Prove the decomposable of the loss function w.r.t edges

First, the regular binary cross-entropy loss of an arbitrary node v can be described as following

$$L(\theta; v, E) = -\sum_{c=1}^{2} y_{v,c} \log \frac{\exp(Aggr(v)_c)}{\sum_{k=1}^{2} \exp(Aggr(v)_k)}, \tag{1}$$

y_c is the true label of class c for node v. Please note that we use $Aggr(v)_c$ to denote the output of the last GNN layer. We omit the detail of aggregation only for simplicity of deriving the loss function. Later we will demonstrate the aggregation methods of GCN, GragphGraph, and GIN work under our loss function.

Since $y_{v,c}$ is an one-hot vector of true label of node v, either $y_{v,1} = 0$ or $y_{v,2} = 0$. Thus, we only need to consider one case. Suppose $y_{v,2} = 0$, we have

$$L(\theta; v, E) = -\log \frac{\exp(Aggr(v)_1)}{\sum_{k=1}^{2} \exp(Aggr(v)_k)}. \tag{2}$$

Then, let the numerator and the denominator of the fraction in the above equation divide $\exp(\sum_{u \in N(v)} X_u \cdot \theta_1)$ at the same time, we get

$$\begin{aligned} L(\theta; v, E) &= -\log \frac{1}{1 + \exp(Aggr(v)_2)/\exp(Aggr(v)_1)}. \\ &= \log \big(1 + \frac{\exp(Aggr(v)_2)}{\exp(Aggr(v)_1)}\big). \end{aligned}$$

According to $\log(1 + x) \sim x$, we have

$$L(\theta; v, E) \approx \exp(Aggr(v)_2 - \exp(Aggr(v)_1). \tag{3}$$

Suppose the GNN model can correctly classify node v as label 1 with a very high confident $((Aggr(v)_2 \ll Aggr(v)_1)$, we drop the first term and obtain

$$L(\theta; v, E) \approx \exp(-Aggr(v)_1). \tag{4}$$

It is obvious that Eqn (4) is linear with respect to $Aggr(v)$ if we can apply a logarithm to it. Consequently, the loss becomes $L(\theta; v, E) \approx -Aggr(v)_1$. Hence, we present a binary log-cross-entropy loss,

$$\mathcal{L}(\theta; v, E) = -\sum_{c=1}^{2} y_{v,c} \log \Big[\log \frac{\exp(Aggr(v)_c}{\sum_{k=1}^{2} \exp(Aggr(v)_k)}\Big], \tag{5}$$

Next, we demonstrate that the binary log-cross-entropy is linear with respect to edges for different aggregation approaches, GCN, GraphSAGE, and GIN.

GCN

Proof. The aggregation method of GCN in node-wise is formulated as

$$Aggr_{GCN}(v) = \sum_{u \in N(v) \bigcup \{v\}} \frac{X_u}{\sqrt{\hat{d}_u, \hat{d}_v}} \theta_c, \tag{6}$$

where $\hat{d}_u$ denotes the degree of node u plus 1, $N(v)$ is the set of nodes that connect to node v, $X \in \mathbb{R}^{N \times d}$ is a matrix represents node features, and $\theta \in \mathbb{R}^{C \times d}$ is the learnable parameter. Please note that $X_u \in \mathbb{R}^d$ and $\theta_c \in \mathbb{R}^d$ correspond to the vector of node features of u and the vector of the learnable parameter for class c, respectively. $\sqrt{\hat{d}_u, \hat{d}_v}$ can be considered as a normalization term. We omit the normalization mainly for analytical purposes. Our empirical results show that with normalization EraEdge still achieves competitive performance. Thus, we have

$$Aggr_{GCN}(v) = \sum_{u \in N(v) \bigcup \{v\}} X_u \theta_c. \tag{7}$$

Plugin the aggregation function to Eqn (4), we obtain the loss function of node v for a GNN model,

$$\mathcal{L}(\theta; v, E) \approx - \sum_{u \in N(v) \bigcup \{v\}} X_u \theta_c. \tag{8}$$

Suppose we remove a set of edges E_{UL}, then the loss becomes

$$\mathcal{L}(\theta; v, E \backslash E_{UL}) \approx - \sum_{u \in N'(v) \bigcup \{v\}} X_u \theta_c, \tag{9}$$

where $N'(v) = \{u | \forall u \in V, (u, v) \notin E_{UL}\}$. Similarly, the loss of v caused by E_{UL} can be formulated as

$$\mathcal{L}(\theta; v, E_{UL}) \approx - \sum_{u \in \bar{N}(v)} X_u \theta_c, \tag{10}$$

where $\bar{N}(v) = \{u | (u, v) \in E_{UL}\}$. It is worth mentioning that Eqn (10) does not have node v itself which is really easy to implement. And we can see

$$\mathcal{L}(\theta; v, E) = \mathcal{L}(\theta; v, E \backslash E_{UL}) + \mathcal{L}(\theta; v, E_{UL}). \tag{11}$$

□

GraphSAGE

Proof. The aggregation method of GraphSAGE can be formulated as

$$aggr_{GraphSAGE}(v) = \theta_{1,c} X_v + \theta_{2,c} \cdot mean_{u \in N(i)}(X_u), \tag{12}$$

where θ_1 and θ_2 are two learnable parameters of GraphSAGE, $mean(\cdot)$ an average function. In this work, we use an alternative option $sum(\cdot)$. Then, the loss function of node v for a GraphSAGE model is

$$\mathcal{L}(\theta; v, E) \approx -\theta_{1,c} X_v + \theta_{2,c} \cdot \sum_{u \in N(i)} X_u, \tag{13}$$

Similar to GCN, we can get

$$\mathcal{L}(\theta; v, E) = \mathcal{L}(\theta; v, E \backslash E_{UL}) + \mathcal{L}(\theta; v, E_{UL}). \tag{14}$$

□

GIN

Proof. The aggregation method of GIN in node-wise can be formulated as

$$aggr_{GIN}(v) = h_\theta(\sum_{u \in N(v)} X_u + (1 + \epsilon) X_v), \tag{15}$$

where ϵ is a hyper-parameter of GIN, h_θ denotes a neural network, such as MLP. In this work, we only apply a linear as h_θ. Then we obtain the loss of node v as

$$\mathcal{L}(\theta; v, E) \approx -(\sum_{u \in N(v)} X_u + (1 + \epsilon) X_v) \theta_c, \tag{16}$$

Similar to GCN, we can get

$$\mathcal{L}(\theta; v, E) = \mathcal{L}(\theta; v, E \backslash E_{UL}) + \mathcal{L}(\theta; v, E_{UL}). \tag{17}$$

□