

《智能信息处理》课程考试

基于知识图谱的图神经网络推荐模型

吴磊

考核	到课[10]	作业[20]	考试[70]	课程成绩[100]
得分				

2021 年 12 月 18 日

基于知识图谱的图神经网络推荐模型

吴磊

(大连海事大学 信息科学技术学院, 辽宁省大连市 中国 116026)

摘 要 知识图谱 (KG) 蕴含丰富的结构与关联信息, 不仅可以缓解推荐系统中数据稀疏、冷启动等问题, 还可以更准确地进行个性化推荐, 因此提出一种基于知识图谱驱动的端到端图神经网络推荐模型 KGLN。首先使用单层神经网络框架对图中单个节点进行特征融合, 并加入影响因子来改变不同邻居实体的聚合权重; 然后通过迭代的方式将单层扩展到多层, 使实体可以获得丰富的多阶关联实体信息; 最后结合实体特征与用户特征产生预测评分进行推荐。分析并研究了不同聚合方法及影响因子对推荐效果的影响。实验结果表明, 在数据集 MovieLens-1M 以及 Book-Crossing 上与基准方法因子分解库 (LibFM)、深度分解机 (DeepFM)、Wide&Deep、RippleNet 的对比中, KGLN 的曲线下面积 (AUC) 分别提升了 0.3%~5.9% 和 1.1%~8.2%。

关键词 推荐系统; 知识图谱; 图神经网络; 网络特征学习; 个性化推荐; 兴趣挖掘

中图法分类号 TP311

文献标识码 A

Knowledge graph driven recommendation model of graph neural network

Wu lei

(School of Information Science and Technology, Dalian maritime university, Liaoning Dalian 116026 China)

Abstract The abundant structure and association information contained in Knowledge Graph (KG) can not only alleviate the data sparseness and cold-start in the recommender systems, but also make personalized recommendation more accurately. Therefore, a knowledge graph driven end-to-end recommendation model of graph neural network, named KGLN, was proposed. First, a signal-layer neural network framework was used to fuse the features of individual nodes in the graph, then the aggregation weights of different neighbor entities were changed by adding influence factors. Second, the single-layer was extended to multi-layer by iteration, so that the entities were able to obtain abundant multi-order associated entity information. Finally, the obtained features of entities and users were integrated to generate the prediction score for recommendation. The effects of different aggregation methods and influence factors on the recommendation results were analyzed. Experimental results show that on the datasets MovieLen-1M and Book-Crossing, compared with the benchmark methods such as Factorization Machine Library (LibFM), Deep Factorization Machine (DeepFM), Wide&Deep and RippleNet, KGLN obtains an AUC (Area Under ROC (Receiver Operating Characteristic) curve) improvement of 0.3%-5.9% and 1.1%-8.2%, respectively.

Key words recommender system; Knowledge Graph (KG); graph neural network; network feature learning; personalized recommendation; interest mining

1 引言

随着互联网技术的进步, 人们可以获取到大量的在线内容, 比如新闻、电影以及各种商品等。但在线信息的爆炸性增长经常让用户不堪重负, 推荐系统则是一个有效的信息过滤工具, 在减轻信息过载的同时又可以为用户提供良好的体验。推荐系统

通常会搜索并选择较小数量的内容来满足用户的个性化兴趣。协同过滤算法是一种经典的推荐技术, 它将用户以及物品表示为向量形式, 通过特定操作 (内积、神经网络^[1]等) 对它们的交互进行建模; 然而, 协同过滤算法通常都会面临稀疏性以及冷启动等问题。为了缓解基于协同过滤的推荐系统的稀疏性和冷启动问题, 研究人员通常会收集用户和物品的属性, 并设计精巧的算法来利用这些额

外的信息^[2]。

最近的一些研究^{[3]-[6]}都使用了物品属性来进行建模,这些研究成果表明属性并不是孤立的,而是被相互连接起来的,其中一种形式就是知识图谱(Knowledge Graph, KG)。KG是一种典型的异构网络,它的每个节点代表一个实体,这些实体可以是物品也可以是属性,两个存在关系的实体由一条边相连。KG拥有丰富的语义关联信息,可以为推荐系统提供丰富的辅助信息。与没有KG辅助的推荐模型相比,使用KG辅助的推荐模型可以让推荐结果拥有以下三个特征^[7]:1)精确性。丰富的语义相关信息可以发掘更深层次的隐藏关系并且提高结果的准确率。2)多样性。KG中多类别的关系可以扩展用户兴趣,合理地发散推荐结果,提高推荐结果的多样性。3)可解释性。KG可以连接用户的历史点击记录和推荐记录,为推荐系统提供可解释性。虽然KG有着上述优点,但是由于KG天然的高维性以及异构性,如何利用KG进行推荐仍然是一个挑战。目前的一种可行方法就是KG嵌入,将KG中的实体或者关系映射为低维表示向量。然而,一些常用的KG嵌入方法,诸如TransE^[8]和TransR^[9]等,都侧重于对语义关联进行建模,更适用于KG补全和链路预测等图应用,而不是推荐。在图嵌入的方法中,图的建模和特征表示与下游任务是独立的,下游任务获得的结果无法用来优化图的特征表示;而且这种方法缺乏归纳能力,当有新节点加入时,需要重新学习整个图的特征表示。为了解决这些问题,本文提出一种基于知识图谱驱动的端到端神经网络模型(Knowledge Graph driven Learning Network, KGLN),利用KG中的语义相关信息,驱动用户以及物品的表示学习。在端到端模型中,下游任务得到的结果可以用来优化图的特征表示并且具备归纳能力,当有新节点加入时,通过共享参数可以直接获得该节点的特征表示。

KGLN中,在实体表示学习的过程中根据用户偏好来影响邻居节点聚合的权重,这样做有两个优点:1)邻居节点拥有不同的权重,聚合时可以让当前实体获得更加精确的表示,提高推荐的准确率;2)由于用户偏好的影响,在学习过程中可以扩展到更多与用户兴趣相关的内容,可以提高推荐的多样性。

2 相关工作

2.1 图神经网络

图神经网络旨在将神经网络应用于非欧氏空间的数据(如图数据)来进行特征学习。图卷积神经网络(Graph Convolutional neural Network, GCN)由Kipf于2017年提出^[10],它为图结构数据的处理提供了一种崭新的思路,将卷积神经网络(Convolutional Neural Network, CCN)应用到了图数据中。

最近,很多的研究工作都将GCN运用到了推荐系统中:PinSAGE^[11]应用到了Pinterest中的图片与图集这个二分图的推荐;Monti等^[12]和van den Berg等^[13]将推荐系统建模为矩阵分解并分别设计了不同的GCN模型来为二分图中的用户以及项目进行表示学习;Wu等^[14]在用户/物品的结构图上使用GCN去学习用户/物品的表示。这些工作与本文工作的区别是它们的模型都是为同质二分图设计,由于不存在复杂的关系,所以使用GCN建模比较简单。

2.2 基于深度学习的推荐系统

深度学习对数据进行多重非线性变换,可以拟合出较为复杂的预测函数。推荐系统中的核心算法是协同过滤,其目标从机器学习的角度可以看成拟合用户和物品之间的交互函数。因此,近期一系列的工作也将深度学习技术应用于协同过滤的交互函数上。例如,DeepFM^[15]扩展了分解机(Factorization Machine, FM)方法,在FM中引入了一个深度线性模型来拟合特征之间复杂的交互关系;Wide&Deep^[2]的Wide部分采用和FM一样的线性回归模型,Deep部分采用了基于特征表示学习的多层感知机模型,提出了一个应用于视频推荐的深度神经网络推荐模型;NCF(Neural Collaborative Filtering)^[1]用多层感知机替换传统协同过滤中的点积操作;DMF(Deep Matrix Factorization)^[16]类似于DeepFM,在传统矩阵分解模型中引入了一个深度学习模块来提高模型的表达能力。

2.3 使用知识图谱的推荐系统

目前的KG感知推荐系统主要是基于图嵌入的方法,将KG使用KG嵌入算法进行预处理,然后将学习好的实体向量融合到推荐系统中。这种方法将图嵌入以及推荐两个过程分开,导致推荐结果完

全取决于图嵌入算法的优劣,而本文提出的端到端图神经网络推荐模型可以根据推荐结果的反馈来优化图的表示学习,很好地解决了上述问题。同样使用 KG 作为辅助信息的模型还有多任务知识图谱特征学习推荐模型 (Multi-task feature learning for Knowledge graph enhanced Recommendation, MKR)^[17]以及 RippleNet^[18]。MKR 包含推荐系统模块与知识图谱实体特征学习模块,这两个模块通过一个交叉压缩模块进行桥接,交叉压缩模块可以自动学习两个任务中物品和实体的高阶特征交互,通过依次训练来共享特征信息。RippleNet 的核心思想是兴趣传播,将用户历史作为种子,然后通过 KG 中的连接迭代地向外扩展用户的兴趣,以此获得用户向量来进行下行预测任务。两种模型都在真实数据集上拥有出色的效果,本文模型也将与二者进行实验结果对比来验证本文模型的有效性。

3 图神经网络推荐模型

3.1 问题定义

在一个典型的推荐场景中,会有一个包含 M 个用户的集合 $U = \{u_1, u_2, \dots, u_M\}$ 和一个包含 N 个物品的集合 $V = \{v_1, v_2, \dots, v_N\}$ 。用户与物品的交互矩阵为 $Y \in \mathbf{R}^{M \times N}$,它反映了用户的隐式反馈,当 $y_{uv} = 1$ 代表用户 u 与物品 v 存在联系,比如点击、浏览或者购买等,反之 $y_{uv} = 0$ 。此外,还有一个知识图谱 G ,它由实体-关系-实体三元组 (h, r, t) 组成。此处 $h \in E$ 、 $r \in R$ 和 $t \in E$ 表示三元组的头部、关系和尾部, E 和 R 分别是知识图中的实体和关系集。例如三元组 (哈利波特, 作者, JK 罗琳) 表示 JK 罗琳创作了哈利波特这个事实。在许多的推荐场景中,一个物品是与一个实体相关的。比如在书籍推荐中,《哈利波特》就可以作为知识图谱中的一个实体。给定一个用户物品交互矩阵 Y 以及知识图谱 G ,目标是预测用户 u 是否对物品 v 有潜在的兴趣。也就是学习函数 $\hat{y} = \mathcal{F}(u, v | Y, G, \Theta)$, 其中: \hat{y} 表示用户 u 喜欢物品 v 的概率, Θ 为函数 \mathcal{F} 的参数。

3.2 KGLN整体框架

整个 KGLN 的模型框架如图 1 所示,图中展示了阶数为 2 时的整体流程。对于每对用户 u 和物品 v ,使用它们的原始特征作为输入: 用户特征 u 主要参与影响因子的计算以及最后推荐评分的预测; 物品特征 v 在单层 KGLN 中首先采样出邻居样本(图

1 中分别用 e_1^0 以及 e_2^0 表示 v 的邻居实体 e_1 以及 e_2 的原始特征),之后通过用户特征 u 及物品与邻居之间的关系特征 r 计算出影响因子 α 作为权重进行邻居特征聚合,再通过聚合器得到一阶特征 v^1 来更新原始的 v^0 ,这就完成了一次单层 KGLN。为了得到物品的二阶特征 v^2 ,需要先对 e_1^0 以及 e_2^0 分别进行一次单层 KGLN 得到两个邻居的一阶特征 e_1^1 和 e_2^1 ,再对物品的一阶特征进行一次单层 KGLN 得到信息更加丰富的二阶特征 v^2 ,之后与用户特征 u 来预测评分 \hat{y} 。

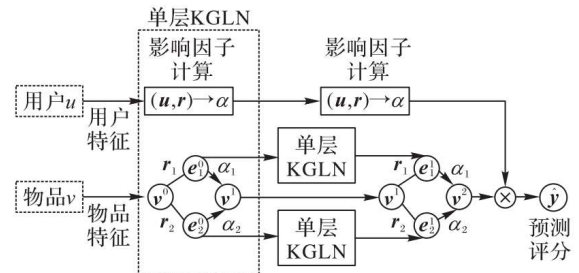


图1 KGLN整体框架

3.3 单层KGLN模型

3.3.1 影响因子

KGLN 的核心思想是通过 KG 中实体间的近邻关系来驱动模型进行用户和物品的表示学习。首先介绍单层的 KGLN,考虑一个用户 u 和物品 v , v 在知识图谱中存在一个实体,用 $N(v)$ 来表示实体 v 在知识图谱中的邻居实体的集合, r_{e_i, e_j} 表示两个实体 e_i 和 e_j 之间的关系。如果用户 u 曾经为物品 v 评分过,在 KGLN 中,会在知识图谱中找到与物品 v 相关的实体,接下来对该实体做邻居节点特征聚合。在邻居节点特征聚合的过程中,加入影响因子来影响聚合的结果,影响因子可以当作是对用户偏好的衡量,这种影响因子类似于注意力机制。具体来说,引入了一个评分函数 $g: \mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}$ 来计算影响因子,也就是用户与关系之间的分值:

$$\alpha_r^u = g(u, r) \quad (1)$$

其中: $u \in \mathbf{R}^d$ 和 $v \in \mathbf{R}^d$ 分别代表用户 u 和关系 r 的特征, d 是特征的维度。一般而言, α_r^u 已经刻画了关系 r 对用户 u 的重要程度,可以用来表示用户的兴趣偏好。比如用户 A 可能会对曾经阅读过的书籍的作者写的另一部作品感兴趣,而用户 B 更多是因为作品的题材而对该作品感兴趣。在 KGLN 中,不仅要计算用户偏好的影响因子,也要计算不同邻居节点对当前中心实体的影响因子:

$$\alpha_e^v = g(v, e) \quad (2)$$

其中： \mathbf{e} 为当前实体 \mathbf{v} 的所有邻居节点的特征。 α_e^v 刻画了所有邻居实体 \mathbf{e} 对当前实体 \mathbf{v} 的影响程度。得到两种影响因子后，分别对其进行归一化处理：

$$\bar{\alpha}_{r,v,e}^u = \frac{\exp(\alpha_{r,e}^u)}{\sum_{e \in N(v)} \exp(\alpha_{r,e}^u)} \quad (3)$$

$$\bar{\alpha}_e^v = \frac{\exp(\alpha_e^v)}{\sum_{e \in N(v)} \exp(\alpha_e^v)} \quad (4)$$

3.3.2 邻居采样

在对每一个节点进行邻居节点实体特征融合的过程前都需要对其邻居进行采样，取得需要进行特征融合的邻居集合。由于在知识图谱中一个实体的邻居数据不会相同，而在一种极端的情况下，其数量可能会相差非常多。为了提高模型的训练效率，对每个实体没有使用它的全部邻居，而是均匀随机地在所有邻居中采样出一个集合 $S(v) = \{e \mid e \sim N(v)\}$ ，采样的邻居个数用 K 控制。图2是 $K=2$ 时的采样情况，即每个节点只会选取邻近自己的两个邻居节点。在 KGLN 中， $S(v)$ 也可以被称为实体 \mathbf{v} 的（单层的）感受野（receptive field），因为实体 \mathbf{v} 的最终特征计算对这些范围敏感（图2中的深色节点）。在均匀随机地采样了 K 个邻居节点后，根据式（1）~（4）计算每个节点的影响因子，然后对物品 \mathbf{v} 的邻居节点进行建模，具体如下：

$$V_{N(v)}^u = \sum_{e \in N(v)} (\bar{\alpha}_{r,e}^u \mathbf{e} + \bar{\alpha}_e^v \mathbf{e}) \quad (5)$$

其中 \mathbf{e} 为实体 \mathbf{v} 邻居节点的特征。影响因子的引入可以让我们根据用户偏好来个性化地聚合邻居节点特征。

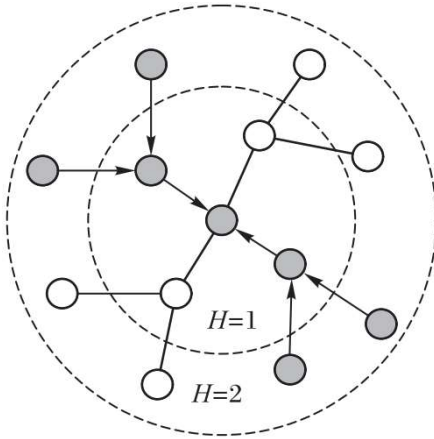


图2 采样完成的邻居节点向内聚合

3.3.3 聚合实体特征

最后一个步骤就是融合当前实体特征 \mathbf{v} 与它的邻居特征 $\mathbf{v}_{N(v)}^u$ ，将获得的结果作为当前实体新的特征表示。使用了三种类型的聚合器：

1) GCN 聚合器 (GCN aggregator)^[10] 直接将两个特征相加，然后使用一个非线性函数得到结果：

$$f_{GCN} = \text{LeakyReLU}(\mathbf{W} \cdot (\mathbf{v} + \mathbf{v}_{N(v)}^u) + b) \quad (6)$$

将激活函数设置为 LeakyReLU。 $\mathbf{W} \in \mathbf{R}^{d \times d}$ 是可训练的权重矩阵，用来提取有用信息进行传播操作。

2) GraphSage 聚合器 (GraphSage aggregator)^[19] 将两个特征拼接再应用一个非线性变换得到结果：

$$f_{GS} = \text{LeakyReLU}(\mathbf{W} \cdot \text{concat}(\mathbf{v}, \mathbf{v}_{N(v)}^u) + b) \quad (7)$$

激活函数为 LeakyReLU， $\mathbf{W} \in \mathbf{R}^{d \times d}$ 是可训练的权重矩阵。

3) 双交互聚合器 (Bi-Interaction aggregator)^[20] 考虑了当前节点特征 \mathbf{v} 与邻居特征 $\mathbf{v}_{N(v)}^u$ 的两种信息交互：

$$f_{BI} = \text{LeakyReLU}(\mathbf{W}_1 \cdot (\mathbf{v} + \mathbf{v}_{N(v)}^u)) + \text{LeakyReLU}(\mathbf{W}_2 \cdot (\mathbf{v} \odot \mathbf{v}_{N(v)}^u)) \quad (8)$$

其中： $\mathbf{W}_1, \mathbf{W}_2 \in \mathbf{R}^{d \times d}$ 都为可训练权重矩阵； \odot 表示向量按元素相乘。与 GCN 聚合器和 GraphSAGE 聚合器不同，双交聚合器模型传播的信息对 \mathbf{v} 与 $\mathbf{v}_{N(v)}^u$ 的相关性比较敏感，它会传递更多类似当前实体节点的信息。

整个聚合邻居实体特征的过程如图3所示，当前节点在 h 层的实体特征为 \mathbf{v}^h ，两个邻居的特征分别为 \mathbf{e}_1 和 \mathbf{e}_2 。首先计算出两个邻居节点的影响因子：属于 \mathbf{e}_1 的用户偏好影响因子 $\bar{\alpha}_{r,v,e_1}^u$ 以及中心实体影响因子 $\bar{\alpha}_{e_1}^v$ ，属于 \mathbf{e}_2 的用户偏好影响因子 $\bar{\alpha}_{r,v,e_2}^u$ ，以及中心实体影响因子 $\bar{\alpha}_{e_2}^v$ 。最后通过式（5）得到 $\mathbf{v}_{N(v)}^u$ ，再与 \mathbf{v}^h 通过聚合器计算得到当前节点在 $h+1$ 层实体特征 \mathbf{v}^{h+1} 。

属于 \mathbf{e}_1 的用户偏好影响因子 $\bar{\alpha}_{r,v,e_1}^u$ 以及中心实体影响因子 $\bar{\alpha}_{e_1}^v$ ，属于 \mathbf{e}_2 的用户偏好影响因子 $\bar{\alpha}_{r,v,e_2}^u$ ，以及中心实体影响因子 $\bar{\alpha}_{e_2}^v$ 。最后通过式（5）得到 $\mathbf{v}_{N(v)}^u$ ，再与 \mathbf{v}^h 通过聚合器计算得到当前节点在 $h+1$ 层实体特征 \mathbf{v}^{h+1} 。

3.4 学习算法

经过一个单层的 KGLN，一个实体节点得到的实体表示依赖于其邻居节点，称其为一阶实体特征。当我们将 KGLN 扩展到多层时，就可以探索用户更深层次的潜在兴趣，这是十分自然合理的，具体的算法如下所述：

1) 对于一个给定的用户-物品对 (u, v) ，首先对 \mathbf{v} 进行邻居采样，经过随机采样之后得到 \mathbf{v} 的 K 个邻居节点，接下来对这 K 个邻居节点进行相同的采样操作。这样，以一种迭代的计算过程得到 \mathbf{v} 的多阶邻居集合。这里用 H 来表示感受野的深度，也就

是迭代的次数,如图2所示为 $H=2$ 时 v 的感受野(图中的深色节点)。

2) 邻居采样完成后,考虑 $H=2$ 时的计算过程。首先计算 v 的一阶实体特征 v^1 (角标代表阶数),这里需要用到 v 的零阶实体特征 v^0 ,也就是原始特征。在经过单层KGLN之后,可以得到融合了邻居节点特征 v 的新实体特征 v^1 。之后用一层KGLN计算 v 的所有邻居的一阶实体特征,用一阶实体特征更新原始特征得到所有的 $e_n^1, e_n \in N(v)$ 。最后再使用一层KGLN更新 v 的实体特征,得到 v 的二阶实体特征 v^2 。整个迭代过程如图4所示,自底向上不断进行邻居节点特征融合。

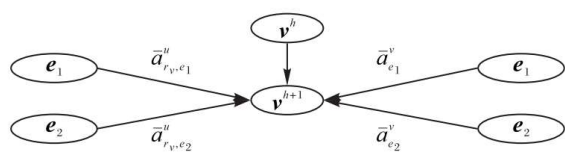


图3 邻居实体特征的聚合过程

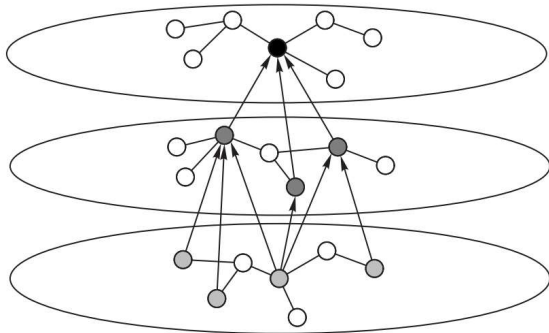


图4 邻居节点特征融合的迭代过程

3) 将所有用户-物品对 (u, v) 执行1)~2)过程后,可以得到最终的 H 阶实体特征,这里记为 v^u 。将 v^u 与用户特征 u 一起输入函数 $f: \mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}$,得到最终的预测概率:

$$\hat{y}_{uv} = f(u, v^u)$$

整个模型损失函数如下:

$$\mathcal{L} = - \sum_{u \in \mathcal{U}} \left(\sum_{v: y_{uv}=1} \phi(y_{uv}, \hat{y}_{uv}) - \sum_{i=1}^{T_u} E_{v_i \sim P(v_i)} \phi(y_{uv_i}, \hat{y}_{wv_i}) \right) + \lambda \| \mathcal{F} \|_2^2 \quad (9)$$

为了计算更有效率,在训练中使用了负采样技术。其中 ϕ 是交叉熵损失函数, T_u 是用户 u 的负样本个数, P 是负样本的分布。在本模型中, T_u 与用户正样本的个数相同, P 则为均匀分布。最后一项为L2正则项,包括 $\mathcal{F}(u, v | Y, G, \Theta)$ 中所有的参数。

4 实验与分析

4.1 数据集

实验部分选择了两种数据集:MovieLens-1M以及Book-Crossing。MovieLens-1M是电影推荐中的一个广泛使用的基准数据集,它包含了在MovieLens网站上用户对电影的显式的从1到5的评分;Book-Crossing是Book-Crossing图书社区的用户对图书进行0~10评分的数据。

将两种数据集进行处理:在MovisLens-1M中,如果用户对电影做出了4分以上的评分,则将用户与该电影的交互置为1,反之为0;对Book-Crossing也做相同的处理,评分范围规定为0分以上,即只要用户对该图书做了评分则将交互置为1。

知识图谱部分使用了MKR中构建的知识图谱,该知识图谱使用微软Satori构造。对于MovisLens-1M和Book-Crossing,首先从整个知识图谱中选择一个子图,其中关系的名字包括电影以及图书。从初步筛出的子图中比照了所有数据集中的电影名或者图书名,来进行物品与实体的配对,最后再删掉无法找到实体的数据集数据,得到最终的知识图谱与交互矩阵。两种数据集的基本统计信息如表1所示。

表1 两种数据集的统计结果

数据集	用户数	物品数	交互数	实体数	关系数
MovieLens-1M	6 036	2 347	753 772	7 008	7
Book-Crossing	19 860	19 967	170 746	25 787	18

图5为使用MovieLens-1M构造的部分知识图谱展示,该部分图谱以“Forrest Gump”为中心节点。为了便于展示,在关系前加上了该关系三元组中头节点的属性。在MovieLens-1M数据集中,作为中心实体的都是电影,每个电影通过不同的关系连接不同的属性,KGLN将会用这些丰富的属性信息对每个电影实体进行特征学习。

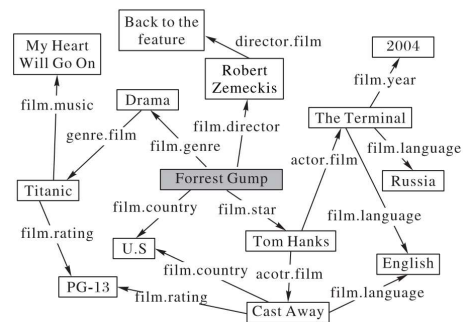


图5 知识图谱样例

4.2 基准方法

在 KGLN 的实验中, 使用了如下的基准方法:

LibFM^[21]是一个用于点击率 (Click-Through-Rate) 预测的基于特征分解的模型, 使用用户与物品的原始特征作为输入。

Wide&Deep^[2]是一个基于深度学习推荐算法, 是一种融合浅层 (Wide) 模型和深层 (Deep) 模型进行联合训练的框架。Wide 部分的输入需要特征工程得到的二阶交叉特征, 在本实验过程中使用原始稀疏特征作为输入。

DeepFM^[15]是 Wide&Deep 的改进, 通过浅层模型和深层模型来训练得到。它包含两个部分: 神经网络模块与因子分解机模块, 分别负责低阶特征的提取和高阶特征的提取, 两部分输入与 LibFM 相同均为原始特征。

MKR^[17]包含推荐系统模块以及知识图谱特征学习模块, 使用用户以及物品的原始特征作为推荐系统模块的输入, 数据集上的知识图谱作为知识图谱特征学习模块的输入。

RippleNet^[18]是混合模型的代表, 通过在知识图谱上传播用户偏好的记忆网络推荐模型, 输入同样为用户和物品的原始特征以及知识图谱。

4.3 实验准备

KGLN 的实验准备如下: 在 KGLN 中, 将 MovieLens-1M 的跳数 H 设置为 2, Book-Crossing 的跳数 H 设置为 1, 其他超参数的设置如表 2 所示, 其中: d 为特征维度; ε 为学习率; 模型中 g 设置为内积函数, 非最后一层聚合器的激活函数为 LeakyReLU, 最后一层聚合器的激活函数为 tanh, 所有超参数在验证集上通过优化 AUC 值得到。

表 2 KGLN 的超参数设置

数据集	d	K	H	λ	ε
MovieLens-1M	16	4	2	1×10^{-5}	0.01
BookCrossing	8	8	1	2×10^{-6}	0.005

基准方法的超参数设置如下: 对于 LibFM, 其维度为 $\{1, 1, 8\}$, 学习轮数为 50; 对于 DeepFM, 两个数据集上的特征维度都为 $d=8$, 模型维度为 $\{16, 16, 16\}$; 对于 RippleNet, 其特征维度与学习率设置为与 KGLN 相同。Wide&Deep 以及 MKR 的超参数选择和原始论文或代码默认值一致。

在实验中, 数据集的训练集、验证集和测试集的比例为 6:2:2, 最终实验结果均为 5 次实验结果取平均得到的。实验情景为 CTR 预测, 通过受试

者工作特征曲线 (Receiver Operating Characteristic, ROC) 下面积 (Area Under Curve of ROC, AUC) 以及 F1 值衡量模型性能。

4.4 实验结果

4.4.1 与基准方法的比较

KGLN 与其他基准方法的结果如表 3 所示, 从中可以看出:

LibFM 输入为数据集的原始特征, 通过 FM 来学习一阶特征和二阶交叉特征, 得到低维向量表示; DeepFM 和 Wide&Deep 都是用深度神经网络来得到向量表示, 区别是前者使用 FM 来负责一阶特征以及二阶交叉特征, 后者使用 Wide 部分的逻辑回归模型记忆二阶交叉特征。这三种方法都只用了数据集本身的特征信息, 没有任何辅助信息, 在评价指标上要低于所有使用知识图谱作为辅助信息的方法, 这表明引入知识图谱是有效的。

MKR 在 Book-Crossing 上的结果要优于所有基准方法, 因为 Book-Crossing 数据集相对来说比较稀疏, 而 MKR 中的交叉共享单元可以让图学习模块以及推荐模块共享对方的额外信息, 这种交互学习方式一定程度上弥补了数据集稀疏的问题。KGLN 在 Book-Crossing 上的效果略低于 MKR, 也是由于数据集的稀疏性, 在融合邻居节点特征时容易引入噪声。而在 MovieLens-1M 这种相对稠密的数据集上, KGLN 的表现要优于所有基准方法, 这也体现了越丰富的知识图谱对推荐效果起的作用越大。

RippleNet 与 KGLN 的结果非常相近, 注意到 RippleNet 也用到了多跳关联实体信息, 这表明在使用知识图谱辅助时对局部结构以及邻居节点关系进行建模十分重要。

表 3 中 KGLN-BI、KGLN-GS 和 KGLN-GCN 表示 KGLN 中分别使用不同聚合器 (见 2.3.3 节) 得到的结果。可以发现: 双向交互聚合器一般而言表现最好, 是因为双交互聚合器可以更多地保留自身的特征, 并融合更多类似于自身信息而来自邻居节点的特征; GCN 与 GraphSAGE 聚合器效果略是由于在聚合过程中结构信息的比重与节点自身信息的比重一致而造成的, 这表明在建模过程中应该更多考虑节点本身的信息。

为了验证影响因子的效果, 对比了使用影响因子与未使用影响因子 (即邻居结构建模时使用所有邻居节点特征的均值作为邻居特征, 即 $\mathbf{v}_{N(v)}^u =$

$\frac{1}{K} \sum_{e \in N(v)} e$ 的 AUC 值, 如表 4 所示。未使用影响因

子的结果要比使用影响因子的效果差, 这说明注意力机制确实对推荐结果有提升作用。

表 3 KGLN 与其他基准方法在 MovieLens-1M 和 Book-Crossing 数据集上的 AUC 与 F1 比较

模型	MovieLens-1M		Book-Crossing	
	AUC	F1	AUC	F1
DeepFM	0.872	0.794	0.668	0.602
LibFM	0.882	0.812	0.691	0.618
Wide&Deep	0.892	0.816	0.712	0.624
MKR	0.912	0.839	0.735	0.705
RippleNet	0.921	0.844	0.715	0.650
KGLN-BI	0.924	0.849	0.723	0.682
KGLN-GS	0.913	0.832	0.677	0.633
KGLN-GCN	0.918	0.837	0.698	0.647

表 4 影响因子在不同聚合器上的效果 (AUC 值)

聚合器	MovieLens-1M		Book-Crossing	
	使用	未使用	使用	未使用
	影响因子	影响因子	影响因子	影响因子
GCN	0.918	0.891	0.698	0.683
GraphSAGE	0.913	0.906	0.677	0.654
Bi-Interaction	0.924	0.915	0.723	0.702

表 5 不同感受野深度 H 下的实验结果

感受野深度 H	MovieLens-1M		Book-Crossing	
	AUC	F1	AUC	F1
1	0.912	0.836	0.723	0.682
2	0.924	0.849	0.664	0.608
3	0.885	0.809	0.509	0.503

4.4.2 感受野深度的影响

表 5 给出了感受野深度 H 从 1 变化到 3 时对 KGLN 结果的影响, 可以看出 KGLN 对 H 的改变比较敏感。感受野会随着深度的增加一个指数式级扩大^[22], 一个较大的 H 在模型中会让实体节点对以其为中心的大范围实体敏感, 因为该实体的最终特征计算依赖于这些实体, 而这些大范围的实体中会存在很多对中心实体特征表示毫无用处的实体, 这就会引入大量的噪声, 导致模型结果下降。

Book-Crossing 数据集在 H=3 时得到了相对较差的结果, 这表明模型在融合三阶实体特征时已经引入了大量的噪声。在现实情况中, 过深的探索用户兴趣也是没有必要的。比如在电影推荐的场景中, 用户 A 喜欢电影 B 是因为该电影的演员 C, 那么可以通过演员 C 这个实体去发掘该演员演过的电影, 在这些电影实体中融合更多演员 C 这个实体的信息可以达到挖掘用户兴趣的目的。可见, H 为 1 或 2 在真实场景中已经可以满足需求。

5 结论

本文提出了图神经网络推荐模型, 在知识图谱特征学习的过程中引入了影响因子, 通过改变邻居节点特征的权重来进行聚合, 不仅学习到了知识图谱的结构和语义信息, 还可以挖掘用户的个性化兴趣以及潜在爱好, 提高了推荐结果的多样性和丰富性。在两种公开数据集 MovieLens-1M 与 Book-Crossing 上进行实验, 分析研究了不同聚合器及影响因子对推荐结果的影响。使用 AUC 值与 F1 值作为评价标准, 将 KGLN 与经典模型 DeepFM、LibFM、Wide&Deep、RippleNet 进行对比, 结果表明 KGLN 在推荐效果上都有所提升, 验证了 KGLN 的可行性和有效性。KGLN 所考虑的知识图谱是静态的, 而现实生活中大部分情况知识图谱都会随着时间而变化; 不仅如此, 用户的喜好也会随着时间而变化, 如何刻画这种随时间演变的情况, 也是未来研究的方向。

参考文献

- [1] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering [C] // Proceedings of the 26th International Conference on World Wide Web. Republic and Canton of Geneva: International World Wide Web Conferences Steering Committee, 2017: 173-182.
- [2] CHENG H T, KOC L, HARMSSEN J, et al. Wide & deep learning for recommender systems [C] // Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. New York: ACM, 2016: 7-10.
- [3] HUANG J, ZHAO W X, DOU H, et al. Improving sequential recommendation with knowledge-enhanced memory networks [C] // Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2018: 505-514.
- [4] WANG H, ZHANG F, XIE X, et al. DKN: deep knowledge-aware network for news recommendation [C] // Proceedings of the 2018 World Wide Web Conference. Republic and Canton of Geneva: International World Wide Web Conferences Steering Committee, 2018: 1835-1844.
- [5] ZHANG F, YUAN N J, LIAN D, et al. Collaborative knowledge base embedding for recommender systems [C] // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 353-362.
- [6] YU X, REN X, SUN Y, et al. Personalized entity recommendation: a heterogeneous information network approach [C] // Proceedings of the 7th ACM International Conference on Web Search and Data Mining. New York: ACM, 2014: 283-292.
- [7] WANG Q, MAO Z, WANG B, et al. Knowledge graph embedding:

- a survey of approaches and applications [J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29 (12): 2724-2743.
- [8] BORDES, ANTOINE U, NICOLAS G, et al. Translating embeddings for modeling multi-relational data [C] // Proceedings of the 26th International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2013: 2787-2795.
- [9] LIN Y, LIU Z, SUN M, et al. Learning entity and relation embeddings for knowledge graph completion [C] // Proceedings of the 29th AAAI Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2015: 2181-2187.
- [10] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks [EB/OL]. [2020-06-20]. <https://arxiv.org/pdf/1609.02907.pdf>.
- [11] YING R, HE R, CHEN K, et al. Graph convolutional neural networks for web-scale recommender systems [C] // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2018: 974-983.
- [12] MONTI F, BRONSTEIN M M, BRESSON X. Geometric matrix completion with recurrent multi-graph neural networks [C] // Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2017: 3700-3710.
- [13] VAN DEN BERG R, KIPF T N, WELING M. Graph convolutional matrix completion [EB/OL]. [2020-06-20]. <https://arxiv.org/pdf/1706.02263.pdf>.
- [14] WU Y, LIU H, YANG Y. Graph convolutional matrix completion for bipartite edge prediction [EB/OL]. [2020-06-20]. <http://nyc.lti.cs.cmu.edu/yiming/Publications/wu-kdir18.pdf>.
- [15] GUO H, TANG R, YE Y, et al. DeepFM: a factorization machine based neural network for CTR prediction [C] // Proceedings of the 26th International Joint Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2017: 1725-1731.
- [16] XUE H, DAI X, ZHANG J, et al. Deep matrix factorization models for recommender systems [C] // Proceedings of the 26th International Joint Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2017: 3203-3209.
- [17] WANG H, ZHANG F, ZHAO M, et al. Multi-task feature learning for knowledge graph enhanced recommendation [C] // Proceedings of the 2019 World Wide Web Conference. New York: ACM, 2019: 2000-2010.
- [18] WANG H, ZHANG F, WANG J, et al. RippleNet: propagating user preferences on the knowledge graph for recommender systems [C] // Proceedings of the 27th ACM International Conference on Information and Knowledge Management. New York: ACM, 2018: 417-426.
- [19] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs [C] // Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2017: 1025-1035.
- [20] WANG X, HE X, CAO Y, et al. KGAT: knowledge graph attention network for recommendation [C] // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2019: 950-958.
- [21] RENDLE S. Factorization machines with libFM [J]. ACM Transactions on Intelligent Systems and Technology, 2012, 3 (3): No. 57.
- [22] WU Z, PAN S, CHEN F, et al. A comprehensive survey on graph neural networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32 (1): 4-24.