

## Scaling up by Simplifying GNNs

通过移除 GCN 中的非线性函数, 即可提升大规模数据的可用性。

- 原始 GCN 形式为:  $H^{(k+1)} = \text{RELU}(\tilde{A} H^{(k)} W_k^T)$

其中  $\tilde{A} = D^{-1}A$  或  $\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$  better

- 考虑一个无 RELU 的版本:  $H^{(k+1)} = \tilde{A} H^{(k)} W_k^T$

展开:  $H^{(k)} = \tilde{A} H^{(k-1)} W_{k-1}^T$

$$= \tilde{A} (\tilde{A} H^{(k-2)} W_{k-2}^T) W_{k-1}^T$$

$$= \tilde{A} (\tilde{A} (\dots (\tilde{A} H^{(0)} W_0^T) \dots) W_{k-2}^T) W_{k-1}^T$$

$$= \tilde{A}^k X (W_0^T \dots W_{k-1}^T)$$

$$= \tilde{A}^k X W^T$$

不包含学习参数, 可以预计算

Let  $X = \tilde{A}^k X$ , 则  $H^{(k)} = \tilde{X} W^T$  ( $h_v^{(k)} = W \tilde{X}_v$ )

在训练过程中, 可以首先预计算  $\tilde{X} = \tilde{A}^k X$ , 然后从  $\tilde{X}$  中采

样  $\{v_1, v_2, \dots, v_m\}$ , 计算并嵌入  $h_{v_1}^{(k)} = W \tilde{X}_{v_1}$ ,  $h_{v_2}^{(k)} = W \tilde{X}_{v_2}$ ,

$\dots$ ,  $h_{v_m}^{(k)} = W \tilde{X}_{v_m}$ , 然后进行梯度下降。

日期:

/

- 简化的 GCN 可以高效产生嵌入
- 更灵活的采样方式
- 更稳定的训练
- 较差的表现力 (实践中只是略差, 因为图同质性的存在)