

融合自注意力机制和知识图谱的多任务推荐模型

李宇轩¹ 郑博^{1,2} 吴茂念^{1,2} 孙悦¹ 朱绍军^{1,2*}

¹(湖州师范学院信息工程学院 浙江 湖州 313000)

²(浙江省现代农业资源智慧管理与应用研究重点实验室(湖州师范学院) 浙江 湖州 313000)

摘要 借助知识图谱提供辅助信息以提升推荐系统性能愈加受到研究者的关注。针对基于知识图谱的推荐算法用户表示较为单一,无法充分挖掘隐藏信息的问题,提出一种融合自注意力机制和知识图谱的推荐模型KSMR。通过自注意力捕获用户交互序列的上下文信息,得到融合兴趣转移的用户向量,采用文本卷积网络实现特征修正与再提取;交替训练推荐任务和知识图谱嵌入任务,达到协同优化的目的。在真实数据集 MovieLens-1M 与 Last.FM 上的实验结果表明,模型的点击率预测(CTR)性能相较于对比算法均有明显提升。

关键词 推荐系统 知识图谱 自注意力机制 兴趣转移 交替训练

中图分类号 TP391

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2025.03.020

MULTI-TASK RECOMMENDATION MODEL COMBINING SELF-ATTENTION MECHANISM AND KNOWLEDGE GRAPH

Li Yuxuan¹ Zheng Bo^{1,2} Wu Maonian^{1,2} Sun Yue¹ Zhu Shaojun^{1,2*}

¹(School of Information Engineering, Huzhou University, Huzhou 313000, Zhejiang, China)

²(Key Laboratory of Intelligent Management & Application Research of Modern Agricultural Resources(Huzhou University),
Huzhou 313000, Zhejiang, China)

Abstract Researchers have got increasingly attention to obtain auxiliary information with the help of knowledge graph. Aimed at the problem that recommendation algorithms based on knowledge graph have single user representation and cannot fully mine hidden information, a recommendation model combining self-attention mechanism and knowledge graph (KSMR) is proposed. The context information of user interaction sequence was captured by self-attention mechanism to obtain the user vector fused with interest transfer, and the feature correction and re-extraction were realized by text CNNs. Alternating training was used to combine the knowledge graph embedding task and recommendation task to achieve the purpose of collaborative optimization. Experimental results on real datasets MovieLens-1M and Last.FM show that, the CTR (Click Through Rate) prediction performance of the model has obvious advantages over the comparison algorithms.

Keywords Recommendation system Knowledge graph Self-attention mechanism Interest transfer Alternate training

0 引言

当今科技的蓬勃进步,使得互联网中的信息呈现爆炸般增长,用户如果逐一尝试所有选择,付出的代价可能会过于昂贵^[1];同时,用户满足自身物质与精神世

界的需求也愈加丰富。因此,推荐系统被开发,为使用者独有的偏好提供个性化的推荐功能,处理信息冗杂的问题^[2]。

协同过滤(Collaborative Filtering, CF)推荐以及基于内容的推荐是较传统的推荐方法,它们都面临着数据稀疏与用户、项目冷启动的问题。因此,借助有效的

辅助信息进行推荐系统建模,解决传统推荐方法存在的问题,成为目前学术界的探索方向之一,知识图谱就是其中一种。

目前,基于知识图谱的推荐算法主要分为基于嵌入的方法、基于路径的方法与混合方法^[3]。其中,基于嵌入的方法通过获得用户及项目的低阶嵌入向量来计算相似性,被广泛应用与研究。Wang 等^[4-5]在基于嵌入方法的领域先后提出了新闻推荐中的深度知识感知网络模型(Deep Knowledge-aware Network, DKN)和面向知识图谱增强推荐的多任务特征学习模型(Multi-task learning for Knowledge graph enhanced Recommendation, MKR)。在 DKN 中,每个新闻实体在知识图谱中的直接邻居被充当补充信息,与实体进行聚合。但这种方法在知识图谱中获取的信息较为有限,缺乏深层次隐藏信息的传播;而在 MKR 中,模型将知识图谱领域旨在解决知识图谱不完整性的链接预测任务加入其中,与推荐任务交替训练,改进了 DKN 中只聚合直接邻居的问题。但模型依旧存在不足:项目向量可以从知识图谱中获取辅助信息,而用户向量的训练只通过若干条交互的项目列表单独进行,由此训练出的用户向量表示较为单一,缺乏多元化信息。

序列推荐系统(Sequential Recommender Systems, SRS)则试图理解和建模用户行为以及用户偏好和项目受欢迎程度随时间的演变^[6]。目前,依赖循环神经网络的序列推荐模型中,一种名为“自注意力”的机制代替了常被使用的卷积或递归模块,能够提炼长句中词与词之间的语义联系。Kang 等^[7]提出序列推荐模型 SASRec(Self-attentive Sequential Recommendation),基于自注意力机制迭代地为序列中的所有项目分配权重,取得了优异的性能表现。因此,基于序列推荐的思想处理用户交互的项目信息,得到项目权重来侧面代替用户画像,能够为用户向量的表示提供先验信息。

捕获序列信息更依靠细粒度的特征向量表示,因此需要维度层面的提升。而 MKR 中采用若干个多层感知机(Multilayer Perceptron, MLP)执行用户和关系向量的特征提取任务,其在面对低维数据时简单高效,在面对高维数据时需要堆叠隐层结构才能达到可观的效果,但同样面临过拟合的风险,欠缺了对空间信息的提取能力。文本卷积网络(Text Convolutional Neural Networks, Text-CNN)^[8]是卷积神经网络的变体,其使用不同尺寸的单层卷积核提取特征,在面对文本信息时继承了图像处理的优异性能,能够更好地承担对于较高维度向量的特征提取任务。

综上所述,本文针对已有模型用户表示单一、特征提取不够充分的问题,将序列推荐与多任务推荐相结

合,优化特征提取环节,提出一种融合自注意力机制与知识图谱的多任务推荐模型,命名为 KSMR,旨在从用户交互的项目序列中捕获用户兴趣转移以实现用户表示的多元化,有效地提高了模型的性能。本文的主要贡献如下:

(1) 将基于自注意力机制的序列推荐和基于知识图谱的多任务推荐融合到统一的框架中,交替训练推荐模块与知识图谱嵌入模块,实现多任务的协同优化。

(2) 通过自注意力机制捕获用户交互序列的兴趣转移,将其融入用户的向量表示,使用户向量包含更多的潜在信息。

(3) 优化特征提取环节,用文本卷积网络代替多层感知机,充分考虑高维向量的空间信息,使特征提取丰富、高效,模型的预测性能得到有效提升。

1 相关工作

目前基于知识图谱嵌入的推荐算法大多使用项目图为项目表示提供辅助信息,但融合方式各有异同。文献[9]提出了协同知识库嵌入模型(Collaborative Knowledge Base Embedding, CKE),将项目的文本、结构和视觉知识共同融合成项目表示的隐向量,再通过协同过滤计算用户与项目的相关性,验证了结构化知识对推荐的增强效果,但其对特征的筛选过程过于繁琐。之后,文献[4]提出了深度知识感知网络(Deep Knowledge-aware Network, DKN)应用于新闻推荐,通过知识感知卷积神经网络进行特征提取,解决了手动捕获特征的问题;同时,将注意力机制引入模型以捕捉不同权重的用户兴趣,但该模型缺乏深层次隐藏信息的传播。为通过知识图谱传播深层偏好,文献[10]提出了波纹网络(RippleNet),将辅助信息融入端到端的推荐模型中,用户的潜在兴趣跟随知识图谱中的路径不断传播,最终形成备选目标的偏好分布。但模型并没有根据传播的距离远近进行合适的权重分配,对特征的把握不够充分。

多任务学习的出现解决了上述模型存在的弊端,同样成为该领域的研究方向之一。知识图谱链接预测任务被应用到模型中,以辅助推荐任务完成,项目特征被自动学习,二者因此实现协同优化。基于此思想,文献[11]提出了基于知识图谱增强翻译的用户偏好模型(Knowledge-enhanced Translation-based User Preference Model, KTUP),建模用户、项目间的隐式关系,这一关系与知识图谱中项目间的关系产生对应,联合训练。之后,文献[5]提出 MKR,建模知识图谱嵌入任务以辅助推荐任务,知识图谱中的项目信息与用户交互

的项目实现信息共享。在此基础上,文献[12]提出了基于知识图谱的增强多模态推荐(SI-MKR),针对已有模型中存在的忽略数据多样性问题,将用户与项目的文本类型属性与多值类型属性进行特征提取及嵌入;文献[13]提出了基于知识图谱的面向对象多任务推荐算法(KGeRec),通过嵌入边信息及融合知识图谱深层语义匹配方法(Deep Semantic Matching, DSM)来改善推荐效果。但是,以上方法都缺乏时序特征的影响。相反,文献[14]提出了融合知识图谱和短期偏好的推荐算法(MKASR),其通过门控循环单元(Gated Recurrent Unit, GRU)提取用户的短期偏好,还使用 Ripple-Net 建模用户与知识图谱的关系,但复杂的混合模型最终导致 CTR 预测结果的提升较为有限。在以上工作的基础上,聚焦于丰富用户端信息,本文通过自注意力机制捕获用户兴趣转移,通过文本卷积网络优化特征提取环节,以求解决以上问题。

2 模型设计

2.1 问题描述

在典型的推荐场景中,存在一组 M 个用户 $U = \{u_1, u_2, \dots, u_M\}$, 以及一组 N 个项目 $V = \{v_1, v_2, \dots, v_N\}$ 。根据用户的隐式反馈(如点击、观看、浏览等行为)可以定义用户-项目交互矩阵 $Y = \{y_{uv} \mid u \in U, v \in V\}$, 其中:

$$y_{uv} = \begin{cases} 1 & \text{存在交互} \\ 0 & \text{不存在交互} \end{cases} \quad (1)$$

同时,将交互矩阵看作用户的动作序列 $s^u = (s_1^u, s_2^u, \dots, s_{|s_{u1}^u}|}^u)$ 。在训练过程中的时间 t , 模型根据先前的 t 个项目预测下一个项目, 此时可以将模型的输入看作 $(S_1^u, S_2^u, \dots, S_{|s_{u1}^u}|}^u)$, 并在兴趣转移提取模块的末尾输出等长度的错位版本 $(S_2^u, S_3^u, \dots, S_{|s_{u1}^u}|}^u)$ 。此向量序列将再次作为用户向量输入推荐模块。此外,还有一个知识图谱 G , 它由大量的三元组 (h, r, t) 构成, 其中 h, r, t 分别表示知识三元组的头实体、关系实体和尾实体。

本模型的任务为通过用户对项目的交互、评分记录、知识图谱 G 的三元组集, 对用户未曾交互过的项目 v 进行潜在兴趣的预测; 同时训练预测函数 $\hat{y}_{uv} = F(u, v \mid \theta, Y, G)$ 的参数, 其中: \hat{y}_{uv} 表示用户 u 参与项目 v 的概率, θ 是函数 F 的模型参数。

2.2 模型框架

本文依托序列推荐模型 SASRec 和多任务推荐模型 MKR, 并进行适当改进, 设计融合自注意力和知识

图谱的多任务推荐模型 KSMR。模型结构如图 1 所示, 主要由 4 个部分组成: 兴趣转移提取模块、推荐模块、知识图谱嵌入预测模块和交叉压缩单元。

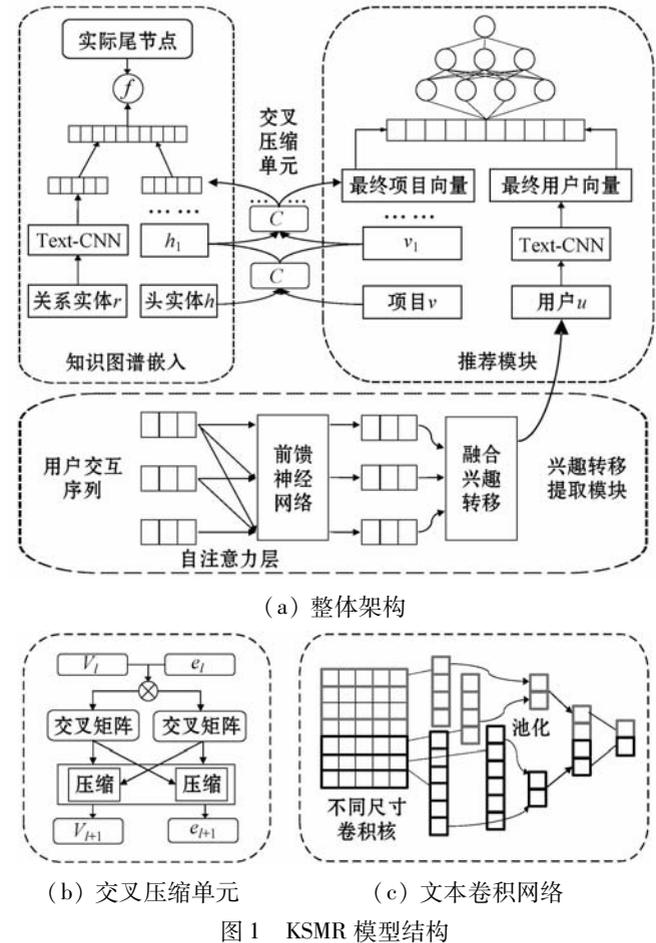


图1 KSMR 模型结构

(1) 兴趣转移提取模块以用户-项目交互序列作为输入, 经过嵌入层、自注意力层、前馈神经网络及预测层后, 得到包含兴趣转移的用户向量作为输出。

(2) 推荐模块输入用户和项目, 使用文本卷积网络和交叉压缩单元分别从用户和项目中提取特征, 再将提取的特征传输到 MLP 中输出预测概率。

(3) 与推荐模块相似, 知识图谱嵌入模块同样依靠交叉压缩单元和文本卷积网络分别从头实体和关系实体中提取特征, 计算相关性分数后与真实的尾实体进行对比学习。

(4) 由交叉压缩单元实现推荐模块与知识图谱嵌入模块的交互, 推荐模块中的项目和知识图谱嵌入模块中的头实体由此实现特征的共享, 特征表示得到丰富。

2.3 融合兴趣转移的用户表示建模

将训练序列 $(S_1^u, S_2^u, \dots, S_{|s_{u1}^u}|}^u)$ 变换为固定长度的序列 (s_1, s_2, \dots, s_n) , 其中 n 为自定义的预期处理长度。如果序列长度大于 n , 就只取最近的 n 个动作; 如果序列长度小于 n , 则向左侧重复添加填充项(常数 0

向量),直到长度为 n 。嵌入矩阵 $\mathbf{M} \in \mathbf{R}^{l \times d_1}$,其中 d_1 为潜在维数。同时,由于自注意模块中没有传递位置信息的单元,因此要将位置学习参数 $\mathbf{P} \in \mathbf{R}^{n \times d_1}$ 注入到输入嵌入中:

$$\mathbf{E} = \begin{bmatrix} \mathbf{M}_{s_1} + \mathbf{P}_1 \\ \mathbf{M}_{s_2} + \mathbf{P}_2 \\ \vdots \\ \mathbf{M}_{s_n} + \mathbf{P}_n \end{bmatrix} \quad (2)$$

注意力计算式为:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \quad (3)$$

$$\mathbf{S} = \mathbf{W}\mathbf{A}(\mathbf{E}) = \text{Attention}(\mathbf{E}\mathbf{W}^Q, \mathbf{E}\mathbf{W}^K, \mathbf{E}\mathbf{W}^V) \quad (4)$$

式中: $\mathbf{W}\mathbf{A}(\mathbf{E})$ 分别表示投影矩阵、Attention、序列的初始嵌入矩阵; \mathbf{Q} 、 \mathbf{K} 和 \mathbf{V} 分别表示 queries(查询)、keys(键)和 values(值),注意力层计算所有值的加权和,query i 与 value j 之间的权重与 query i 与 key j 之间的交互有关。设置比例因子 \sqrt{d} ,避免维度较高时内积的值过大。在本模型中,嵌入矩阵将与初始化的三个矩阵进行乘积运算,并将它们提供给注意力层。其中投影矩阵 $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbf{R}^{d_1 \times d_1}$,其能够使模型更具灵活性,对非对称交互具有一定的学习能力。

不同于类似机器翻译的自然语言处理任务,推荐系统中模型对还未交互过的项目不具有已知性。在预测第 $t+1$ 个项目时,模型应该只考虑前 t 个项目,该项的输出不包含后续项的嵌入。因此,模型进行了局部修改,禁止了 \mathbf{Q}_{ij} 和 \mathbf{K}_j 之间的所有链接。之后,对所有 \mathbf{S}_i 采用两层点式前馈网络:

$$\mathbf{F}_i = \text{FFN}(\mathbf{S}_i) = \text{ReLU}(\mathbf{S}_i\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\mathbf{W}^{(2)} + \mathbf{b}^{(2)} \quad (5)$$

式中: $\mathbf{W}^{(1)}$ 、 $\mathbf{W}^{(2)}$ 是 $d_1 \times d_1$ 维矩阵, $\mathbf{b}^{(1)}$ 、 $\mathbf{b}^{(2)}$ 是 d_1 维向量。最终计算项目之间的相关性 $\mathbf{r}_{i,t}$:

$$\mathbf{r}_{i,t} = \mathbf{F}_i\mathbf{N}_t^T \quad (6)$$

式中: $\mathbf{r}_{i,t}$ 是前 t 个项目与各自的下一个项目之间的相关性, $\mathbf{N} \in \mathbf{R}^{l \times d_1}$ 为项目嵌入矩阵。高交互分数代表高相关性,通过分数生成排名来进行模型的训练,并将训练完成后的序列向量取出作为下一阶段用户向量的输入。

2.4 交叉压缩单元实现特征交互与共享

推荐模块与知识图谱嵌入模块可看作近似对称的结构,兴趣转移提取模块中得到的用户向量与原始的物品、头节点与关系节点的嵌入向量会作为此部分的输入。用户向量 \mathbf{u} 与关系向量 \mathbf{r} 通过文本卷积神经网络进行特征提取:

$$\mathbf{u}_L = T(\mathbf{u}) \quad (7)$$

$$\mathbf{r}_L = T(\mathbf{r}) \quad (8)$$

式中: $T(\cdot)$ 为卷积操作。 \mathbf{u} 与 \mathbf{r} 在被输入前需进行维度的扩展 $d_1 = s \times d$,用 d 表示词向量的维数, s 表示句子的长度,便可将一个句子看作尺寸为 $s \times d$ 的“图像”,用滤波器对其进行卷积操作,得到输出序列 \mathbf{C}_i :

$$\mathbf{C}_i = f(\mathbf{w} \cdot \mathbf{A}[i:i+h-1] + \mathbf{b}) \quad (9)$$

式中: $i = 1, 2, \dots, s-h+1$, \cdot 为子矩阵与滤波器之间的点积,滤波器 \mathbf{w} 是宽为 d 、高为 h 的矩阵,之后再为每个 \mathbf{O}_i 添加偏置项 $\mathbf{b} \in \mathbf{R}$ 及激活函数 f 实现特征映射。对得到的特征进行最大池化后得到某一尺寸滤波器对应的特征点,采用多个大小相异的滤波器分别实施卷积操作,再将所有特征点进行拼接,最后通过全连接层实现与之后模型向量维度的匹配。

项目向量 \mathbf{v} 与头向量 \mathbf{h} 则通过交叉压缩单元实现特征交互与共享,首先将二者组合为交叉特征矩阵 \mathbf{C}_l :

$$\mathbf{C}_l = \mathbf{v}\mathbf{h}^T = \begin{bmatrix} \mathbf{v}^{(1)}\mathbf{h}^{(1)} & \mathbf{v}^{(1)}\mathbf{h}^{(2)} & \dots & \mathbf{v}^{(1)}\mathbf{h}^{d_2} \\ \mathbf{v}^{(2)}\mathbf{h}^{(1)} & \mathbf{v}^{(2)}\mathbf{h}^{(2)} & \dots & \mathbf{v}^{(2)}\mathbf{h}^{d_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}^{d_2}\mathbf{h}^{(1)} & \mathbf{v}^{d_2}\mathbf{h}^{(2)} & \dots & \mathbf{v}^{d_2}\mathbf{h}^{d_2} \end{bmatrix} \quad (10)$$

式中: $\mathbf{C}_l \in \mathbf{R}^{d_2 \times d_2}$ 为 l 层的交叉特征矩阵, d_2 为隐藏层维度。之后,将矩阵投影到各自的潜在表示空间来获得下一项项目与实体的特征向量:

$$\mathbf{v}_{l+1} = \mathbf{C}_l\mathbf{w}_l^{VV} + \mathbf{C}_l^T\mathbf{w}_l^{EV} + \mathbf{b}_l^V = \mathbf{v}_l\mathbf{e}_l^T\mathbf{w}_l^{VV} + \mathbf{e}_l\mathbf{v}_l^T\mathbf{w}_l^{EV} + \mathbf{b}_l^V \quad (11)$$

$$\mathbf{h}_{l+1} = \mathbf{C}_l\mathbf{w}_l^{VE} + \mathbf{C}_l^T\mathbf{w}_l^{EE} + \mathbf{b}_l^E = \mathbf{v}_l\mathbf{e}_l^T\mathbf{w}_l^{VE} + \mathbf{e}_l\mathbf{v}_l^T\mathbf{w}_l^{EE} + \mathbf{b}_l^E \quad (12)$$

式中: $\mathbf{w}_l^{\cdot\cdot} \in \mathbf{R}^d$ 与 $\mathbf{b}_l^{\cdot} \in \mathbf{R}^d$ 为可训练的权重与偏差向量,通过 L 层交叉压缩单元后得到项目与头实体的最终表示 $\mathbf{v}_L, \mathbf{h}_L$ 。

推荐模块中,在获得用户与物品向量的最终表示后,将其结合以预测概率:

$$\hat{\mathbf{y}}_{uv} = \sigma(f_{\text{RS}}(\mathbf{u}_L, \mathbf{v}_L)) \quad (13)$$

式中: f_{RS} 为多层 MLP 单元,再通过 sigmoid 函数得到最终结果。知识图谱嵌入模块中,通过 MLP 聚合头实体与关系的最终表示并得到预测分数:

$$\hat{\mathbf{t}} = \mathbf{M}^K \left(\begin{bmatrix} \mathbf{h}_L \\ \mathbf{r}_L \end{bmatrix} \right) \quad (14)$$

$$\text{score}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = f_{\text{KG}}(\mathbf{t}, \hat{\mathbf{t}}) \quad (15)$$

式中: $\hat{\mathbf{t}}$ 是尾实体 t 的预测向量,使用归一化内积 $f_{\text{KG}}(\mathbf{t}, \hat{\mathbf{t}}) = \sigma(\mathbf{t}^T\hat{\mathbf{t}})$ 作为得分函数。

2.5 模型优化学习

兴趣转移提取模块中 loss 函数表示如下:

$$L_1 = - \sum_{S^u \in S} \sum_{t \in [1, 2, \dots, n]} [\log(\sigma(r_{O_t, t})) + \sum_{j \in S^u} \log(1 - \sigma(r_{j, t}))] \quad (16)$$

式中: O_t 定义为时间点 t 的预期输出。

$$O_t = \begin{cases} \langle \text{pad} \rangle & \text{若 } s_t \text{ 为填充项} \\ s_{t+1} & 1 \leq t < n \\ S_{|S^u|}^u & t = n \end{cases} \quad (17)$$

推荐与知识图谱嵌入双任务的 loss 函数表示如下:

$$L_2 = L_{RS} + L_{KG} + L_{REG} = \sum_{u \in U, v \in V} J(\hat{y}_{uv}, y_{uv}) - \left(\sum_{(h, r, t) \in G} \text{score}(h, r, t) - \sum_{(h', r, t') \notin G} \text{score}(h', r, t') \right) + \lambda \|w\|_2^2 \quad (18)$$

式中:等式右边的三项分别为推荐模块损失、知识图谱嵌入模块损失与正则化项,正则化项的任务为防止过拟合。 λ 为平衡参数。 J 为交叉熵函数,计算式如下:

$$J(p, q) = - \sum_x p(x) \log q(x) \quad (19)$$

在每一个训练周期,对推荐任务重复 t 次,对知识图谱嵌入任务重复 1 次。同时, L_1 和 L_2 均采用 adam 优化器,兴趣提取模块捕获用户原始的兴趣转移后,馈入推荐与知识图谱双任务,进行用户特征向量的修正和再提取。

算法 1 KSMR

输入:交互矩阵 Y ,知识图谱 G 。

输出:预测函数 $F(u, v | \theta, Y, G)$ 。

1. 初始化所有参数
2. 通过式(2) - 式(6)和式(16),训练用户向量
3. for 训练迭代次数 do
4. 以下进行 t 次 //推荐任务
5. 通过交互矩阵采样正负样本
6. 将各 minibatch 传入模型
7. 通过式(7)、式(9) - 式(13)和式(18)更新 F 的参数
8. end for
9. 以下进行 1 次 //知识图谱嵌入任务
10. 通过知识图谱采样正负三元组
11. 将各 minibatch 传入模型
12. 通过式(8)、式(9) - 式(12)、式(14) - 式(15)和式(18)更新 F 的参数
13. end for

3 实验与结果分析

3.1 实验数据集

实验采用了 2 个公开的数据集,MovieLens-1M 数据集采集自现实世界中用户对电影的交互评分,是推荐领域被广泛使用的基准数据集;Last. FM 数据集包

含来自 Last. FM 在线音乐系统的近 2 000 名用户的音乐交互评分。对应电影和音乐数据集的知识图谱来源于 MKR^[5] 公开的数据信息,由 Microsoft Satori 构建而成。将有效的电影和音乐家姓名与数据库中三元组的尾实体相匹配以收集有效 id,同时过滤掉数据库中不存在实体的项目,再将所有头、尾实体中出现有效 id 的三元组提取,得到每个数据集对应的知识图谱。2 个数据集统计信息及对应知识图谱三元组数量如表 1 所示。

表 1 实验数据集

名称	用户数	项目数	交互数	三元组数
MovieLens-1M	6 040	3 882	1 000 209	20 195
Last. FM	1 872	3 846	42 346	15 518

同时,交互评分记录需要由显式反馈转换为隐式反馈,将电影集与音乐集的评分阈值分别设置为 4、0 (Last. FM 数据集较稀疏),评分大于等于此值的交互记录标记为 1,表示用户对项目具有正面的评价。此外,为每个用户随机抽样若干标记为 0 的项目作为未交互的集合。

3.2 对比算法

为验证 KSMR 的性能,本文采用推荐领域内的经典模型及与本文紧密相关的模型作为对比,针对评价指标进行论证。

(1) Wide&Deep^[15] 提出了一种广度和深度学习框架,使用宽线性模型有效地记忆稀疏矩阵特征交互作用,结合深度神经网络低维嵌入将稀疏矩阵特征交互作用进行推广。

(2) RippleNet 提出了联合知识图谱的端到端模型。KGCN^[16] 与 KGAT^[17] 同样基于此思想,但捕捉了更多的高阶信息。

(3) MKR 是一种利用知识图谱提供辅助信息的多任务推荐模型,与本文联系紧密。SI-MKR、KgeRec 和 TMR^[18] 为 MKR 的改进性工作,TMR 将物品输入结合了自身属性信息,同时使用 DeepFM 挖掘用户与物品的特征交互。

(4) PHGR^[19] 利用不同的图卷积分别挖掘用户图、项目图与交互图之间的异构信息,同时具有一定可解释性。

3.3 参数设置及实验条件

在兴趣转移提取单元中,将电影集、音乐集的最大序列长度分别设置为 500、300。使用两个自注意力模块堆叠提取特征,此部分的学习率 $l_r = 0.001$, $d_{\text{dropout}} =$

0.5, $b_{\text{atchsize}} = 128$ 。设置通过此单元后得到的用户向量维度 $d_1 = 64$ 。

在推荐单元及知识图谱嵌入单元,设置高层 MLP 层数 $K_h = 1$,卷积模块层数 $K_c = 1$, $b_{\text{atchsize}} = 4096$;平衡参数 $\lambda = 10^{-6}$;推荐单元学习率 $l_{r_s} = 0.02$,知识图谱嵌入单元学习率 $l_{r_{\text{kge}}} = 0.01$ 。

在文本卷积网络中,分别设置尺寸为 2×8 、 3×8 、 4×8 的单层卷积核对用户向量进行特征提取(实验验证卷积核尺寸的差异对结果影响较小),全连接层层数 $K_f = 4$ 。用户嵌入部分接收到来自兴趣提取单元的 d_1 后,通过卷积模块中的全连接层将维度转换为 $d_2 = 8$ 。

以上参数的设置由参考基准模型及优化验证集上的 AUC 值共同确定。对于两个数据集,设置训练、验证及测试的比例为 6:2:2。每个实验重复 3 次,统计平均性能。本实验实验平台采用 Ubuntu 16 系统、Python 3.6 及 PyTorch 1.6 开发环境;硬件环境为 NVIDIA Tesla V100 GPU,16 GB 显存,2.30 GHz Intel Xeon Gold 5118 CPU。

3.4 实验评价标准

本文采用点击率(CTR)预测中被广泛使用的 ROC 曲线下的面积(Area Under Curve, AUC)与准确率(Accuracy, ACC)来评估模型的性能。 A_{UC} 的计算式为:

$$A_{\text{UC}} = \frac{\sum_{i_{nS_i} \in P_{\text{positiveclass}}} r_{\text{ank}_{inS_i}} - \frac{M \times (M + 1)}{2}}{M \times N} \quad (20)$$

式中: $i_{nS_i} \in P_{\text{positiveclass}}$ 表示正样本的序号, M 、 N 分别为正样本与负样本的个数; $r_{\text{ank}_{inS_i}}$ 表示第 i 条样本的序号。依据预测概率; A_{UC} 表示正样本排在负样本之前的概率,其值越大越好。

A_{CC} 的计算式为:

$$A_{\text{CC}} = \frac{T_P + T_N}{P + N} \quad (21)$$

式中: T_P 为预测正确的正例数量, T_N 为预测正确的负例数量, P 和 N 分别为正例数和负例数。 A_{CC} 表示预测正确的样本数与总样本数之比,其值越大越好。

3.5 实验结果分析

本节对 KSMR 模型分别进行 CTR 预测、参数性能影响和稀疏数据集的对比实验,最后对其变体进行研究。

3.5.1 KSMR 的 CTR 预测对比

在 MovieLens-1M 及 Last. FM 数据集上均取 $d_1 = 64$ 、 $d_2 = 8$,对 KSMR 进行 AUC 与 Accuracy 性能测试,实

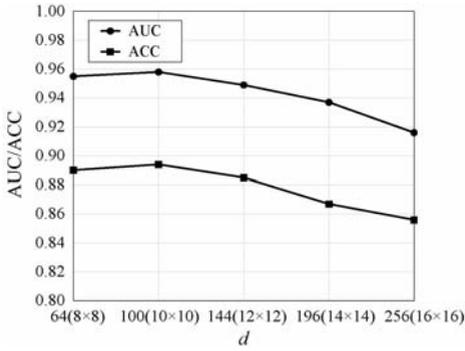
验结果如表 2 所示。观察可得 KSMR 在电影集上有较大提升,同时在音乐集上也超越了对比算法。KSMR 相对于对比算法中综合性能表现最优的 PHGR 在电影集上的 AUC 提升了 2.69%,在音乐集上提升了 1.10%。同时,相较于其紧密关联的 MKR 在电影集上的 AUC 和 ACC 分别提升了 4.14% 和 5.58%,在音乐集上分别提升了 0.88% 和 2.93%。音乐集相较电影集数据项目数相近,而用户数与交互数差距较大,这就直接反映了数据集的稀疏性,也是实验结果差异的直接来源。KSMR 在用户向量的表示中融入了兴趣转移特征,因此,在交互数据量更大的电影数据集上提升更为明显,直观地验证了模型算法的有效性。

表 2 各算法 CTR 预测的 AUC 与 ACC 结果

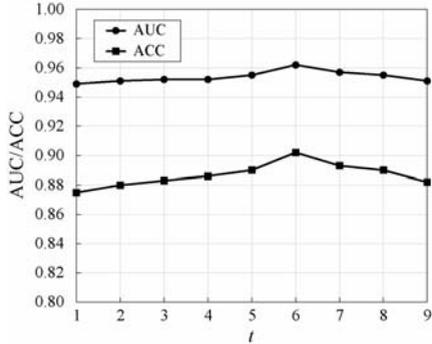
模型	MovieLens-1M		Last. FM	
	AUC	ACC	AUC	ACC
Wide&Deep	0.898 (-6.0%)	0.820 (-7.9%)	0.756 (-8.7%)	0.688 (-11.5%)
RippleNet	0.921 (-3.6%)	0.844 (-5.2%)	0.768 (-7.2%)	0.691 (-11.1%)
KGCN	0.927 (-3.0%)	0.853 (-4.3%)	0.804 (-3.0%)	0.732 (-6.1%)
KGAT	0.921 (-3.7%)	0.846 (-5.2%)	0.809 (-2.3%)	0.748 (-3.9%)
MKR	0.917 (-4.0%)	0.843 (-5.3%)	0.797 (-3.7%)	0.752 (-3.2%)
SI-MKR	0.921 (-3.7%)	0.845 (-5.1%)	—	—
TMR	0.926 (-3.0%)	0.852 (-4.3%)	—	—
KGeRec	0.925 (-3.1%)	0.850 (-4.5%)	0.807 (-2.5%)	0.754 (-3.0%)
PHGR	0.930 (-2.6%)	0.858 (-3.7%)	0.819 (-1.1%)	0.759 (-2.4%)
KSMR	0.955	0.890	0.828	0.777

3.5.2 参数对模型的影响

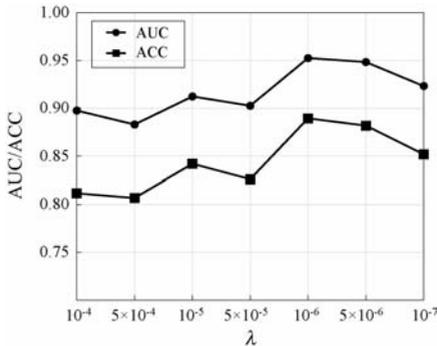
(1) 用户向量维度 d 的影响:模型顺利运行的关键是兴趣转移提取模块的用户向量输出维度需要与推荐模块用户向量输入的维度相匹配。此维度大小直接决定了用户向量表示的细粒化程度,如何取值也成为了影响模型性能的重要因素。为验证其影响,在其他实验参数不变的情况下,测试了 d 值分别在 64 至 400 之间 KSMR 在 MovieLens-1M 数据集上的性能表现,结果如图 2(a) 所示,观察到在 $d = 100$ 左右时性能达到最佳,即当前场景下 100 维左右的向量能最优地表示用户特征。



(a) 物品嵌入维度的影响



(b) 训练频率的影响



(c) 正则化参数的影响

图 2 参数对 KSMR 的影响

(2) 训练频率 t 的影响:参数 t 控制了推荐模块与知识图谱嵌入模块的训练比例,为验证其影响,在其他实验参数不变的情况下,测试了 t 值分别在 1 到 9 之间时 KSMR 在 MovieLens-1M 数据集上的性能表现,如图 2(b) 所示。观察到 AUC 和 ACC 呈现先上升后下降的趋势,在 $t=6$ 时性能达到最佳,分析其原因大致为在 $t=6$ 时,推荐模块损失函数的更新与 KG 传递的辅助信息可以达到相对均衡的状态。

(3) 正则化参数 λ 的影响:参数 λ 控制了函数更新的速率,在优化时尽量让其较小以避免过拟合现象。为找到最合适的 λ 值,在其他实验参数不变的情况下,测试了 λ 值在 10^{-4} 至 10^{-7} 之间时 KSMR 在 MovieLens-1M 数据集上的性能表现,如图 2(c) 所示。观察到在 $\lambda = 10^{-6}$ 左右时性能达到最佳。

3.5.3 稀疏数据集下的性能对比

在 MovieLens-1M 数据集下,将训练集的比例由 100% 逐步减小至 10%,测试对比算法的 AUC 数据,结果如表 3 所示。随着训练集比例的降低,所有算法的性能都有所下降,实验结果可以表征模型在面对冷启动及数据稀疏性问题时的表现。观察可知,KSMR 在 $r=10\%$ 时 AUC 值下降了 3.14%,其值小于所有对比算法,验证了模型的稳定性能,也说明采用自注意力的兴趣转移提取模块能够更加准确地建模用户表示。同时,SI-MKR、TMR 这些以 MKR 为核心的交替学习算法,明显优于 RippleNet 等其他算法。该现象说明通过知识图谱中的辅助信息为项目补充特征的方法,在面对较少交互数据集时更有优势。

表 3 不同训练集比例 r 下 MovieLens-1M 数据集的 AUC 结果

模型	r									
	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
Wide&Deep	0.898	0.884	0.876	0.858	0.840	0.821	0.815	0.809	0.802	0.788(-12.2%)
RippleNet	0.920	0.912	0.901	0.890	0.878	0.870	0.862	0.859	0.851	0.843(-8.4%)
MKR	0.917	0.913	0.908	0.903	0.897	0.889	0.882	0.881	0.874	0.868(-5.34%)
SI-MKR	0.921	—	0.900	—	0.898	—	0.884	—	0.876	—(<-4.89%)
TMR	0.926	0.916	0.913	0.910	0.908	0.905	0.897	0.884	0.881	0.879(-5.08%)
KSMR	0.955	0.954	0.952	0.950	0.947	0.941	0.939	0.937	0.931	0.925(-3.14%)

3.5.4 消融实验

为验证单一因素对模型性能的影响,从两个角度考虑 KSMR 的变体。

(1) KSMR-Conv. 取 $d_1 = 64, d_2 = 8$,将通过兴趣转移提取模块得到的用户向量馈入原模型中的多层感知机进行特征提取。

(2) KSMR-It. 参数同上,此时 d_1 为初始嵌入的用户向量,直接馈入文本卷积网络进行特征提取。

在 MovieLens-1M 和 Last.FM 数据集上验证两个变体的 AUC 指标,结果如图 3 所示,从左至右分别为 MKR、KSMR-Conv、KSMR-It 和 KSMR。可以观察到,在原模型上增加兴趣转移提取模块后,融合偏好的用户

向量会使推荐性能得到小幅提升;只修改特征提取单元,用文本卷积网络代替多层感知机后,在电影数据集上推荐性能得到提升,而在音乐数据集上推荐性能反而下降;综合了两点改进的模型,推荐性能得到了明显提升。

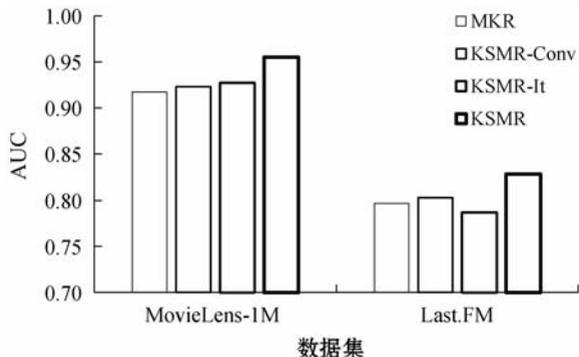


图3 消融实验的 AUC 结果

对实验结果进行总结:(1) 为用户表示提供先验信息对推荐性能的提升具有积极作用;(2) 只用文本卷积网络替换 MLP 层的模型对于密集数据集具有一定优越性,但在稀疏数据集上的表现较平庸;(3) 推荐性能的提升是凭借单一因素的各自优势,结合共同作用后的相互促进达到的。

4 结 语

针对大多数基于知识图谱的推荐算法中用户表示单一、缺乏偏好信息的问题,本文提出一种融合自注意力机制与知识图谱的多任务推荐模型。使用自注意力机制挖掘带有偏好转移的用户向量,再输入经过优化的特征提取环节中进行深层次特征修正与再提取,模型因此能够受益于表示丰富的用户画像,从而改善推荐效果。实验结果表明,在 MovieLens-1M 和 Last.FM 两个常用推荐数据集上,本文方法较其他主流方法具有明显优势。

在未来的工作中,将考虑对项目信息的输入添加更多元化的信息,例如项目的作者、类型等;或尝试寻找更加高效的特征提取单元进行用户、项目特征的提取。

参 考 文 献

[1] 李孟浩,赵学健,余云峰,等. 推荐算法研究进展[J]. 小型微型计算机系统,2022,43(3):544-554.

[2] Cui Z H, Xu X H, Fei X, et al. Personalized recommendation system based on collaborative filtering for IoT scenarios [J]. IEEE Transactions on Services Computing, 2020, 13 (4):685-695.

[3] Guo Q Y, Zhuang F Z, Qin C, et al. A survey on knowl-

edge graph-based recommender systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34 (8): 3549-3568.

- [4] Wang H W, Zhang F Z, Xie X, et al. DKN: Deep knowledge-aware network for news recommendation [C]//The World Wide Web Conference,2018:1835-1844.
- [5] Wang H W, Zhang F Z, Zhao M, et al. Multi-task feature learning for knowledge graph enhanced recommendation [C]//The World Wide Web Conference,2019:2000-2010.
- [6] Wang S J, Hu L, Wang Y, et al. Sequential recommender systems: Challenges, progress and prospects[EB]. arXiv: 2001.04830,2019.
- [7] Kang W C, McAuley J. Self-attentive sequential recommendation[C]//IEEE International Conference on Data Mining, 2018:197-206.
- [8] Zhang Y, Wallace B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification[EB]. arXiv:1510.03820,2015.
- [9] Zhang F Z, Yuan N J, Lian D, et al. Collaborative knowledge base embedding for recommender systems [C]//22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,2016:353-362.
- [10] Wang H W, Zhang F Z, Wang J L, et al. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems[C]//27th ACM International Conference on Information and Knowledge Management,2018:417-426.
- [11] Cao Y X, Wang X, He X N, et al. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences [C]//The World Wide Web Conference,2019:151-161.
- [12] Wang Y Q, Dong L Y, Zhang H, et al. An enhanced multi-modal recommendation based on alternate training with knowledge graph representation[J]. IEEE Access,2020,8: 213012-213026.
- [13] Yan C R, Liu S, Zhang Y T, et al. A multi-task learning approach for recommendation based on knowledge graph[C]//International Joint Conference on Neural Networks,2021.
- [14] 高仰,刘渊. 融合知识图谱和短期偏好的推荐算法[J]. 计算机科学与探索,2021,15(6):1133-1144.
- [15] Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems [C]//1st Workshop on Deep Learning for Recommender Systems,2016.
- [16] Wang H W, Zhao M, Xie X, et al. Knowledge graph convolutional networks for recommender systems[C]//The World Wide Web Conference,2019:3307-3313.
- [17] Wang X, He X N, Cao Y X, et al. KGAT: Knowledge graph attention network for recommendation[C]//25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,2019:950-958.

- 2017:5038 – 5047.
- [2] Fan D P, Wang W G, Cheng M, et al. Shifting more attention to video salient object detection[C]//IEEE Conference on Computer Vision & Pattern Recognition, 2019: 8546 – 8556.
- [3] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770 – 778.
- [4] Woo S, Park J, Lee J Y, et al. CBAM: Convolutional block attention module [C]//European Conference on Computer Vision, 2018: 3 – 19.
- [5] Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 834 – 848.
- [6] Lang C, Nguyen T V, Katti H, et al. Depth matters: Influence of depth cues on visual saliency[C]//European Conference on Computer Vision, 2012: 101 – 115.
- [7] Niu Y Z, Geng Y J, Li X Q, et al. Leveraging stereopsis for saliency analysis[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2012: 454 – 461.
- [8] Peng H W, Li B, Xiong W H, et al. RGB-D salient object detection: A benchmark and algorithms[C]//European Conference on Computer Vision, 2014: 92 – 109.
- [9] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248 – 255.
- [10] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[EB]. arXiv:2010.11929, 2020.
- [11] Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention [EB]. arXiv:2012.12877, 2012.
- [12] Piao Y R, Ji W, Li J, et al. Depth-induced multi-scale recurrent attention network for saliency detection[C]//European Conference on Computer Vision, 2019: 7253 – 7262.
- [13] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//31st International Conference on Neural Information Processing Systems, 2017: 5998 – 6008.
- [14] Lecun Y, Bottou L. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278 – 2324.
- [15] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3431 – 3440.
- [16] Peng H W, Li B, Xiong W H, et al. RGB-D salient object detection: A benchmark and algorithms[C]//European Conference on Computer Vision, 2014: 92 – 109.
- [17] Chen H, Li Y F. Progressively complementarity-aware fusion network for RGB-D salient object detection[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2018: 3051 – 3060.
- [18] Wang X H, Li S, Chen C, et al. Depth quality-aware selective saliency fusion for RGB-D image salient object detection [J]. Neurocomputing, 2020, 432: 44 – 56.
- [19] Piao Y R, Rong Z K, Zhang M, et al. A2dele: Adaptive and attentive depth distiller for efficient RGB-D salient object detection[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2020: 9057 – 9066.
- [20] Fan D P, Lin Z, Zhang Z, et al. Rethinking RGB-D salient object detection: Models, datasets, and large scale benchmarks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021: 2075 – 2089.
- [21] Chen H, Li Y H. Three-stream attention-aware network for RGB-D salient object detection[J]. IEEE Transactions on Image Processing, 2019, 28(6): 2825 – 2835.
- [22] Zhao J X, Cao Y, Fan D P, et al. Contrast prior and fluid pyramid integration for RGB-D salient object detection[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2019: 3922 – 3931.
- [23] Zhang M, Ren W S, Piao Y R, et al. Select, supplement and focus for RGB-D saliency detection[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2020: 3469 – 3478.
- [24] Liu N, Zhang N, Wan K Y, et al. Visual saliency transformer[C]//IEEE International Conference on Computer Vision, 2021: 4722 – 4732.
- [25] Zhou T, Fu H Z, Chen G, et al. Specificity-preserving RGB-D saliency detection[C]//IEEE/CVF International Conference on Computer Vision, 2021: 4681 – 4691.
- [26] Zhou W J, Lv Y, Lei J S, et al. Global and local-contrast guides content-aware fusion for RGB-D saliency prediction [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 51(6): 3641 – 3649.
- ~~~~~
- (上接第 148 页)
- [18] 朱瑞, 张俊三, 朱杰, 等. 一种基于 KCNN 和 MKR 的两阶段深度学习多任务推荐模型[J]. 陕西师范大学学报(自然科学版), 2020, 48(6): 82 – 89.
- [19] Fu Y, Wan J H, Zhao H, et al. Preference-aware heterogeneous graph neural networks for recommendation[C]//32nd International Conference on Tools with Artificial Intelligence, 2020: 41 – 46.