Homework 3

Guoxiang Grayson Tong

November 15, 2022

Problem 1:

(A) Define node embedding matrix at layer k as $\mathbf{H}^{(k)}$

$$\mathbf{H}^{(k)} = \left[\boldsymbol{h}_1^{(k)} \middle| \boldsymbol{h}_2^{(k)}, \cdots \middle| \boldsymbol{h}_{|\mathcal{V}|}^{(k)}\right]^T \in \mathbb{R}^{|\mathcal{V}|\times d} , \tag{1}$$

where d is the number of node embedding features. Since we are considering the self-loop, we re-define the adjacency matrix as

$$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I} , \tag{2}$$

and the degree matrix $\tilde{\mathbf{D}}$ is computed from $\tilde{\mathbf{A}}$ via row-sum and diagonalization.

Using Einstein notation, the GCN propagation rule can be written as:

$$\begin{aligned} H_{ip}^{(k)} &= \sigma(W_{pq}^{(k-1)} \tilde{A}_{ij} H_{jq}^{(k-1)} \frac{1}{\sqrt{\tilde{d}_i \tilde{d}_j}}) \\ &= \sigma(W_{pq}^{(k-1)} \tilde{D}_{ii}^{-1/2} \tilde{A}_{ij} \tilde{D}_{jj}^{-1/2} H_{jq}^{(k-1)}) \\ &= \sigma(\tilde{D}_{ii}^{-1/2} \tilde{A}_{ij} \tilde{D}_{jj}^{-1/2} H_{jq}^{(k-1)} W_{qp}^{(k-1)}). \end{aligned} \tag{3}$$

In matrix notation, this is equivalent to

$$\mathbf{H}^{(k)} = \sigma(\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(k-1)} \mathbf{W}^{(k-1),T}) . \tag{4}$$

Since the matrix $\mathbf{W}$ is arbitrarily initialized at the beginning, we then drop the transpose notation for simplicity.

(B) Please see the script

(C) Please see the script

(D) Please see the script

(E) Please see the script

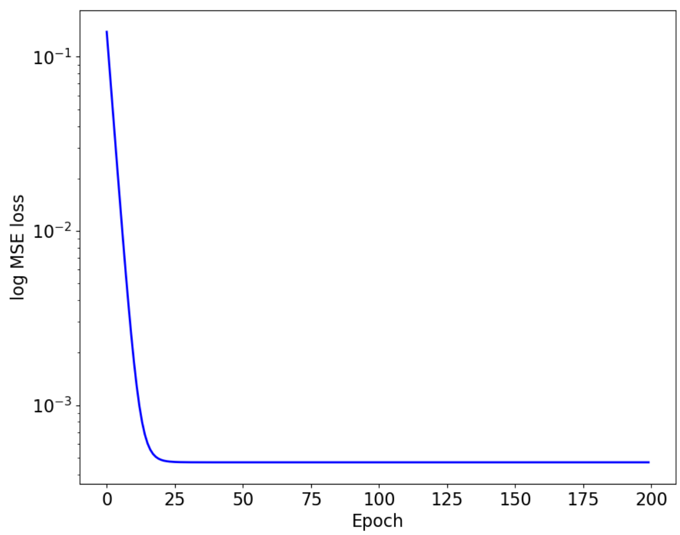


Figure 1: Training loss evolution.

(F) Please see the script

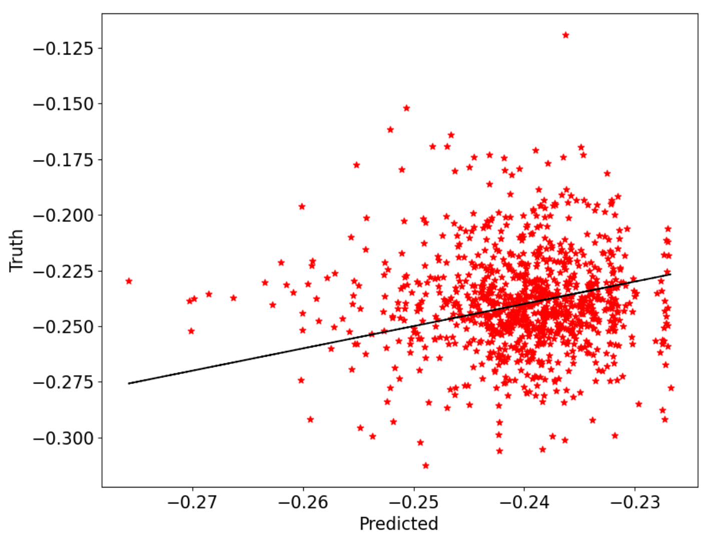


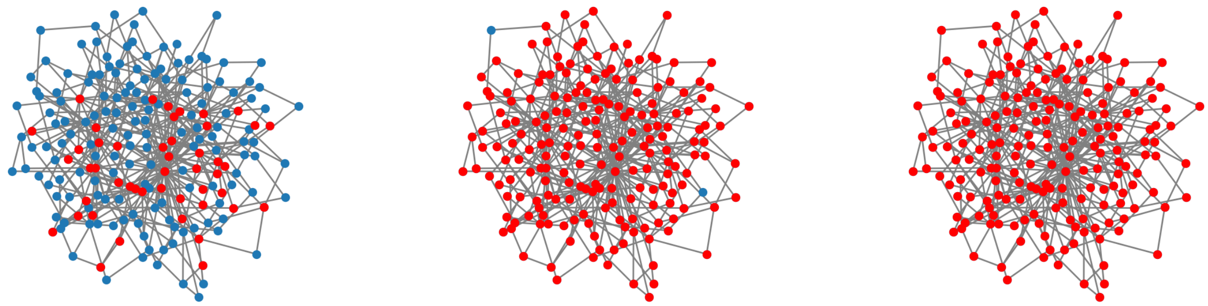
Figure 2: Testing prediction, a line of $y = x$ is super-imposed to show the model performance.

As most points are closely around the $y = x$ line, our model seems to get properly trained.

Problem 2:

(A) Please see the script

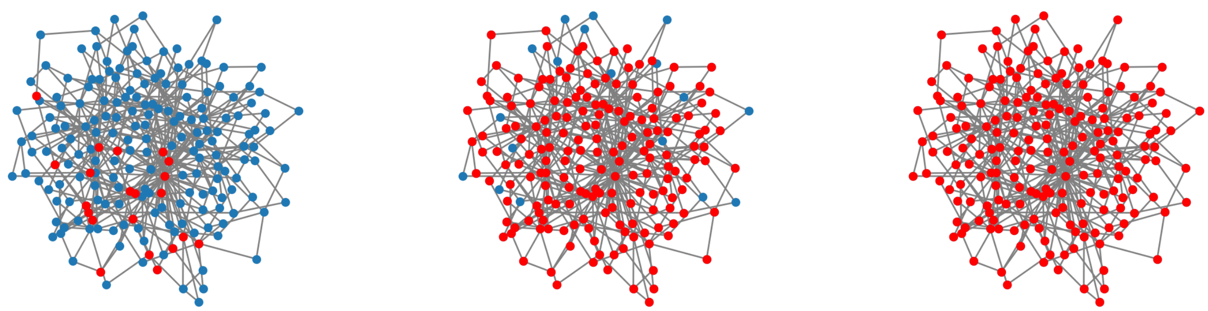
(B) Effective ranges of node 17 with 2,4,6 message passing layers are as follows:



(a) $K = 2$ (b) $K = 4$ (c) $K = 6$

Figure 3: Effective ranges of node 17.

Effective ranges of node 27 with 2,4,6 message passing layers are as follows:

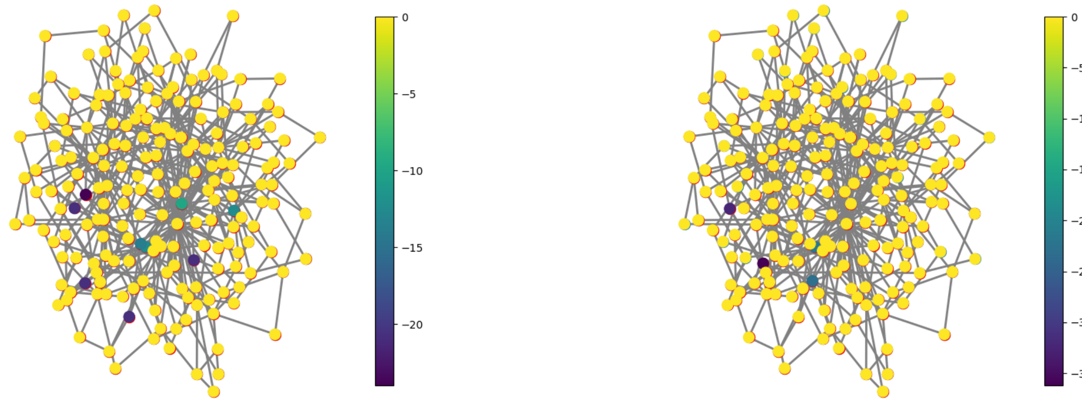


(a) $K = 2$ (b) $K = 4$ (c) $K = 6$

Figure 4: Effective ranges of node 27.

(C) Please see the script

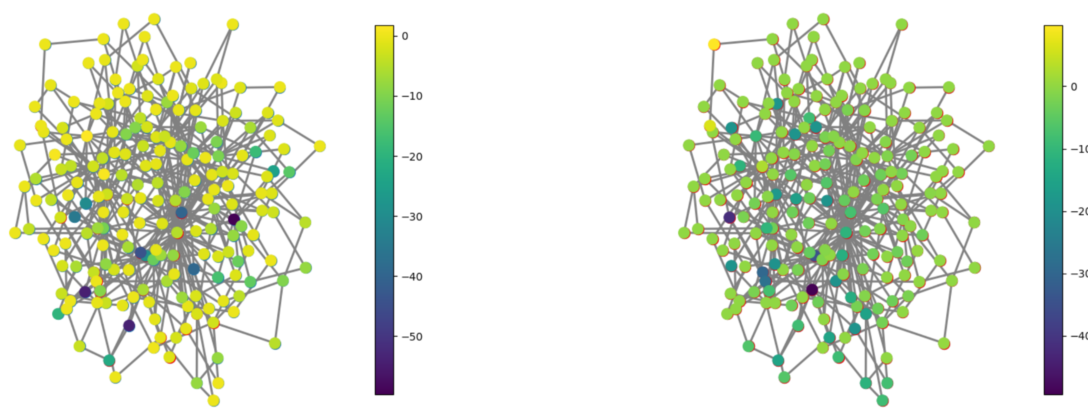
(D) Using model 1 (1-msg passing layer GNN), the influence score I_k as the node embedding is as follows:



(a) $v_{i=17}$ (b) $v_{i=27}$

Figure 5: Influence score I_K of node 17 and node 27, computed by GNN with a single message passing layer.

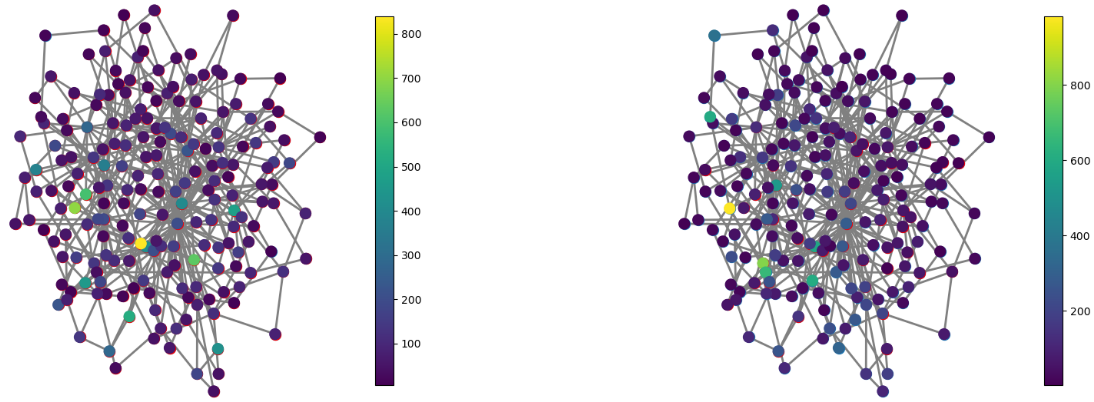
Using model2 (3-msg passing layer GNN), the influence score I_k as the node embedding is as follows:



(a) $v_{i=17}$ (b) $v_{i=27}$

Figure 6: Influence score I_K of node 17 and node 27, computed by GNN with three message passing layers.

Using model3 (5-msg passing layer GNN), the influence score I_k as the node embedding is as follows:



(a) $v_{i=17}$ (b) $v_{i=27}$

Figure 7: Influence score I_K of node 17 and node 27, computed by GNN with five message passing layers.

(E) For the selected nodes, it seems their effective ranges cover the whole graph after around $4 \sim 6$ message passing steps. As a result, the GNN may not be able to obtain new information from the graph after a few message-passing steps but instead averaging out around each neighborhood. This is also clear from the influence score graphs as each node gets more impact from the whole graph as more GCN layers are stacked. Eventually, each node may get equal contributions from all other nodes, i.e. neighborhood information is lost.