

基于知识图谱融入注意力机制商品推荐的可解释性方法

陆倩平^{1,2}, 王红斌^{1,2}

¹昆明理工大学信息工程与自动化学院, 云南 昆明

²昆明理工大学云南省人工智能重点实验室, 云南 昆明

收稿日期: 2022年5月2日; 录用日期: 2022年6月3日; 发布日期: 2022年6月10日

摘要

近年来,知识图谱与推荐系统相结合已经被证明是提高推荐准确率以及为推荐带来可解释性的有效方法,知识图谱中项目之间的连接为用户-项目交互提供了丰富的信息。但现有利用知识图谱的推荐算法中没有考虑到实体之间的相关性。针对这一问题,我们提出了基于知识图谱融入注意力机制商品推荐的可解释性方法。该方法利用注意力机制来区分出邻居节点的重要性,并通过该重要性将实体的邻居节点嵌入实体中,来获取更有效的用户实体表示,提高了推荐的准确性也为推荐项目带来了可解释性。我们将所提出的方法应用于Clothing和Cell_Phones两个数据集,实验结果表明,本文提出的方法通过注意力机制在一定程度上解决了在探索用户偏好时实体间的不相关性问题。

关键词

知识图谱, 可解释推荐, 注意力机制

An Interpretability Method for Commodity Recommendation Based on Knowledge Graph Incorporating Attention Mechanism

Qianping Lu^{1,2}, Hongbin Wang^{1,2}

¹Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming Yunnan

文章引用: 陆倩平, 王红斌. 基于知识图谱融入注意力机制商品推荐的可解释性方法[J]. 计算机科学与应用, 2022, 12(6): 1506-1517. DOI: 10.12677/csa.2022.126150

²Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming Yunnan

Received: May 2nd, 2022; accepted: Jun. 3rd, 2022; published: Jun. 10th, 2022

Abstract

In recent years, the combination of knowledge graphs and recommender systems has been proven to be an effective method to improve recommendation accuracy and bring interpretability to recommendations, and the connections between items in knowledge graphs provide rich information for user-item interactions. However, the existing recommendation algorithms using knowledge graphs do not consider the correlation between entities. In response to this problem, we propose an interpretability method for product recommendation based on knowledge graph with attention mechanism. This method uses the attention mechanism to distinguish the importance of neighbor nodes, and embeds the neighbor nodes of the entity into the entity through the importance to obtain a more effective user entity representation, which improves the accuracy of the recommendation and brings interpretability to the recommended items. We apply the proposed method to the Clothing and Cell_Phones datasets, and the experimental results show that the method proposed in this paper solves the problem of irrelevance between entities when exploring user preferences through the attention mechanism to a certain extent.

Keywords

Knowledge Graph, Explainable Recommendation, Attention Mechanism

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着科学技术的快速发展,我们能接收到的信息量不断上升,我们如何才能在这个信息爆炸的时代快速挑选出我们所需的信息成为了一个热点的研究问题。而个性化推荐系统是信息过滤的重要手段,是解决信息超载现状[1]的有效方法。现有传统的推荐方法存在着以下三个问题,1) 用户和物品之间的交互非常稀疏。例如,一个购物软件上可能有几百万件商品,但用户购买的商品可能只有几百件。推荐方法需要预测大量未知信息,但能利用的已观测到的数据却相对稀少,这会导致过拟合问题的产生。2) 由于系统中缺乏历史交互信息,无法为新用户或项目进行准确建模和推荐,这种情况也称为冷启动。3) 现在的推荐系统对于推荐商品的解释还比较随意甚至是缺乏可解释性。为了解决上述问题,研究者们将知识图谱(Knowledge Graph, KG)引入推荐系统中,知识图谱是一个多关系异构体,其中包含大量的实体和关系,一般以三元组形式呈现,例如(ipad, 品牌, Apple),其中 ipad 和 Apple 是三元组中的头实体和尾实体,品牌是三元组中的关系。然而在知识图谱中实体之间会存在相关性不高的问题,导致推荐的准确率不高。针对该问题我们使用注意力机制融入知识图谱中并将其与实体的上下文信息融入实体来重新定义

实体的嵌入来构建用户向量。综上所述, 本文主要贡献如下:

1) 我们提出了将带有关系的实体间的注意力机制融入基于知识图谱的推荐模型重新定义实体的嵌入, 以此来优化用户向量来作出推荐。

2) 我们将该算法与其它基于知识图谱的推荐算法在数据集 Cell_phone 和 Clothing 上进行了比较, 评价指标准确率、点击率、召回率和 NDCG 的结果验证了本文所提算法能够提高 top-K 推荐的性能, 同时也为推荐结果提供了可解释性。

2. 相关工作

近年来, 对于在推荐系统中加入知识图谱引起了研究者的兴趣。知识图谱被广泛用于各种应用中比如语义搜索、知识问答等等。总的来说, 现有的基于知识图谱的推荐包含三种类型, 基于路径、基于嵌入和基于传播的方法。

基于路径的推荐系统应用知识图谱主要是基于该图谱预先定义用户和项目之间的联系, 然后将其放入模型中进行预测, 基于路径的方法也有一定的效果和可解释性。Zhao 等人[2]提出 FMG 算法, 通过 HIN 和 meta-graph, 提供了一种简单有效的框架, 既能够非常灵活地利用边的信息来提升推荐效果。同时, 还能利用人工设计的 meta-graph 来保留必要的语义信息, 从而对推荐结果提供一定的可解释性。YI 等人[3]提出了一种基于元路径的随机游走策略, 用于生成有意义的网络嵌入节点序列, 以嵌入 HINs。学习节点嵌入首先用一组融合函数进行转换, 然后将其集成为扩展矩阵分解模型。对扩展后的 MF 模型[4]和融合函数执行联合优化预测任务。基于嵌入的方法通过使用 KGE 算法将学到的实体嵌入向量用于推荐系统框架中, 使用用户和项目表征的点积结果来预测推荐结果[5] [6]。无论是基于嵌入的方法还是基于路径的方法都只利用了知识图谱中的部分信息。这就使得知识图谱的信息不能得到充分的利用。因此为了让两种方法的优势能够充分发挥来获得更好的推荐, 提出了将两种方法相结合的方法, 主要思想通过路径中的关系, 再使用嵌入的方法优化向量的表示。例如 Wang 等[7]提出了 Ripple Net 模型, 这是基于传播的推荐方法的一个典型模型, 该模型通过以用户的交互记录为中心在知识图谱中像水波纹一样沿着关系向外扩展寻找用户的偏好, 通过用户偏好为用户建模来进行推荐, 通过实验该方法的有效性也得到了验证。与 Ripple Net 类似, Tang 等人[8]提出了 AKUPM, 该模型通过偏向向不同的实体传播用户的偏好。还有研究者利用图卷积网络[9] [10]和图网络的注意力机制[11] [12] [13]对知识图谱进行建模, 并使其优化目标与推荐系统一致, 从而提升了推荐结果。

尽管以上方法取得了一定的成果, 但基于路径的方法在推荐前需要构建大量的元路径, 严重依赖人工设计的元路径的数量和质量, 并且当知识图谱发生改变时, 就需要重新针对该知识图谱设计出有效的元路径。基于嵌入的方法虽然解决了定义大量元路径的问题但没有充分利用到知识图谱的信息。基于传播的方法通过在知识图谱上传播信息来探索用户的偏好, 但在知识图谱中不是所有实体都有着较高的相关性, 这不可避免地会引入与特定用户项目连接性无关的噪声, 从而误导用户偏好的推断。为解决上述问题, 我们提出了基于知识图谱融入注意力机制商品推荐的可解释性方法。

3. 研究思路和方法

本文受到 RippleNet 框架和关系预测方法[14]的启发提出了基于知识图谱融入注意力机制商品推荐的可解释性方法, 该推荐方法分为以下四个步骤: 第一步使用随机初始化向量表示用户实体、项目实体和他们之间的关系的表示; 第二步通过实体间的注意力机制来更新实体的嵌入; 第三步通过用户偏好和第二步中获得的实体嵌入来优化用户向量; 第四步通过用户嵌入和项目嵌入来预测点击概率。其框架如图 1 所示:

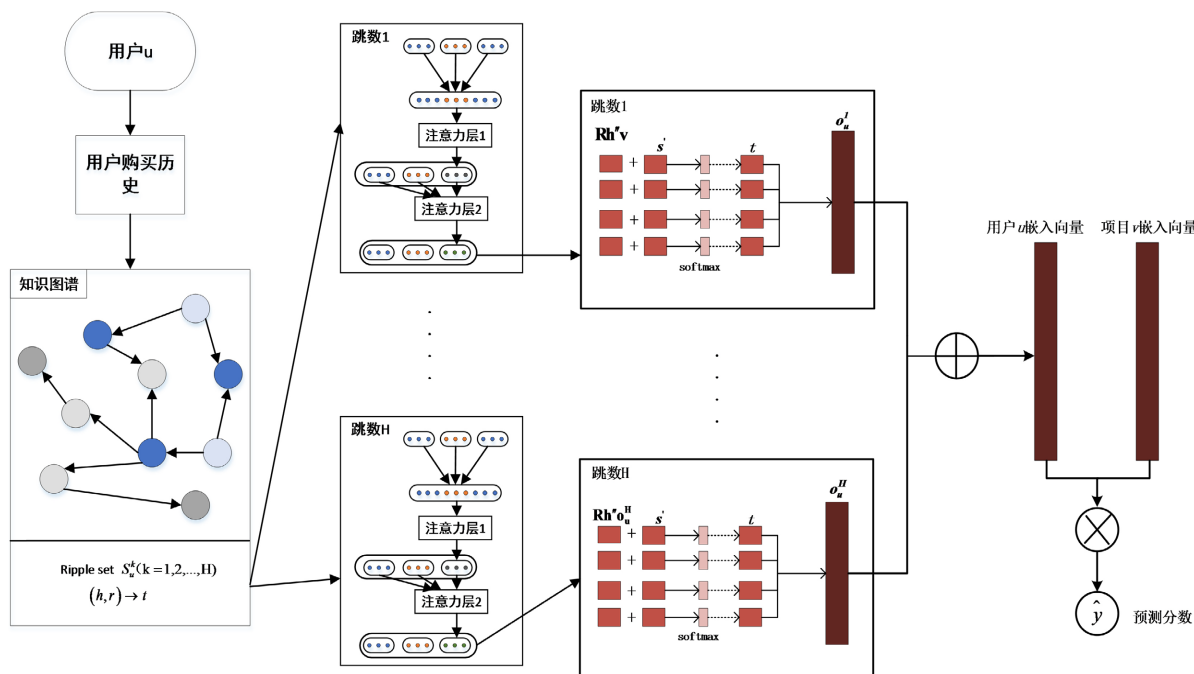


Figure 1. Overall framework of the method

图 1. 方法总体框图

3.1. 相关定义

在商品推荐问题中, 有用户集 U 和商品项目集 V , 用户 - 商品项目的交互矩阵 $Y = \{y_{uv} | u \in U, v \in V\}$ 是基于用户的隐式反馈而定义的, 如下所示:

$$y_{uv} = \begin{cases} 1, & \text{如果用户 } u \text{ 和项目 } v \text{ 有交互} \\ 0, & \text{反之} \end{cases} \quad (1)$$

知识图谱 G , G 中包含大量的三元组 (h, r, t) , $h \in E$, $r \in R$, $t \in E$ 是分别为三元组的头实体、关系和尾实体, 其中 E 表示实体集合, R 表示实体之间的关系集合。给定交互矩阵 Y 和知识图谱 G , 预测用户 u 与项目 v 交互的概率 \hat{y}_{uv} , 并生成 top-N 推荐列表:

$$\hat{y}_{uv} = \mathcal{F}(u, v, \Theta) \quad (2)$$

其中 $\mathcal{F}(\cdot)$ 是推荐方法, Θ 是方法中的参数集合。

为了用 KG 来描述用户层次扩展的偏好, 我们递归地定义了用户的偏好。用户 u 的 k-hop 相关实体集合如下:

本文使用的符号与原始 RippleNet 框架一致, 其中关键的两个相关定义如下:

定义 1 (相关实体) 通过交互矩阵 Y 和知识图谱 G , 对于用户 u 的相关 k-hop 实体的集合定义如下:

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

其中 $\mathcal{E}_u^0 = V_u = \{v | y_{uv} = 1\}$ 是用户过去点击的实体项目集合, 可以看作知识图谱中用户 u 的种子集, H 是用户 u 在知识图谱中的最大跳数。

定义 2 (波纹集) 用户 u 的 k-hop 波纹集定义为从 \mathcal{E}_u^{k-1} 开始的三元组的集合:

$$S_u^k = \{(h, r, t) | (h, r, t) \in G \text{ and } h \in \mathcal{E}_u^{k-1}\}, \quad k = 1, 2, 3, \dots, H. \quad (4)$$

3.2. 注意力聚合层

本节将详细说明提出的实体间的注意力机制, 该机制的作用是通过实体的近邻实体权重来将这些实体聚集在一起表示新实体的嵌入, 使得最终该节点的嵌入信息包含近邻实体的信息, 该实体的这些近邻实体也包括该实体本身, 并且通过此机制避免了不相关实体的相关信息。

为了获得实体 \mathbf{h}_i 的新嵌入, 我们通过对实体和关系特征向量的连接进行线性变换来学习这些嵌入, 该特征向量对应于 S_u^k 中一个相应的三元组 (h_i, r_i, t_i) , 如公式(5)所示:

$$\mathbf{c}_{h_i r_i t_i}^k = \mathbf{W}_1 [\mathbf{h}_i \parallel \mathbf{R}_i \parallel \mathbf{t}_i] \quad (5)$$

其中 $\mathbf{R}_i \in \mathbb{R}^d$ 、 $\mathbf{h}_i \in \mathbb{R}^d$ 和 $\mathbf{t}_i \in \mathbb{R}^d$ 是关系 r_i 、头 h_i 和尾 t_i 的嵌入, \parallel 表示拼接操作, \mathbf{W}_1 表示权值矩阵, 然后我们学习了第 k 跳中每个三元组的重要性 $p_{h_i r_i t_i}^k$ 。我们通过一个权值矩阵 \mathbf{W}_2 的线性变换, 然后应用 LeakyRelu 激活函数来得到三元组的绝对关注值, 也就是尾实体 t 在关系 r 下对头实体 h 的重要程度, 公式如(6)所示:

$$p_{h_i r_i t_i}^k = \text{LeakyRelu}(\mathbf{W}_2 \mathbf{c}_{h_i r_i t_i}^k) \quad (6)$$

在获得尾实体 r 对于头实体 h 的绝对关注值后, 然后使用 softmax 函数对 $p_{h_i r_i t_i}^k$ 进行归一化处理, 公式如(7)所示:

$$\alpha_{h_i r_i t_i}^k = \text{softmax}(p_{h_i r_i t_i}^k) = \frac{\exp(p_{h_i r_i t_i}^k)}{\sum_{j \in R_{h_i}^k} \sum_{e \in E_{h_i}^k} \exp(p_{h_i j_e}^k)} \quad (7)$$

其中 $R_{h_i}^k$ 是在第 k 跳中连接头实体 h_i 和尾实体 t_i 的关系集合, $E_{h_i}^k$ 是头实体 h_i 的相邻实体集合。新的实体 h'_i 的嵌入是由每个三元组嵌入和注意力值加权的表征, 如方程(8)所示:

$$\mathbf{h}'_i = \sigma \left(\sum_{j \in R_{h_i}^k} \sum_{e \in E_{h_i}^k} \alpha_{h_i r_i t_i}^k \mathbf{c}_{h_i r_i t_i}^k \right) \quad (8)$$

每个实体 h 嵌入都是由其近邻实体聚合表示的。我们利用多头注意力来稳定学习过程, 将实体注意力重复 N 次, 公式为(9)所示:

$$\mathbf{h}'_i = \parallel_{n=1}^N \sigma \left(\sum_{j \in R_{h_i}^k} \sum_{e \in E_{h_i}^k} \alpha_{h_i r_i t_i}^k \mathbf{c}_{h_i r_i t_i}^k \right) \quad (9)$$

其中 \parallel 代表的是拼接操作, σ 是一个任意的非线性激活函数, 在我们模型的最后一层中, 我们采用平均的方法来得到实体的最终嵌入向量, 如公式(10)所示。

$$\mathbf{h}'_i = \sigma \left(\frac{1}{N} \sum_{N=1}^N \sum_{j \in R_{h_i}^k} \sum_{e \in E_{h_i}^k} \alpha_{h_i r_i t_i}^k \mathbf{c}_{h_i r_i t_i}^k \right) \quad (10)$$

对于用户向量中头实体的嵌入我们的模型最终采用将基于注意力嵌入层得到的新 \mathbf{h}'_i 和原实体 \mathbf{h}_i 进行相加, 使得原实体的特征也参与了最终实体的表示, 得到最终实体的嵌入, 如公式(11)所示。

$$\mathbf{h}''_i = \mathbf{h}_i + \mathbf{h}'_i \quad (11)$$

3.3. 用户向量构建层

在上节中, 通过注意力嵌入层我们获取到了新的实体嵌入 \mathbf{h}''_i , 然后将每个候选项目嵌入表示为一个

d 维向量 $\mathbf{v} \in \mathbb{R}^d$, 给定项目嵌入 \mathbf{v} 和用户的 1-hop 波纹集合 S_u^1 , 通过比较项目 v 和头 h_i 和该三元组中的关系 r_i 进行比较然后和尾实体的情感分数进行相加, 为 S_u^1 中的每个三元组 (h_i, r_i, t_i) 计算一个相关概率 p_i , 如公式(12)、(13)所示:

$$s_i = \frac{\sum_{j=1}^n (pol_j + r_j)}{n} \quad (12)$$

$$p_i = \text{softmax}(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i'' + s_i') = \frac{\exp(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i'' + s_i')}{\sum_{(h,r,t) \in S_u^1} \exp(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i'' + s_i')} \quad (13)$$

其中 \mathbf{R}_i 和 \mathbf{h}_i'' 是关系 r_i 和更新后头实体 h_i 的嵌入, s_i 是一个商品评论综合分数, pol 是用户评论文本的情感极性, 包括积极的情感、消极的情感和中性的情感, r 是用户对商品的评论分数, n 是同一个商品的评论总个数。得到商品评论综合分数后, 我们将 s_i 转化为与 $\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i''$ 同维度的矩阵 s_i' , 再把 s_i' 加入到每跳波纹集中。

在上一步得到相关概率后, 取对应相关概率加权的 S_u^1 中尾部的总和, 得到向量 \mathbf{o}_u^1 , 如公式(14)所示:

$$\mathbf{o}_u^1 = \sum_{(h_i, r_i, t_i) \in S_u^1} p_i \mathbf{t}_i \quad (14)$$

其中 \mathbf{t}_i 是尾部 t_i 的嵌入。向量 \mathbf{o}_u^1 可以理解为用户 u 的点击历史 V_u 相对于项目 v 的一阶响应。通过公式(4)和(13)中的运算, 用户的兴趣沿着 \mathcal{E}_u^1 中的连接从其历史集 V_u 转移到其一跳相关实体 S_u^1 的集合。

在公式(14)中用向量 \mathbf{o}_u^1 替代向量 \mathbf{v} , 可以实现重复该过程以获得用户 u 的二阶响应向量 \mathbf{o}_u^2 , 对于更高阶波纹集兴趣 $\mathbf{o}_u^3, \mathbf{o}_u^4, \dots, \mathbf{o}_u^H$ 可以通过重复上述操作来得到。而用户 u 的向量是通过将各阶波纹集上兴趣的累加。

$$\mathbf{u} = \mathbf{o}_u^1 + \mathbf{o}_u^2 + \dots + \mathbf{o}_u^H \quad (15)$$

3.4. 预测层

最后我们将获得的用户嵌入 \mathbf{u} 和项目嵌入 \mathbf{v} 进行结合来输出预测的点击概率 \hat{y}_{uv} :

$$\hat{y}_{uv} = \sigma(\mathbf{u}^T \mathbf{v}), \quad (16)$$

其中 $\sigma(x) = \frac{1}{1 + \exp(-x)}$ 是 sigmoid 函数。

4. 实验应用分析

4.1. 实验数据集

在两个数据集: Clothing 和 Cell_Phones 上分析我们的模型, 两个数据集来自于 Amazon 5-core 的 Clothing Shoes and Jewelry 和 Cell Phones and Accessories 两个产品类别。每个数据集包含评论、产品元数据和链接。评论数据由用户项目评分和相应的评论文本组成。产品元数据包括类别和品牌。链接数据包括一起浏览/一起购买。为了构建知识图谱, 我们从 Clothing 和 Cell_phones 数据集中抽取了其出现的所有商品及相关属性, 将每个数据集的商品和它们的属性值映射链接到相应的实体, 每个数据集的项目属性和用户的行为属性视为关系来构建三元组。例如(1182702295, categories, Accessories)其中 1182702295 是商品 ID。总体而言, 用户、物品和其他实体的交互非常稀疏, 对于数据集, 我们随机抽取每个用户交互的 60% 作为训练集, 20% 为验证集其余 20% 作为测试集。数据集的统计数据汇总如表 1 所示。

Table 1. Statistics of the data set**表 1.** 数据集的统计

数据集	用户数	商品数	交互数	关系种类	实体种类
Clothing	39,387	12,101	278,677	5	4
Cell_phone	27,879	10,429	194,439	5	4

Amazon 数据集并不包含我们所需的评论情感极性, 针对此问题我们使用现有的 BERT [15]模型来对评论进行训练、测试。我们发现评论集中含有一些只有分数而没有评论问题的数据, 所以首先将评论中含有空白评论文本进行处理。再从两个数据集中分别挑选出一万条数据人工标注情感极性作为训练数据, 最后将剩余所有数据作为测试数据。将所有数据送入模型训练、测试得到所有评论的情感极性, 部分评论如表 2 所示:

Table 2. Examples of comment data sets**表 2.** 评论数据集实例

评论商品	评论内容	标签	评论分数
0000031887	This is a great tutu and at a really great price. It doesn't look cheap at all. I'm so glad I looked on Amazon and found such an affordable tutu that isn't made poorly. A++	positive	5.0
B00001W0KA	Good costume. Very detailed. I like that the body is a foam so it stands up a little on its own. You might not be able to tell from the picture, but there are shin-guard type pieces for the legs, and gauntlets on the sleeves. Nice touch. It is a little thin overall. You'll definitely want to put something underneath if you're in a colder climate. My only disappointment is that the back of the torso is entirely black. It's ok for us because I bought the inflatable wings, but without the wings, I think it would look weird	positive	4.0
B000072UMJ	These shoes run so big, and the size is very confusing. I emailed to find out if I could just exchange for a smaller size, but they won't do that. I have to send them back, and buy another pair. No thanks.	negative	1.0
B000072UMJ	Do these shoes always run so big? I had to sell them to a friend who has a much bigger foot than I.	neutral	2.0

我们将评论情感划分为 3 类情感等级, 分别是 Negative、Positive、Neutral。这 3 类情感词语的情感等级和对应的情感极性如表 3 所示:

Table 3. Sentiment classification**表 3.** 情感等级分类

情感词语	等级	极性
Negative	0	-1
Neutral	1	0
Positive	2	1

4.2. 实验参数设置

在我们的模型中, Clothing 和 Cell_phone 的跳数 H 都设为 2, 根据实验结果, 更多的跳数几乎不会提高性能, 而且会导致更重的计算开销。完整的超参数设置在表 4 中给出, 其中 d 表示实体和知识图谱的嵌入维度, 表示学习率, 表示知识图谱嵌入权重, 是 L2 正则化的权重。超参数是通过在验证集上优化 Precision 来确定的。

Table 4. Data set hyperparameter settings

表 4. 数据集超参数设置

数据集	d	H	λ_1	λ_2	η
Clothing	64	2	10^{-7}	0.05	0.005
Cell_phone	4	2	10^{-7}	0.02	0.001

4.3. 评价指标

为了评估模型的推荐性能, 我们采用精确率(Precision), 召回率(Recall), 归一化折损累计增益(NDCG)和命中率(HR)作为评价指标。这四个指标得分越高, 推荐性能越好。在最优推荐集中(Top-K), 使用训练好的模型为测试集中的每个用户在测试集商品中, 筛选预测概率最高的前 10 个候选商品作为推荐结果。

精确率是在推荐列表中用户交互的项目占用户推荐列表项目的比例, 而召回率是在推荐列表中用户交互的项目占测试集中用户交互的项目的比例, 精确率和召回率计算公式如下。

$$Precision @ N = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (17)$$

$$Recall @ N = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (18)$$

其中 $R(u)$ 是为用户 u 生成的推荐列表; $T(u)$ 是用户在测试集中交互的项目列表。

NDCG 是通过项目在列表中的位置来评价排序性能, 计算公式为:

$$NDCG @ N = \frac{1}{IDCG_i} \sum_{i=1}^N \frac{2^{rel_i} - 1}{\log_2(i+1)} \quad (19)$$

其中 rel_i 代表项目 i 这个位置上的概率。IDCG 是理想情况下最大的 DCG 值。

$$IDCG @ N = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \quad (20)$$

其中 $|REL|$ 表示结果按概率从小到大的顺序排列, 然后取前 N 个结果组成的集合。在 top-K 推荐中, HR 是一种常用的衡量召回率的指标, 计算公式为:

$$HR @ N = \frac{Number\ of\ Hits @ N}{GT} \quad (21)$$

其中分母是所有的测试集合, 分子表示每个用户推荐列表中属于测试集合个数的总和。

4.4. 对比方法

本文方法和以下方法在 Clothing 和 Cell_Phones 两个数据集进行了对比:

BPR [16]: Bayesian Personalized Ranking (BPR)是基于用户的隐式反馈, 为用户提供物品的推荐, 并且是直接对排序进行优化。贝叶斯个性化排序(Bayesian personalized ranking, BPR)是一种 Pairwise 方法, 并且借鉴了矩阵分解的思路。

RippleNet: 本文方法所依据的框架, 这是一种类似记忆网络的方法在知识图谱上传播用户的偏好推荐。

DKN [17]: Deep Knowledge-aware Network (DKN)是基于内容的深度推荐框架, 用于点击率预测。DKN 的关键部分是一个多通道和单词 - 实体对齐的知识感知卷积神经网络(KCNN), 它融合了新闻的语义层面和知识层面的表示。KCNN 将单词和实体视为多个通道, 并在卷积过程中显式地保持它们之间的对齐关系。此外, 为了解决用户不同兴趣的问题, 作者还在 DKN 中设计了一个注意力模块, 以便动态地聚合当前候选对象的用户历史记录。

RuleRec [18]: RuleRec 提出了一种新的、有效的联合优化框架, 用于从包含商品的知识图谱中归纳出规则, 并基于归纳出的规则进行推荐。该框架由两个模块组成: 规则学习模块和推荐模块。规则学习模块能够在具有不同类型商品关联的知识图谱中导出有用的规则, 推荐模块将这些规则引入到推荐模型中以获得更好的性能。

4.5. 实验结果对比与分析

为了验证提出方法的有效性, 本节设置进行了三个实验: 1) 将我们提出的方法和几种推荐算法进行对比; 2) 设置多跳跳数参数实验, 分析不同跳数对实验结果的影响; 3) 设置波纹集数量参数实验, 分析对于模型所提出参数设置不同对模型性能的影响。

1) 推荐性能

Sentiment-RippleNet-Att 和基线模型在最优推荐集(Top-K)中的性能表现如表 5 所示:

Table 5. The recommended performance of the baseline and our model in the top-10, bold is the best performance of the baseline

表 5. 基线和我们的模型在 top-10 的推荐性能, 加粗是最佳性能

数据集	Clothing				Cell_phone			
	Precision	Recall	NDCG	HR	Precision	Recall	NDCG	HR
BRP	0.196	1.086	0.598	1.801	0.624	3.363	1.892	5.323
DKN	0.106	0.727	0.279	1.012	0.456	3.817	1.603	4.484
RuleRec	0.210	1.150	0.639	1.921	0.674	3.565	1.966	5.669
Ripple Net	0.339	1.428	0.868	3.321	0.695	2.962	1.675	6.631
Sentiment-RippleNet	0.383	1.621	0.976	3.686	0.789	3.491	1.970	7.495
Sentiment-RippleNet-Att	0.403	1.733	1.000	3.866	0.805	3.508	2.022	7.548
提升(%)	5.222	6.909	2.458	5.779	2.028	-8.095	2.640	0.707

实验结果分析和讨论如下: 从表 5 中可以观察到我们的模型在两个数据集上关于 top-10 的指标大部分都优于基线。其中 Sentiment-RippleNet 是没有融入注意力机制的模型, 这表明了我们提出的融入注意力机制的算法能进一步的优化用户向量的表示, 为用户提供更加精准的 top-K 推荐, 能够有效的解决在

构建用户偏好时三元组头尾实体不相关的问题, 这也证明了我们模型的有效性。但从表 5 中也可以看出在 Cell_phone 数据集中 Recall 指标并没有超过最优基线, 分析原因可能在于不同模型中对于知识图谱的处理方式不同而导致 Cell_phones 数据集中 Recall 指标没有提升, 这意味着知识图谱这一辅助信息对于推荐系统的推荐效果十分重要。

2) 跳数实验

我们设置不同跳数进行了实验, 实验结果如表 6 所示。

Table 6. Accuracy results of hop number

表 6. 跳数准确率结果

跳数	1	2	3	4
Clothing	0.367	0.403	0.374	0.321
Cell_phone	0.766	0.805	0.729	0.658

从表 6 中我们可以观察到该实验结果在跳数取 2 时获得最高的准确率, 而且数据集的准确度会随着跳数的增加先升后降, 这是因为一跳路径所包含的语义信息较少, 而多跳路径就很难探索长距离的实体间相关性和依赖性, 会为探索用户偏好带来更多噪声, 因此本文选择多跳路径的长度为 2。

3) 波纹集数量实验

我们改变用户每一跳波纹集的数量, 以进一步对模型进行实验验证。两个数据集上准确率结果如表 7 所示。

Table 7. Accuracy results of the number of ripple sets

表 7. 波纹集数量准确率结果

波纹集数量	2	4	8	16	32	64	128
Clothing	0.301	0.328	0.337	0.369	0.388	0.403	0.374
Cell_phone	0.223	0.497	0.574	0.679	0.805	0.653	0.610

从表 7 中可以看出波纹集数量对于实验结果的影响, 两个数据集对于波纹集大小的结果基本是一致的, 随着波纹集的增大, 数据集的准确率先升后降, Clothing 数据集在 64 准确率达到最高值, Cell_phone 则在 32 达到最大值, 这是因为更大的波纹集可以编码更多来自知识图谱的知识, 但当波纹集过大时会带来噪声, 从而出现先升后降的效果。

4.6. 可解释性

可解释性推荐不仅为用户做出推荐, 为用户做出推荐解释同样重要, 下面给出一个直观的例子来感受一下此方法的可解释性, 如图 2 所示。

图 2 中, 带有颜色的椭圆框为商品实体, 其它是品牌、类别实体。本文随机选取一位用户的三个交互记录, 并在测试集中选取一个标签为 1 的候选商品, 此处选择的候选商品为 B003X6LPRK。对于每个用户的 k-hop 相关实体, 我们计算该实体与候选商品或其 k 阶响应之间的关联概率, 如图 2 中连线的分数所示。为了更清楚的表达我们省略了关系的表示, 从图 2 中我们观察到候选商品和用户相关商品通过不同实体链接起来, 候选商品的推荐理由是: 从用户记录中发现, 用户兴趣分布于商品 120401325X、

B00CJFXEE 和 B00A6KELHM, 而这些实体与商品 B003X6LPRK 高度关联。另外, 我们的模型从用户到候选商品构建了大量链接, 知识图谱中的几个实体也得到了用户的多次关注, 这些可以作为用户未来的潜在兴趣。

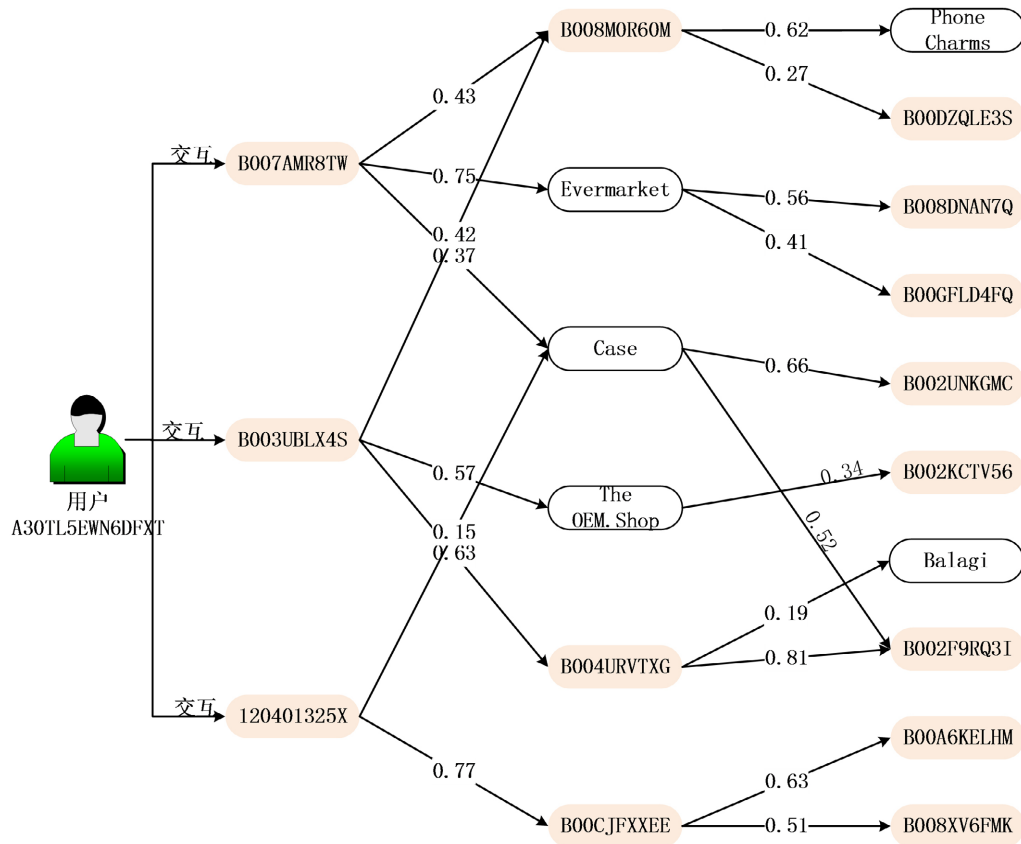


Figure 2. Case display
图 2. 案例展示

5. 结论

本文通过研究发现通过知识图谱中的关系来探索用户偏好时会存在与用户交互记录有不相关实体的问题, 因此本文提出了一种基于知识图谱融入注意力机制商品推荐的可解释性方法。该算法通过知识图谱在构建出用户潜在偏好时, 通过将知识图谱融合注意力机制来嵌入实体更为相关的近邻实体的语义信息, 更新后的实体嵌入中能包含更多信息, 然后通过用户偏好和获取的实体嵌入构建用户向量, 最后通过用户向量、项目向量以及评论情感和评分相结合得到推荐项目并给出推荐项目的解释。最后在 Amazon 5-core 的 Clothing Shoes and Jewelry 和 Cell Phones and Accessories 两个数据集中通过实验证明了实验的有效性。在未来我们将更进一步挖掘出知识图谱之间的高阶关系, 进一步改进模型, 提高模型的能力。

参考文献

- [1] 顾犇. 信息过载问题及其研究[J]. 中国图书馆学报, 2000, 26(5): 42-45+76.
- [2] Zhao, H., Yao, Q., Li, J., Song, Y. and Lee, D.L. (2017) Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, 13-17 August 2017, 635-644. <https://doi.org/10.1145/3097983.3098063>

-
- [3] Yi, F., Jiang, B. and Wu, J. (2020) Heterogeneous Information Network Embedding for Mention Recommendation. *IEEE Access*, **8**, 91394-91404. <https://doi.org/10.1109/ACCESS.2020.2994313>
- [4] Rendle, S. (2010) Factorization Machines. 2010 *IEEE International Conference on Data Mining*, Sydney, 13-17 December 2010, 995-1000. <https://doi.org/10.1109/ICDM.2010.127>
- [5] Zhang, Y., Ai, Q., Chen, X. and Wang, P. (2018) Learning over Knowledge-Base Embeddings for Recommendation. *Algorithms*, **11**, Article No. 137. <https://doi.org/10.3390/a11090137>
- [6] Dadoun, A., Troncy, R., Ratier, O. and Petitti, R. (2019) Location Embeddings for Next Trip Recommendation. *Companion Proceedings of the 2019 World Wide Web Conference*, San Francisco, 13-17 May 2019, 896-903. <https://doi.org/10.1145/3308560.3316535>
- [7] Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., et al. (2018) Ripplenet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Torino, 22-26 October 2018, 417-426. <https://doi.org/10.1145/3269206.3271739>
- [8] Tang, X., Wang, T., Yang, H. and Song, H. (2019) AKUPM: Attention-Enhanced Knowledge-Aware User Preference Model for Recommendation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, 4-8 July 2019, 1891-1899. <https://doi.org/10.1145/3292500.3330705>
- [9] Kipf, T.N. and Welling, M. (2016) Semi-Supervised Classification with Graph Convolutional Networks. arXiv preprint arXiv:1609.02907.
- [10] Wang, H., Zhao, M., Xie, X., Li, W. and Guo, M. (2019) Knowledge Graph Convolutional Networks for Recommender Systems. *The World Wide Web Conference*, San Francisco, 13-17 May 2019, 3307-3313. <https://doi.org/10.1145/3308558.3313417>
- [11] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y. (2017) Graph Attention Networks. arXiv preprint arXiv:1710.10903.
- [12] Wang, X., He, X., Cao, Y., Liu, M. and Chua, T. (2019) Kgat: Knowledge Graph Attention Network for Recommendation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, 4-8 July 2019, 950-958. <https://doi.org/10.1145/3292500.3330989>
- [13] Sha, X., Sun, Z. and Zhang, J. (2021) Hierarchical Attentive Knowledge Graph Embedding for Personalized Recommendation. *Electronic Commerce Research and Applications*, **48**, Article ID: 101071. <https://doi.org/10.1016/j.elerap.2021.101071>
- [14] Nathani, D., Chauhan, J., Sharma, C. and Kaul, M. (2019) Learning Attention-Based Embeddings for Relation Prediction in Knowledge Graphs. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, July 2019, 1710-4723. <https://doi.org/10.18653/v1/P19-1466>
- [15] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.
- [16] Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2012) BPR: Bayesian Personalized Ranking from Implicit Feedback. arXiv preprint arXiv:1205.2618.
- [17] Wang, H., Zhang, F., Xie, X. and Guo, M. (2018) DKN: Deep Knowledge-Aware Network for News Recommendation. *Proceedings of the 2018 World Wide Web Conference*, Lyon, 23-27 April 2018, 1835-1844. <https://doi.org/10.1145/3178876.3186175>
- [18] Ma, W., Zhang, M., Cao, Y., Jin, W., Wang, C. and Liu, Y. (2019) Jointly Learning Explainable Rules for Recommendation with Knowledge Graph. *Proceedings of the 2019 World Wide Web Conference*, San Francisco, 3-17 May 2019, 1210-1221. <https://doi.org/10.1145/3308558.3313607>