

SIGIR19年将神经网络与协同过滤结合的一篇论文

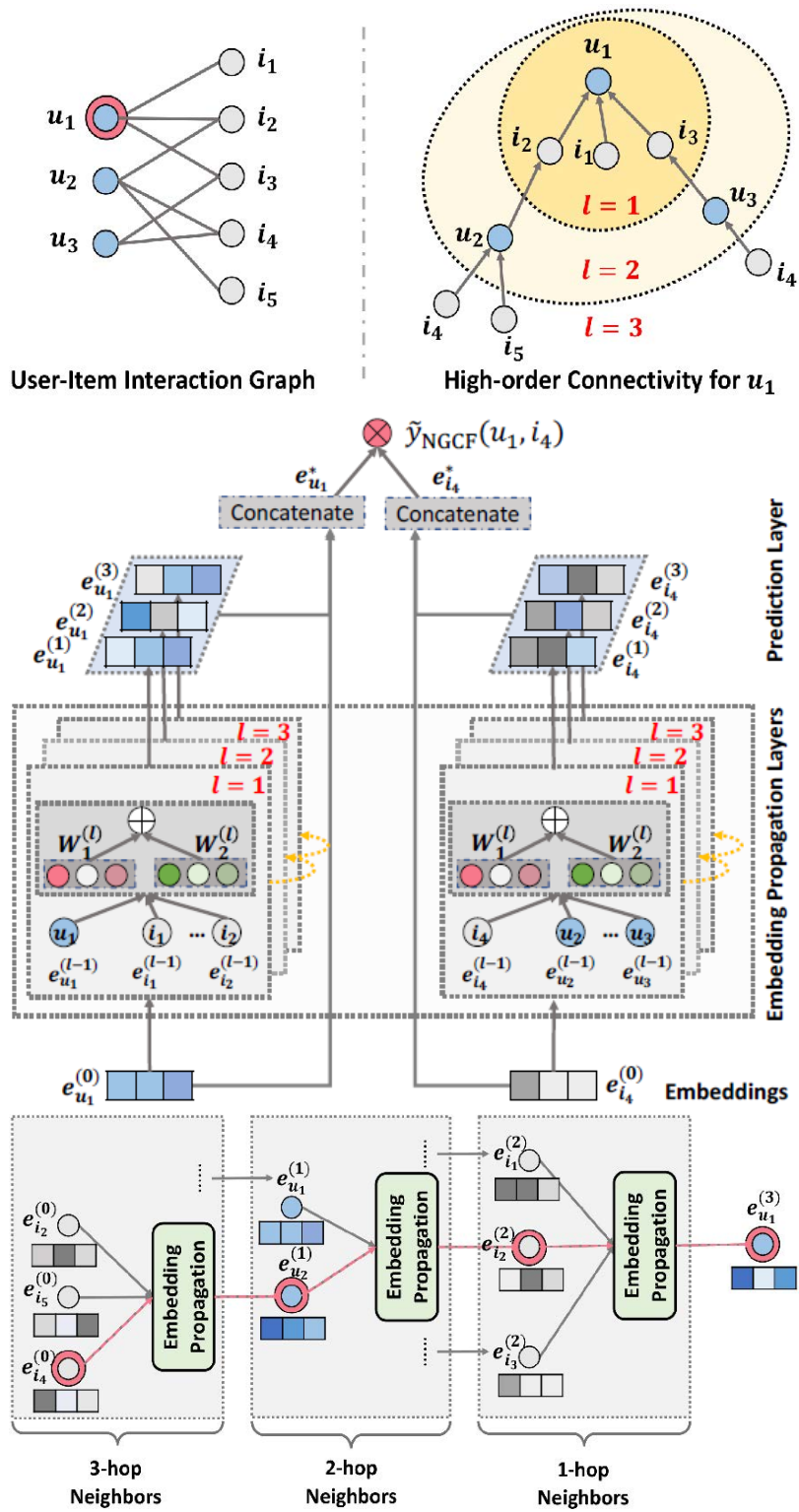
解决的问题

在现有的推荐模型中，用户和物品的embedding只考虑了它们自身的特征，没有考虑用户-物品的交互信息

做法及创新

$$\begin{aligned}\hat{y}_{NGCF}(u, i) &= e_u^{*T} e_i^* \\ e_u^* &= e_u^{(0)} \parallel \dots \parallel e_u^{(L)} \\ e_i^* &= e_i^{(0)} \parallel \dots \parallel e_i^{(L)} \\ e_u^{(l)} &= \text{LeakyReLU}(m_{u \leftarrow u}^{(l)} + \sum_{i \in N_u} m_{u \leftarrow i}^{(l)}) \\ \begin{cases} m_{u \leftarrow i}^{(l)} &= p_{ui}(W_1^{(l)} e_i^{(l-1)} + W_2^{(l)}(e_i^{(l-1)} \odot e_u^{(l-1)})) \\ m_{u \leftarrow u}^{(l)} &= W_1^{(l)} e_u^{(l-1)} \end{cases} \\ m_{u \leftarrow i} &= \frac{1}{\sqrt{|N_u||N_i|}}(W_1 e_i + W_2(e_i \odot e_u))\end{aligned}$$

1. 通过堆叠 l 层embedding传播层，一个用户（物品）可以获得它的 l 跳邻居所传播的信息，如下图所示，通过这种方法来建模用户-物品交互信息中的高阶connectivity，下图展示的是一个三阶的例子:



2. 传统GCN推荐方法中，message embedding只考虑物品embedding e_i ，论文中将用户embedding与物品embedding的交互也纳入考虑，解释为“This makes the message dependent on the affinity between e_i and e_u , e.g., passing more messages from the similar items.”
3. 两个层面上的dropout: message & node dropout。前者表示在第 l 层传播层中，只有部分信息会对最后的表示有贡献；后者表示在第 l 层传播层中，随机地丢弃一些顶点。

数据集

Gowalla、Yelp2018、Amazon-book

