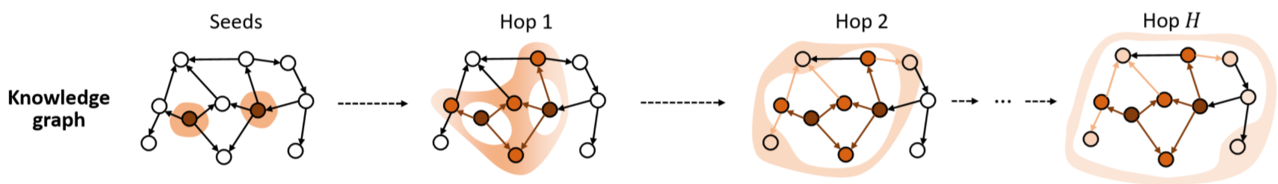
介绍

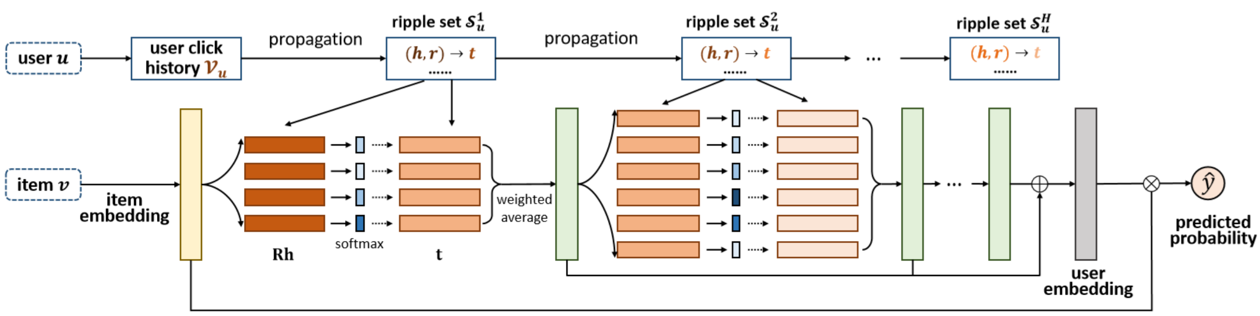
为了解决协同过滤的稀疏性和冷启动问题，通常会利用辅助信息来提高推荐系统的准确率。这篇文章是通过将知识图谱作为辅助信息的来源来提高推荐系统的准确性。现有的利用知识图谱做推荐的主要有两类：**一类是基于嵌入的方法、一类是基于路径的方法**。基于嵌入的方法更时候用于在图谱上的运用，比如做连接预测。基于路径的方法严重依赖于手动设计的元路径在实践中很难优化，另外在一些情况下也无法手动设计元路径。为了解决现有方法的局限性，这篇问题提出了Ripplenet

Ripple是波纹的意思，RippleNet就是模拟用户兴趣在知识图谱上的一个传播过程，如下图所示：波纹是怎么来的呢？是你往水里面扔东西产生的。所以在这里ripple net与RS的结合，就相当于往水里投了一块石头（这里是石头就是波纹的中心，也就是RS中的用户历史点击矩阵） 扔石头就会激起一层又一层的波纹，这里被激起的波纹就对应了知识图谱一个又一个的实体。由用户的历史纪录激起的水波，就是用户潜在感兴趣的item 除此之外波纹还有一个特点，它会随着层数的变大而逐渐衰减，这里类比到ripplenet也是同样的



具体方法：

The framework of RippleNet is illustrated in Figure 2. RippleNet takes a user u and an item v as input, and outputs the predicted probability that user u will click item v. For the input user u, his



对于每一个user，他的历史点击被最为石头投入水中（图谱）利用ripplenet来传播波纹。

这里有两个定义：

Definition 1 (relevant entity). *Given interaction matrix* $\mathbf{Y}$ *and knowledge graph* $\mathcal{G}$, *the set of* k*-hop relevant entities for user* u *is defined as*

$$\mathcal{E}_u^k = \{t \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \mathcal{E}_u^{k-1}\}, \quad k = 1, 2, ..., H, \tag{2}$$

where $\mathcal{E}_u^0 = \mathcal{V}_u = \{v \mid y_{uv} = 1\}$ *is the set of user's clicked items in the past, which can be seen as the seed set of user* u *in KG.*

Ripple Set：用户u的k-hop ripple set被定义为以k-1 Relevant Entity 为head的相关三元组

Definition 2 (ripple set). *The* k*-hop ripple set of user* u *is defined as the set of knowledge triples starting from* $\mathcal{E}_u^{k-1}$:

$$\mathcal{S}_u^k = \{(h, r, t) \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \mathcal{E}_u^{k-1}\}, \quad k = 1, 2, ..., H. \tag{3}$$

- 第一步计算pi：在第一层对item的embedding、关系的embedding和head的embedding的积做一个softmax

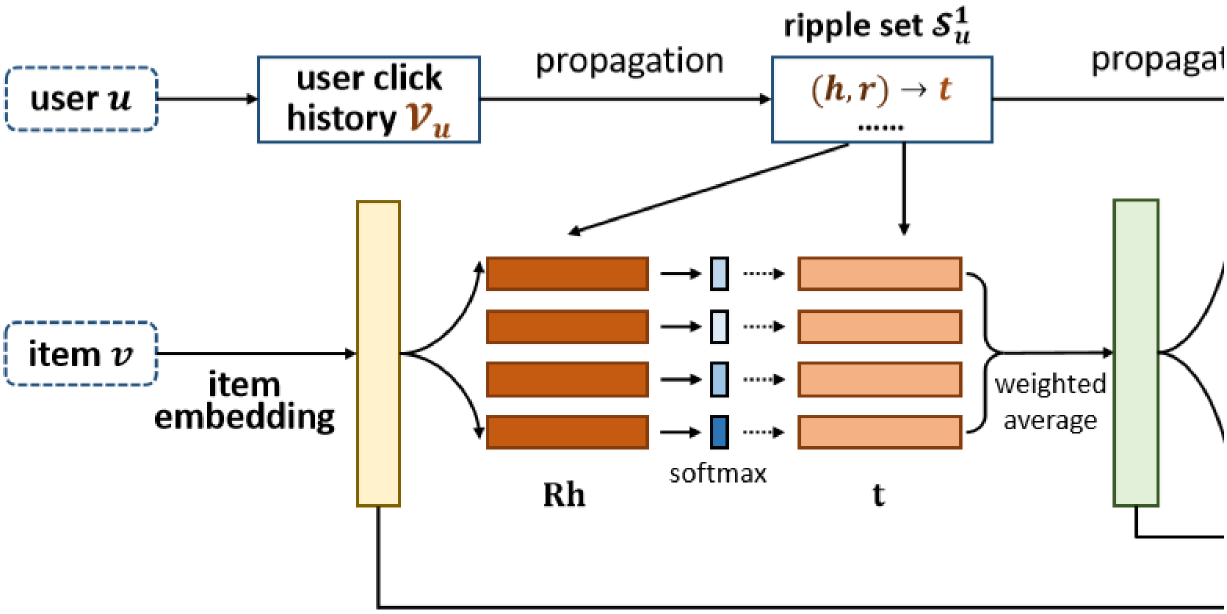
$$p_i = \text{softmax}\left(\mathbf{v}^{\mathrm{T}} \mathbf{R}_i \mathbf{h}_i\right) = \frac{\exp\left(\mathbf{v}^{\mathrm{T}} \mathbf{R}_i \mathbf{h}_i\right)}{\sum_{(h,r,t) \in \mathcal{S}_u^1} \exp\left(\mathbf{v}^{\mathrm{T}} \mathbf{R} \mathbf{h}\right)}, \tag{4}$$

pi可以看作是item v和头实体在R关系下的相似度

- 第二步，在得到概率后，把在第一层的ripple set 的尾实体的做一个按权求和得到向量o_u^1.向量o_u^1可以看作是用户点击历史对于item v的一阶响应（第一层水波）

$$\mathbf{o}_u^1 = \sum_{(h_i, r_i, t_i) \in \mathcal{S}_u^1} p_i \mathbf{t}_i,$$

- 第三步重复这个步骤，把得到的向量o_u^1.作为v传到下一层用同样的方法计算。



对应到流程图里就是这样的，绿色的长方形就是向量o_u^1.

最后虽然最后一层的响应向量o_u^H包含之前所有层的信息，但是由于之前的信息可能会被冲淡，因此对于用户u的关于item v的响应还是把所有层的响应加在了一起。得到向量u

$\mathbf{o}_u^1, \mathbf{o}_u^2, ..., \mathbf{o}_u^H$. The embedding of user u with respect to item v is calculated by combining the responses of all orders:

$$\mathbf{u} = \mathbf{o}_u^1 + \mathbf{o}_u^2 + ... + \mathbf{o}_u^H, \tag{6}$$

since they may be diluted in $\mathbf{o}_u^H$. Finally, the user embedding and item embedding are combined to output the predicted clicking probability:

$$\hat{y}_{uv} = \sigma(\mathbf{u}^\mathrm{T}\mathbf{v}), \tag{7}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function.

最后，把用户embedding和item embedding的积做一个sigmoid得到预测点击概率。

loss函数

$$\min \mathcal{L} = -\log\left(p(\mathrm{Y}|\Theta,\mathcal{G})\cdot p(\mathcal{G}|\Theta)\cdot p(\Theta)\right)$$
$$= \sum_{(u,v)\in \mathrm{Y}} -\Big(y_{uv}\log\sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v}) + (1-y_{uv})\log\left(1-\sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v})\right)\Big)$$
$$+ \frac{\lambda_2}{2}\sum_{r\in\mathcal{R}}\|\mathbf{I}_r - \mathbf{E}^{\mathrm{T}}\mathbf{R}\mathbf{E}\|_2^2 + \frac{\lambda_1}{2}\Big(\|\mathbf{V}\|_2^2 + \|\mathbf{E}\|_2^2 + \sum_{r\in\mathcal{R}}\|\mathbf{R}\|_2^2\Big) \tag{13}$$

loss是通过最大化后验概率得到的

事情还没有发生，要求这件事情发生的可能性的大小，是先验概率。

事情已经发生，要求这件事情发生的原因是由某个因素引起的可能性的大小，是后验概率。

在给定知识图谱G，用户的隐式反馈Y（得的结果）时，我们希望最大化后验概率：

$$\max p(\Theta|\mathcal{G},\mathrm{Y}),$$

后验概率通过贝叶斯公式展开如下：

$$p(\Theta|\mathcal{G},\mathrm{Y}) = \frac{p(\Theta,\mathcal{G},\mathrm{Y})}{p(\mathcal{G},\mathrm{Y})} \propto p(\Theta)\cdot p(\mathcal{G}|\Theta)\cdot p(\mathrm{Y}|\Theta,\mathcal{G})$$

其中，我们认为参数的先验概率服从0均值的正态分布：

$$p(\Theta) = \mathcal{N}(\mathbf{0},\ \lambda_1^{-1}\mathbf{I}).$$

第二项的似然函数形式如下： three-way tensor factorization method

希望得到embedding能够更好的还原三元组关系

类似transE transH等，这个是一种为推荐特殊设置的嵌入方法

$$p(\mathcal{G}|\Theta) = \prod_{(h,r,t)\in\mathcal{E}\times\mathcal{R}\times\mathcal{E}} p((h,r,t)|\Theta)$$
$$= \prod_{(h,r,t)\in\mathcal{E}\times\mathcal{R}\times\mathcal{E}} \mathcal{N}(I_{h,r,t} - \mathbf{h}^{\mathrm{T}}\mathbf{R}\mathbf{t},\ \lambda_2^{-1}),$$

第三项二分类似然函数：

$$p(\mathrm{Y}|\Theta,\mathcal{G}) = \prod_{(u,v)\in\mathrm{Y}} \sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v})^{y_{uv}} \cdot \left(1 - \sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v})\right)^{1-y_{uv}}$$

因此最优问题就转化为最小化上述三项似然函数乘积的负对数。得到最后的loss函数：

$$\begin{aligned}\min \mathcal{L} &= -\log\left(p(\mathrm{Y}|\Theta,\mathcal{G}) \cdot p(\mathcal{G}|\Theta) \cdot p(\Theta)\right)\\ &= \sum_{(u,v)\in\mathrm{Y}} -\left(y_{uv}\log\sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v}) + (1-y_{uv})\log\left(1-\sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v})\right)\right)\\ &\quad + \frac{\lambda_2}{2}\sum_{r\in\mathcal{R}} \|\mathbf{I}_r - \mathbf{E}^{\mathrm{T}}\mathbf{R}\mathbf{E}\|_2^2 + \frac{\lambda_1}{2}\left(\|\mathbf{V}\|_2^2 + \|\mathbf{E}\|_2^2 + \sum_{r\in\mathcal{R}}\|\mathbf{R}\|_2^2\right)\end{aligned} \tag{13}$$