

ICLR19将PageRank与GNN结合以解决GCN层数无法加深的一篇论文

解决的问题

GCN层数增加后性能反而变差，如何加深GCN的层数。

根据GCN的定义，每一层网络用来捕获一跳邻居的信息，例如一个三层的GCN网络捕获的就是一个顶点三跳邻居以内的信息，而现在如果只能用浅层模型，表示只能捕获有限跳内的邻域信息，而有时候要多几跳才能捕获到有用的信息，例如JK-Net中的例子。

做法及创新

这一篇论文的工作其实是接着JK-Net继续往下，在那篇论文中，作者分析了GCN中信息传递这个过程与随机漫步之间的关系，论证了当层数加深之后，GCN会收敛到这个随机漫步的极限分布，而这个极限分布只与图的全局属性有关，没有把随机漫步的起始顶点，或者说是GCN中从邻域中传递和聚合信息的根顶点考虑在内，这么一来，层数加深之后每个顶点聚合出来的样子都差不多，无法区分从而导致性能变差，另一个看待的角度是，因为原始GCN是对所有聚合的信息做平均操作，层数加深之后各个顶点的邻域都变得跟整张图差不多，既然每个顶点的邻域都变得差不多，做的又是平均操作，每个顶点聚合出来的样子就会都差不多。

论文提出的解决办法是引入PageRank的思想，这也是从JK-Net中的结论观察出来的。JK-Net中所说的GCN会收敛到的极限分布的计算方法如下：

$$\pi_{lim} = \hat{A}\pi_{lim}$$

而PageRank的计算方法如下：

$$\pi_{pr} = A_{rw}\pi_{pr}$$

其中 $A_{rw} = AD^{-1}$ ，两个计算方法明显地相似，区别在于，PageRank中邻接矩阵 A 没有考虑根顶点自身，而极限分布的计算里 \hat{A} 是引入了自环的。而Personalized PageRank通过引入自环而考虑了根顶点自身，论文的想法就是将随机漫步的极限分布用Personalized PageRank来代替，它的计算方法为：

$$\begin{aligned}\pi_{ppr}(i_x) &= (1 - \alpha)\hat{A}\pi_{ppr}(i_x) + \alpha i_x \\ \rightarrow \pi_{ppr}(i_x) &= \alpha \left(I_n - (1 - \alpha)\hat{A} \right)^{-1} i_x\end{aligned}$$

其中 i_x 是一个one_hot指示向量，用来从根顶点重新启动。

Personalized PageRank算法的目标是要计算所有节点相对于用户u的相关度。从用户u对应的节点开始游走，每到一个节点都以 α 的概率停止游走并从u重新开始，或者以 $1-\alpha$ 的概率继续游走，从当前节点指向的节点中按照均匀分布随机选择一个节点往下游走。这样经过很多轮游走之后，每个顶点被访问到的概率也会收敛趋于稳定，这个时候我们就可以用概率来进行排名了。

相较于原始的GCN模型，现在根顶点 x 对顶点 y 的影响程度 $I(x, y)$ ，变得与 $\pi_{ppr}(i_x)$ 中的第 y 个元素相关，这个影响程度对于每个根顶点都有不同的取值：

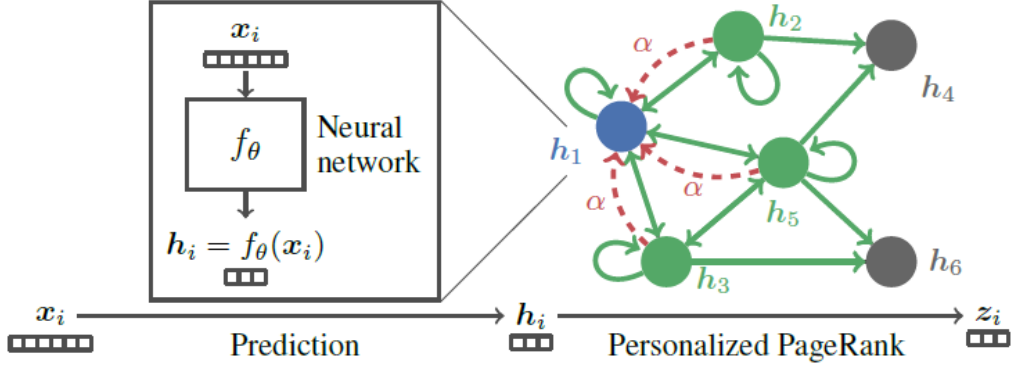
$$I(x, y) \propto \prod_{ppr}^{(yx)} \prod_{ppr}^{(yx)} = \alpha \left(I_n - (1 - \alpha)\hat{A} \right)^{-1} \mathcal{I}_n$$

PPNP

经过上面的铺垫与介绍，论文提出的模型PPNP可以表示为：

$$Z_{PPNP} = \text{softmax}\left(\alpha\left(I_n - (1 - \alpha)\hat{A}\right)^{-1}H\right), H_{i,:} = f_{\theta}(X_{i,:})$$

其中 X 为特征矩阵， f_{θ} 是一个参数为 θ 的神经网络，用来产生预测类别 $H \in \mathbb{R}^{n \times c}$ 。



由公式和图中都可以看到，PPNP其实是由两部分组成，左边的神经网络与右边的信息传递网络，神经网络部分就类似于在[GCN](#)中介绍的，输入顶点特征与图的结构信息（邻接矩阵），输出顶点新的特征表示。信息传递网络部分，在PPNP中通过它来得到预测标签，而原始GCN的做法是 $Z_{GCN} = \text{softmax}(\hat{A}HW)$ ，其中 W 是每层网络的参数。

APPNP

从前面的构造方式可以看到，矩阵 \prod_{ppr} 将会有 $\mathbb{R}^{n \times n}$ 大小，会带来时间和空间上的复杂度。因此论文提出了一种近似的计算方法APPNP，计算方式如下：

$$\begin{aligned} Z^{(0)} &= H = f_{\theta}(X) \\ Z^{(k+1)} &= (1 - \alpha)\hat{A}Z^{(k)} + \alpha H \\ Z^{(K)} &= \text{softmax}\left((1 - \alpha)\hat{A}Z^{(K-1)} + \alpha H\right) \end{aligned}$$

其中 K 为信息传递的跳数或者说是随机漫步的步数， $k \in [0, K - 2]$ ，这样一来就不用构造一个 $\mathbb{R}^{n \times n}$ 的矩阵了。（不知道为什么...）

数据集

Citeseer、Cora-ML、Pubmed、MS Academic