question, a meta-path π_p is defined by an edge as a composite relation connecting strictly two nodes, e.g."drug - drug_target_interaction - protein", "atom - single_bond - atom", etc. We denote the meta-path set for graph as $\pi = \{\pi_0, \pi_1, \ldots, \pi_p\}$ and the node type set as $\tau = \{\tau_0, \tau_1, \ldots, \tau_i\}$.

Each type of node may have initial features of different dimensions, so we design the type-specific auto-encoder A_τ to project the diverse representations into a unified feature space. For nodes of type τ_i, suppose the initial representation of a node goes h_i, the projected result is:

$$h_i^{'} = A_{\tau_i} \cdot h_i \tag{1}$$

In practice, the auto-encoders are applied on specific types of nodes that actually need feature transformation, such as that the dimension of drug node vectors are lifted from 27 to 128 in order to align to other type of nodes in G_{bio}, while nodes representing diseases in the same graph are unprocessed since they are initiated to be 128-d. Those unprocessed nodes are also denoted as $h^{'}$ for a unified notation as we view their auto-encoder as a fixed identity matrix. The weight of each node can then be calculated as each node fits in the same dimension, which is implemented by self attention mechanism. First, the node-level attention of a meta-path π_p is calculated as:

$$e_{ij}^{\pi_p} = N_{node}(h_i^{'}, h_j^{'}; \pi_p) = \sigma(a_{\pi_p}^T \cdot [h_i^{'} || h_j^{'}]) \tag{2}$$

where the nodes i and j is connected by π_p, N_{node} is a deep neural network performing node-level attention that is shared by all π_p based node pairs, σ is the sigmoid activation function and $||$ represents the concatenation of two vectors. A node-level attention vector $a_{\pi_p}^T$ is devised to implement the DNN. As the importance between meta-path based node pairs is obtained, we may calculate the weight coefficient by normalizing them using softmax:

$$\alpha_{ij}^{\pi_p} = softmax_j(e_{ij}^{\pi_p}) \tag{3}$$

By aggregating the projected features of all the neighbour nodes of a node regarding a certain meta-path, its meta-path based embedding can be learned as:

$$z_i^{\pi_p} = \sigma(\sum_{j \in N_i^{\pi_p}} \alpha_{ij}^{\pi_p} \cdot h_j^{'}) \tag{4}$$

We then introduced multi-head attention mechanism to node-level attention to cater for high variance of graph data. A number K of attention heads are predefined, as the learning process will be repeated K times under different parameters and the semantic-specific embedding is obtained by concatenating all learned embeddings. We denote the embedding matrix of nodes of type π_p as Z_{π_p}.

In order to learn a more informative node embedding, the importance of each meta-path should be considered since they indicate semantic information in the graph. A semantic-level vector q and a simple MLP shared for all meta-paths are designed to curve the importance of a meta-path π_p as:

$$w_{\pi_p} = \frac{1}{|V|} \sum_{i \in V} q^T \cdot tanh(W \cdot z_i^{\pi_p} + b) \tag{5}$$

where V denotes the node set of graph G and $|V|$ represents the number of nodes, while the MLP is defined by a weight matrix W and a bias b. The weight of each meta-path is then obtained by normalizing all meta-paths. By fusing the learned weights and the semantic-specific embedding of the nodes, the final embedding of a HGAT can be calculated as:

$$Z = \sum_{p=1}^{P} \frac{exp(w_{\pi_p})}{\sum_{i=1}^{P} exp(w_{\pi_i})} \cdot Z_{\pi_p} \tag{6}$$

2 Hyper-parameter Setting

For $HGAT_{macro}$ and $HGAT_{micro}$, we set the dimension of their output vector to 128, while the number of attention heads is set 8 and 16 respectively. To enhance the expression of latent feature, we stack up isomorphic HGATs to form a large attention network, i.e. a primary HGAT is connected with itself multiple times by slightly altering the input and output dimension within intervening layers. In practice, $HGAT_{macro}$ is doubled and $HGAT_{micro}$ remains its original form.

For the contrastive learning module, DNN_{clf} has a learning rate of 0.001 and 3 FC layers with fixed neurons numbers as [512, 256, 1]. The first two FC layers use the ReLU and the last FC layer uses sigmoid as their activate function. The self-supervised learning process is executed for 500 epochs, at which the curve of loss and accuracy usually tends to flatten. We adopt mini-batch method to accelerate the training, and the size of each batch is 512.

For the synergy prediction model DNN_{pred} , we mainly adjust the hidden layer size and the learning rate of the model. The number of neurons in the first two FC layers is chosen from 8192, 4096, 2048, 1024, while the learning rate is chosen from 0.00001, 0.0001, 0.001. The mini-batch method is also applied

for this model with a batch size of 512. The maximum number of epochs per training is 500.