

融合知识图谱和用户行为信息的个性化推荐算法研究

程静文, 王全民

北京工业大学信息学部, 北京
Email: 604662371@qq.com

收稿日期: 2021年3月21日; 录用日期: 2021年4月15日; 发布日期: 2021年4月22日

摘要

针对传统协同过滤存在的稀疏性和冷启动问题, 通常使用深度神经网络(DNN)构建融合知识图谱和推荐系统的推荐任务。但目前的方法未曾考虑特征间的低阶线性关系, 虽可加入因子分解机(FM), 但不同的特征对模型的贡献不同, FM可能会因所有特征交互设置相同的权重而受到阻碍; DNN解决知识图谱这种具有不规则、可扩展、多重结构特性的数据结构不具普适性。针对以上问题, 提出MKAFG模型, 推荐部分加入具有注意力机制的FM, 通过注意力网络区分不同特征交互的重要性, 使FM提取到对目标预测起到重要作用的一阶、二阶线性交互特征。知识嵌入部分使用图卷积神经网络(GCN), 提高推荐系统推荐效果。实验结果表明, MKAFG在MovieLens-1M数据集上的推荐效果优于主流推荐模型。

关键词

推荐系统, 知识图谱, 深度神经网络, 注意力因子分解机, 图卷积神经网络

Research on Personalized Recommendation Algorithm Based on Knowledge Graph and User Behavior Information

Jingwen Cheng, Quanmin Wang

Information Department, Beijing University of Technology, Beijing
Email: 604662371@qq.com

Received: Mar. 21st, 2021; accepted: Apr. 15th, 2021; published: Apr. 22nd, 2021

Abstract

To solve the sparsity and cold start problems of traditional collaborative filtering, deep neural network (DNN) is usually used to construct the recommendation task of fusion knowledge map and recommendation system. However, the current method does not consider the low-order linear relationship between features. Although factor decomposition machine (FM) can be added, different features have different contributions to the model, and FM may be hindered because all features set the same weight interactively. DNN solves the problem that knowledge map, a data structure with irregular, extensible and multiple structural characteristics, is not universal. To solve the above problems, the MKAFG model is proposed, and FM with attention mechanism is recommended. The importance of interaction between different features is distinguished by attention network, so that FM can extract first-order and second-order linear interactive features which play an important role in target prediction. In the part of knowledge embedding, graph convolution neural network (GCN) is used to improve the recommendation effect of recommendation system. Experimental results show that MKAFG's recommendation effect on MovieLens-1M dataset is better than that of mainstream recommendation models.

Keywords

Recommendation System, Knowledge Graph, Deep Neural Network, Attentional Factorization Machines, Graph Convolutional Networks

Copyright © 2021 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

推荐系统旨在解决信息爆炸问题并解决用户的个性化需求。目前使用的协同过滤推荐算法使用用户-物品交互矩阵存在稀疏性和冷启动等问题。因此,通常在推荐系统中加入辅助信息来提高推荐性能。本文将知识图谱作为辅助信息,因其包含丰富的实体和关系。目前经常使用的知识图谱包括:NELL1、DBpedia2、Google Knowledge Graph3 和 Microsoft Satori4。由于知识图谱高维性和异构性,使用知识图谱嵌入(Knowledge Graph Embedding, KGE)将知识图谱中的实体和关系映射到低维向量中,并保持其原有的语义和结构[1]。

文献[2]提出了 CKE 模型,将结构化、文本、视觉等知识输入到贝叶斯框架中,但推荐部分和知识嵌入部分在贝叶斯框架下是松散耦合的[3],因此知识图谱对提高推荐系统的推荐效果起到的辅助性不太明显。文献[4]设计了基于内容的深度知识感知网络 DKN,其结合了实体嵌入和文字嵌入进行新闻推荐。但其只针对文本数据,使用场景受限,也无法做到端到端的训练。文献[5]设计了 RippleNet 记忆网络的类似模型,在知识图谱中传播用户的潜在偏好,探索分级兴趣,但关系的嵌入矩阵很难捕捉关系的重要性。文献[6]设计了一个通用的端到端的深度推荐框架 MKR,通过交叉压缩感知单元让推荐模块和知识嵌入模块进行交替学习,通过知识图谱提高推荐系统的性能。但在推荐过程中只考虑到了特征间的高阶非线性关系;在知识嵌入过程中,深度学习虽能够学习到更高效的特征与模式,但处理图数据结构不具普适性,使用深度神经网络存在一定的限制。

基于上述背景, 本文的主要贡献如下:

a) 基于知识图谱的思想提出了基于注意力分解机和图卷积神经网络的知识图谱的多任务推荐模型(Multi-Task Recommendation of Knowledge Graph Based on Attention Factorization Machine and Graph Convolution Neural Network, MKAFG);

b) 本文将加入 Attention [7]思想的因子分解机(Factorization Machines, FM)引入到深度推荐模型中, Attention 机制对目标有用的特征赋予更高的权重, 提取对目标预测起到重要作用的低阶线性特征交互。

c) 使用图卷积神经网络(Graph Convolutional Networks, GCN) [8]对知识图谱进行的知识嵌入, 解决深度学习模型处理图数据结构不具有普适性的问题。

2. 相关理论

2.1. 基于因子分解机的深度神经网络推荐模型

在推荐领域中, 对辅助信息进行 One-hot 编码时, 会带来数据稀疏问题, 而因子分解机(FM) [9]是一种较好的特征组合的方法来构建新特征[10]。FM 只能构建特征之间的低阶线性关系。深度神经网络具有学习复杂特征的交互潜力, 使用多层感知机(MLP) [11]实现高阶非线性的特征组合, 但无法实现低阶线性特征组合。针对以上 Cheng 等人提出了 Wide & Deep [12], 在 FM 的基础上引入了深度神经网络(Deep Neural Network, DNN), 加强模型的非线性能力, 但 Wide & Deep 没有将二阶交叉特征的信息完全表征出来, 造成 DNN 部分学习更高阶交叉信息效率低下。因此, Xiang 等人提出 NFM [13], 使用双向交互池结构对二阶交叉信息进行处理, 使高阶交叉特征的信息能更好的被 DNN 学习。Guo 等人提出了 DeepFM [14], FM 和 Deep 分别进行特征间低阶线性和高阶非线性组合, 提高训练效果。为了提高预测性, 阿里提出了融合 Attention 机制的 Det Interest Network, 在模型嵌入层和拼接层加入 Attention 单元, 根据特征的贡献程度调整不同的特征权重, 提高预测性[15]。

本文提出将 Attention 机制的 FM 融入到基于知识图谱的推荐算法中, 将具有 Attention 机制的 FM 与 DNN 相结合运用到推荐模块进行推荐。

2.2. 基于知识图谱的特征表示学习

知识图谱表示学习[16], 是知识图谱研究的重点。Mikolov 等人提出了 word2vec 词表示学习模型, 并发现词向量空间存在平移不变现象[17]。针对上述的启发, Bordes 等人提出了 TransE [18]模型, 其便于计算、参数少, 但 TransE 无法处理知识图谱复杂的实体和关系的建模。后人依次提出 TransH [19]、TransR [20]。但这些模型虽然在低维空间中可以重构出原有的网络结构, 但学习到的表征无法进一步学习。网络表示学习, 代表有深度神经网络、卷积神经网络等。深度学习虽能够学习到更高效特征与模式, 但图数据结构, 具有不规则、多重结构、可扩展性等特点, 因此知识表示过程中深度神经网络存在一定的限制。

本文使用图卷积神经网络(GCN)对知识图谱进行嵌入。通过给边和节点赋予特征, 不仅可以学习到知识图谱自身的关系和特征, 在卷积层传播过程中, 节点会接受邻居节点的信息不断更新自身的节点。

3. 方法的提出

本文提出了基于注意力分解机和图卷积神经网络的知识图谱的多任务推荐模型(Multi-Task Recommendation of Knowledge Graph Based on Attention Factorization Machine and Graph Convolution Neural Network, MKAFG)。由推荐部分、知识嵌入部分、交叉压缩感知部分组成。其中推荐部分由具有注意力机制的因子分解机(AFMM)和深度神经网络(DNN)两块组成。

3.1. 问题的定义

在推荐场景中, 假设有 m 位用户 $U = \{u_1, u_2, \dots, u_m\}$, n 个物品 $V = \{v_1, v_2, \dots, v_n\}$ 。组成用户 - 物品交互矩阵 $Y_{m \times n}$, 并用隐式反馈。 $y_{uv} = 1$ 表示用户 u 对物品 v 有过历史行为, 否则 $y_{uv} = 0$ 。同时加入知识图谱 G 来提高推荐效果。 G 由头实体 - 关系 - 尾实体 (h, r, t) 三元组组成。在很多推荐场景中, $v \in V$ 与知识图谱中一个或者多个实体有关联。

本文中, 给定用户-物品交互矩阵 $Y_{m \times n}$ 和知识图谱 G , 去预测用户 u 是否会对之前没有过交互的物品 v 产生兴趣。定义了预测函数 $\hat{y}_{uv} = F(u, v | \Theta, Y, G)$, \hat{y}_{uv} 表示用户 u 对物品 v 点击的概率, Θ 为函数 F 的参数。

3.2. MKAFG 算法框架

MKAFG 算法框架如图 1 所示。由推荐部分、知识嵌入部分、交叉压缩感知部分三个模块组成。左边的推荐模块以用户和物品作为输入, 右边的知识嵌入模块以实体和关系作为输入。左右两部分使用 One-hot 编码, 对输入的数据采用稀疏表示, 通过嵌入层将每个非零特征嵌入到一个稠密向量中。

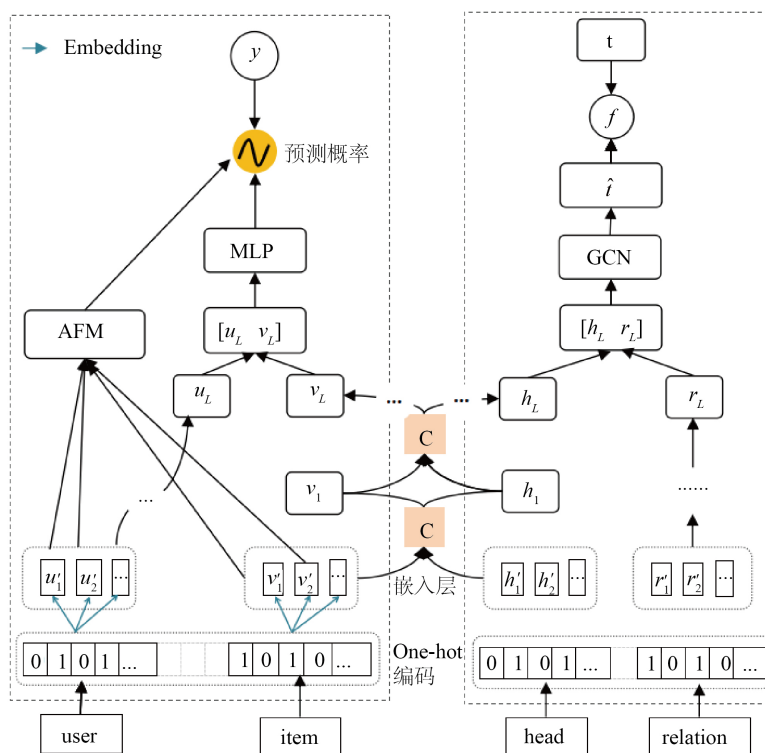


Figure 1. MKAFG algorithm recommendation framework

图 1. MKAFG 算法推荐框架

左边的推荐部分由具有注意力机制的因子分解机(Attentional Factorization Machines, AFM)与深度神经网络(DNN)组成。AFM 部分如图 2, 通过 Attention 机制对目标有用的特征赋予更高的权重, 从而使 FM 提取对目标预测起到重要作用的一阶和二阶线性交互特征; DNN 部分, 融合用户和物品特征向量之间高阶非线性关系。两部分学习到的特征输入到预测函数中, 得到预测概率; 右边的知识嵌入模块通过图卷积神经网络(GCN)从三元组的头实体和关系实体中提取特征, 并在函数 f 和真实的尾实体 t 的监督下输出

预测的尾部实体 \hat{t} ；推荐模块和知识嵌入模块通过交叉压缩单元连接，可以自动学习推荐系统中物品和知识图谱中实体的高阶特征交互。

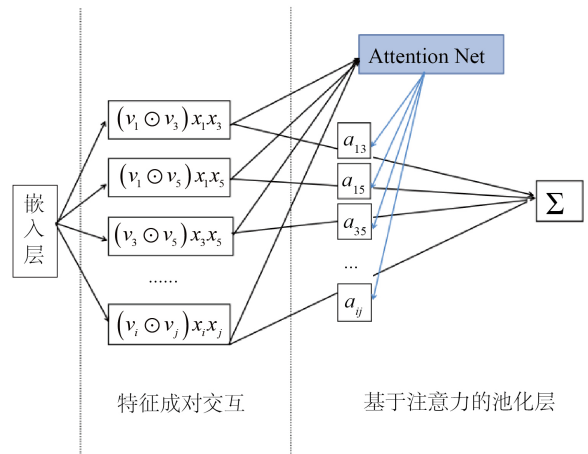


Figure 2. FM based on attention mechanism
图 2. 基于注意力机制的 FM

3.3. 推荐部分

本文将注意力机制(Attention)与 FM 加入到文献[6]提出的 MKR 左边的推荐部分中，由具有注意力机制的因子分解机(AFM)与深度神经网络(DNN)两部分共享相同的输入和嵌入层的输出，模型输出见公式(1)，用来预测用户 u 对物品 v 的点击概率。其中 y_{AFM} 是 AFM 部分的输出， y_{DNN} 为 DNN 部分的输出。

$$\hat{y}_{uv} = \text{sigmoid}(y_{AFM} + y_{DNN}) \quad (1)$$

3.3.1. AFM 模块

AFM 是用来学习加权组合特征的，是具有注意力机制的 FM [7]。AFM 输入层将输入的物品和用户数据进行 One-hot 编码后，通过嵌入层，将每个向量投入到一个密集向量中，得到如下公式：

$$y_{FM}(x) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n w_{ij} x_i x_j \quad (2)$$

其中 w_0 代表全局偏差， n 是经过 One-hot 编码后特征空间的维度， x_i 代表第 i 个特征。我们将上述结果通过文献提出的成对交互层[7]，如图 2，其是将 m 个向量扩展为 $m(m-1)/2$ 个相互作用的向量，其中每个相互作用的向量是两个不同向量元素的点积，来编码它们的相互作用。模型表达式为：

$$y_{FM}(x) = w_0 + \sum_{i=1}^n w_i x_i + p^T \sum_{(i,j) \in R_x} (v_i \odot v_j) x_i x_j \quad (3)$$

$p \in \mathbf{R}^k$ 表示预测层的偏差， \odot 为两个元素的点积， v_i 用于表达特征之间的相互作用。经过成对交互层，AFM 学到了特征向量之间低阶线性组合特征。

并非所有的组合特征会对目标预测产生相同的作用，因此本文采用文献[7]提出的 Attention 机制，对越重要的特征赋予越高的权重。使用多层感知机参数化注意力得分。如图 2，其由两个特征的交互向量 $(v_i \odot v_j) x_i x_j$ 作为输入，输入向量的加权算数平均值为输出。得到的权重代表特征各个组合特征的重要程度，权重函数为：

$$a'_{ij} = W' \sigma \left(W (v_i \odot v_j) x_i x_j + b \right) \quad (4)$$

$$a_{ij} = \frac{\exp(a'_{ij})}{\sum_{(i,j) \in R_x} \exp(a'_{ij})} \quad (5)$$

σ 是激活函数, W' , W , b 为模型学习的参数。对于输入的组合特征, 其返回的是非标准化的权重分值 a'_{ij} 。当得到所有组合特征的分值后, 带入到公式(5)就可以得到标准化的权重分值 a_{ij} 。将 a_{ij} 带入到公式(3)。AFM 提取到对目标预测起到重要作用的一阶和二阶线性交互特征。AFM 模型公式如下:

$$y_{\text{AFM}}(x) = w_0 + \sum_{i=1}^n w_i x_i + p^T \sum_{i=1}^n \sum_{j=i+1}^n a_{ij} (v_i \odot v_j) x_i x_j \quad (6)$$

3.3.2. 深度神经网络(DNN)模块

给定用户 u 的原始特征向量 u , 经过特征嵌入后, 使用一个 L 层 MLP 提取其隐性压缩特征, 见公式(7)。其中 $M(x) = \sigma(wx + b)$ 为全连接的神经网络, w 为权重(使用高斯分布初始化权重), b 为偏差, σ 为非线性激活函数。

$$u_L = M \left(M \left(\dots M(u_{em}) \right) \right) = M^L(u_{em}) \quad (7)$$

对于物品 v , 通过特征嵌入层, 使用交叉压缩单元共享知识图谱中对应实体的信息之后, 获取深层次的特征[6], 见公式(8)。其中 $S(v)$ 是物品 v 相关联的实体 h 集合, $[v]$ 是区分物品向量的标志[6]。

$$v_L = E_{h \sim S(v)} \left[A^L(v_{em}, h) [v] \right] \quad (8)$$

有了用户和物品的潜在特征 u_L 和 v_L 后, 投入到 DNN 中, 得到用户和物品的高阶组合特征。输出为:

$$y_{\text{DNN}} = \sigma \left(F(u_L, v_L) \right) \quad (9)$$

3.4. 知识嵌入部分

知识图谱的嵌入, 是将实体和关系嵌入到连续的向量空间中, 同时保持它原有的信息和结构。本文使用 GCN 提取知识图谱的空间特征, 利用边的信息对节点信息进行聚合从而生成新的节点表示, 信息可沿着边直接从一个节点传递到另一个节点[8]。

对于给定的知识三元组 (h, r, t) , 首先对 One-hot 编码的特征进行嵌入操作。让头部实体 h 经过多个交叉压缩单元来学习深层次的特征, 对于关系实体 r 使用 L 层的 MLP 提取隐性压缩特征。然后将 h_L 和 r_L 一起投入到图卷积神经网络中, 来预测尾实体 \hat{t} 。见如下公式:

$$h_L = E_{v \sim S(h)} \left[A^L(v_{em}, h) [h] \right] \quad (10)$$

$$r_L = M \left(M \left(\dots M(r_{em}) \right) \right) = M^L(r_{em}) \quad (11)$$

$$\hat{t} = \sigma \left(F_{\text{gcn}}^k(h_L, r_L) \right) = \sigma \left(C^* \sigma \left(C^* H^{(l)} W^{(l)} \right) W^{(l+1)} \right) \quad (12)$$

$S(h)$ 是与头实体 h 相关联的物品集合, $[h]$ 区分头实体向量的标志[6]。 \hat{t} 是尾实体的预测向量。 F_{gcn}^k 是 k 层的图卷积神经网络, 本文设置 $k = 2$, 叠加两层 GCN, 每个节点可以把 2-hops 邻居的特征加以聚合, 得到自身特征。 $\tilde{C} = C + I_N$, 表示通过添加自环, 把自身特征和邻居特征结合起来, 更新自身节点。 A 为描述节点之间关系的邻接矩阵。 $C^* = \tilde{D}^{-1/2} \tilde{C} \tilde{D}^{-1/2}$ 对自身节点和邻居节点所传播的信息进行归一化。 $W^{(l)}$ 为特定层的可训练权重矩阵。 σ 为非线性激活函数。使用分数函数 f_{KG} 来计算三元组分数:

$$\text{score}(h, r, t) = f_{\text{KG}}(t, \hat{t}) \quad (13)$$

3.5. 交叉压缩部分

本文使用交叉压缩单元与 MKR [6]一致。物品 v 向量和实体 h 向量通过交叉特征共享单元进行信息交互，弥补自身信息稀疏性的不足[6]，如图 3。

对于物品 v 与其相关联的实体 h ，其 L 层的隐特征分别为 $v_L \in \mathbf{R}^d$ 和 $h_L \in \mathbf{R}^d$ 。通过计算 v_L 与 h_L 的外积，构建了一个 $d \times d$ 维的隐特征成对相互作用的交叉特征矩阵。如下：

$$A_L = v_L h_L^T = \begin{bmatrix} v_L^{(1)} h_L^{(1)} & \cdots & v_L^{(1)} h_L^{(d)} \\ \vdots & \ddots & \vdots \\ v_L^{(d)} h_L^{(1)} & \cdots & v_L^{(d)} h_L^{(d)} \end{bmatrix} \quad (14)$$

其中 $A_L \in \mathbf{R}^{d \times d}$ 是 L 层的交叉特征矩阵， d 是嵌入向量的维度。由公式(14)可知，交叉特征矩阵 A_L 显示建模了物品 v 和其相关联的实体 h 的每个特征间的交互 $v_L^{(i)} h_L^{(j)}, \forall (i, j) \in \{1, \dots, d\}^2$ [6]。然后，将 A_L 投影到隐性表示空间，输出下一层的项目和实体的特征向量，此过程为压缩操作，见如下公式：

$$v_{L+1} = A_L w_L^{vv} + A_L^T w_L^{hv} + b_L^v = v_L h_L^T w_L^{vv} + h_L v_L^T w_L^{hv} + b \quad (15)$$

$$h_{L+1} = A_L w_L^{vh} + A_L^T w_L^{hh} + b_L^h = v_L h_L^T w_L^{vh} + h_L v_L^T w_L^{hh} + b_L^h \quad (16)$$

其 $w_L^v \in \mathbf{R}^d$ 和 $b_L^v \in \mathbf{R}^d$ 是可训练的权重和偏差向量。权重向量将 A_L 从 $\mathbf{R}^{d \times d}$ 维压缩到 \mathbf{R}^d 维。 A_L 沿着，水平和垂直两个方向进行对称压缩。交叉压缩单元可简单表示为：

$$[v_{L+1}, h_{L+1}] = A(v_L, h_L) \quad (17)$$

$[v]$ 和 $[h]$ 分别表示物品向量和实体向量的输出，通过交叉压缩感知单元，MKAFG 可以自适应地调整知识转移的权重，并学习两个任务之间的相关性[6]。

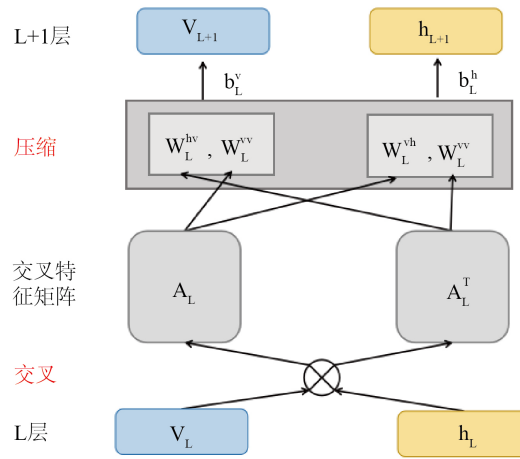


Figure 3. Cross compression module

图 3. 交叉压缩感知单元

3.6. 算法设计

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{RS} + \mathcal{L}_{KG} + \mathcal{L}_\theta \\ &= \sum_{u \in \mathbf{U}, v \in \mathbf{V}} \mathcal{H}(\hat{y}_{uv}, y_{uv}) - \lambda_1 \left(\sum_{(h,r,t) \in G} \text{score}(h, r, t) - \sum_{(h',r',t') \notin G} \text{score}(h', r', t') \right) + \lambda_2 \|W\|_2^2 \end{aligned} \quad (18)$$

MKAFG 的损失函数如公式(18)。 \mathcal{L}_{RS} 为推荐部分的损失函数， \mathcal{H} 为交叉熵函数； \mathcal{L}_{KG} 为知识嵌入部分的计算损失，旨在提高正确三元组得分，降低错误三元组得分； \mathcal{L}_θ 为正则化项，用来防止过拟合，其中 λ_1 和 λ_2 表示平衡系数， λ_1 可视为两项任务的两个学习率之比，本文设置 $\lambda_1 = 0.5$ ， $\lambda_2 = 10^{-6}$ 。上述公式遍历了所有可能的用户 - 交互矩阵和知识三元组。本文在训练过程中使用负采样策略[21]来提高计算的有效性。见“算法 1”(表 1)，由推荐部分与知识嵌入部分联合训练。每次迭代过程中，更加倾向于提高推荐性能，所以 RS 训练 t 次(t 是超参， $t > 1$)后，KGE 训练 1 次。

Table 1. Model Training Structure about MKAFG

表 1. MKAFG 模型训练结构

算法 1: MKAFG 算法流程

输入: 用户-物品交互矩阵 Y_{mn} ，知识图谱 G

输出: 预测函数: $\hat{y}_{uv} = F(u, v | \Theta, Y, G)$

1. 初始化参数
2. **for** k **do**
- //推荐部分(AFM 模块+DNN 模块)
3. **for** t steps **do**
4. 从 Y 中采集正负相互作用的小批量样本，得到物品 v 相关联的实体 $h \sim S(v)$;
5. 对输入的物品、用户数据进行 One-hot 编码;
6. One-hot 编码后的用户和物品特征向量进行嵌入操作;
- //AFM 模块
7. 使用神经注意力网络，公式(4)-(5)，以不同的权重 a_j 对所有的特征建模后，再通过公式(6)获取特征间的低阶线性交互关系。
- //DNN 模块
8. 获取用户和物品高阶组合特征，见公式(7)-(9);
9. 通过公式(1)预测用户 u 对物品 v 的点击概率;
10. 通过公式(18)更新推荐模块的所有参数;
11. **end for**
- //GCN 知识嵌入部分
12. 从知识图谱 G 中采集正负相互作用的小批量样本，得到头实体 h 相关联的物品 $v \sim S(h)$;
13. 对输入的头实体、关系实体数据进行 One-hot 编码;
14. 头实体、关系实体特征向量进行嵌入操作;
15. 头实体 h 和关系实体 r 构建邻接矩阵，预测尾实体 \hat{t} ;
16. 通过等式上的梯度下降更新 F 参数，公式(13)，(18);
17. **end for**

4. 实验

4.1. 实验环境

本实验的运行环境为: Python3.6、TensorFlow2.1、32G 内存、Intel Core i5CPU、GeForce GTX 1050 2G 显卡。

4.2. 数据集及预处理

4.2.1. 数据集

推荐模块数据集: 选用公开的 MovieLens-1M 数据集。其中评分数据集含有来自 6036 名用户对 3952 部电影约 100 万条显示评分数据(评分值从 1~5，值越高表示用户对该电影越喜欢); 用户数据集包含了用户属性信息(年龄、性别等); 电影属性数据集包含了电影属性(类别、年代等)。

知识图谱数据集: 使用 Microsoft Satori 来构建电影知识图谱。Microsoft Satori 存储实体(真实世界中的人物、地点、事件等)的信息和关系，索引量并在不断增长。

4.2.2. 数据预处理

推荐模块数据预处理: 在 MovieLens-1M 中需要将显示反馈数据转换为隐式反馈数据。用户对电影的评分为最高为 5, 最低为 1。本文根据电影评分的经验, 设阈值为 4, 当用户对某一部电影的评分 ≥ 4 时, 表示用户对该部电影很喜欢, 是正反馈, 用 1 表示; 反之是负反馈, 用 0 表示。0 不只意味着用户不喜欢这个电影, 它是用户不喜欢此电影和暂无交互(用户还未注意到此电影)的混合。

知识图谱数据预处理: 电影知识图谱的构建与 MKR [6]中保持一致。先从整个 Microsoft Satori 中选出置信度大于 0.9 的三元组子集, 进一步选择出关系名称为“电影”的三元组集合, 来减少知识图谱的大小。在得到的子集中, 通过将所有有效电影与三元组的尾部(head, film.film.name, tail)进行匹配来收集所有有效电影的 ID 标识, 将没有匹配或多个匹配的实体 ID 删除, 然后将剩下的 ID 集合与 KG 子集中的三元组头尾部进行匹配, 从 KG 中选择能够匹配上的三元组[6]。但用户、项目及交互的数量比原始数据集少, 因为过滤掉了 KG 中没有对应到的实体的项目。

4.3. 衡量指标及实验结果

本文将 MovieLens-1M 数据集, 按照 6:2:2 的比例划分为训练集、测试集、验证集。实验重复 3 次, 计算 3 次平均的结果。

4.3.1. 衡量指标

使用 ACC 和 AUC 来评估点击率预估(CTR)预测 MKAFG 推荐框架的性能。其中 ACC 可以从混淆矩阵中导出, 混淆矩阵见表 2。

Table 2. Confusion Matrix

表 2. 混淆矩阵

推荐框架	用户喜欢	用户不喜欢
推荐	TP	FP
未推荐	FN	TN

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (19)$$

AUC 被定义为 ROC 曲线下的面积。ROC 曲线的横坐标是伪阳性率(FPR), 纵坐标是真阳性率(TPR)。AUC 越大模型的推荐效果更好。

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (20)$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (21)$$

4.3.2. 实验结果

本实验使用 Movielens-1M 电影数据集进行了点击率预估任务。第一部分将 MKAFG 与主流推荐模型进行比较; 第二部分是自身实验的比较, 见表 2。结果最好已经加粗表示。

PER 在电影数据集上表现不佳, 因为 PER 需要手动进行元路径的设计, 手动设计很难使结果达到最佳。DKN 在所有推荐模型中表现最差, 可能是 DKN 主要应用在新闻领域的推荐, 新闻文本的特点是语言高度浓缩, 而电影名称较短无法提供有用的信息。RippleNet 在所有模型中表现较好, 说明通过兴趣点

传播可以更好的捕获用户的兴趣偏好。MKR 在所有模型中表现较佳, 说明知识图谱作为辅助信息可以辅助推荐任务进行学习, 其物品、实体通过交叉压缩单元共享信息表现出多任务学习的有效性。

自身实验的比较。MKFM 表示在推荐模块加入 FM, 学习用户、物品的低阶交互特征。由表 3 可知, MKFM 与基础模型相比要好, 说明在推荐模块加入 FM 的有效性, 但推荐效果不如 MKAFM, 其是加入了具有 Attention 的 FM, 能更好的能获取到, 起到重要作用的一阶二阶交互特征。推荐效果最好的是本文提出的 MKAFG 模型, AUC 分别比 MKR、MKFM、MKAFM 提高了 2.8%、1.1%、0.6%。说明加入的图卷积神经网络能更好的学习知识图谱的特征。因此, 在电影推荐领域, 本文提出的 MKAFM 效果最佳。

Table 3. Click rate estimation results of the model

表 3. 模型的点击率预估结果

Approachs	Movielens-1M	
	AUC	ACC
PER	0.710	0.664
CKE	0.801	0.742
DKN	0.655	0.589
RippleNet	0.920	0.842
MKR	0.917	0.843
MKFM	0.934	0.858
MKAFM	0.939	0.866
MKAFG	0.945	0.876

4.4. 推荐模块超参数分析

本部分实验研究在推荐模块引入 AFM 机制后, 推荐模块的 DNN 部分的激活函数、神经元数量、隐藏层数量使 MKAFG 达到推荐效果最佳。

4.4.1. 激活函数

此部分研究了 ReLU 和 Tanh 作为左侧推荐模块的激活函数时, 对总体框架结果的影响。由图 4 可得 ReLU 比 Tanh 性能更好。

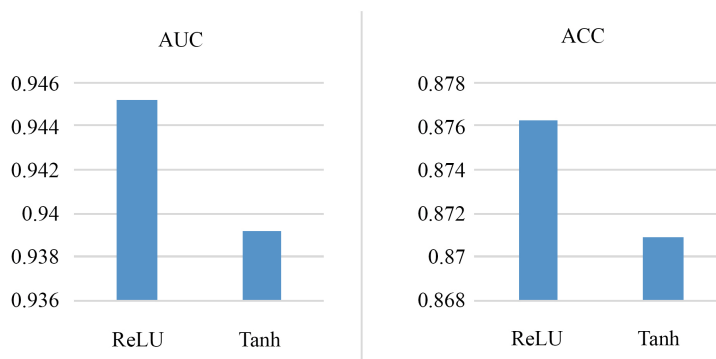


Figure 4. Influence of activation function on results

图 4. 激活函数对结果的影响

4.4.2. 隐藏层数量

一般认为, 隐藏层的个数越多, 模型的学习能力越强。但随着隐藏层数量的不断增加, 更容易产生过拟合, 同时训练难度也不断加大。如图 5, 当神经网络隐藏层数为 2 时, 模型的推荐效果最佳。

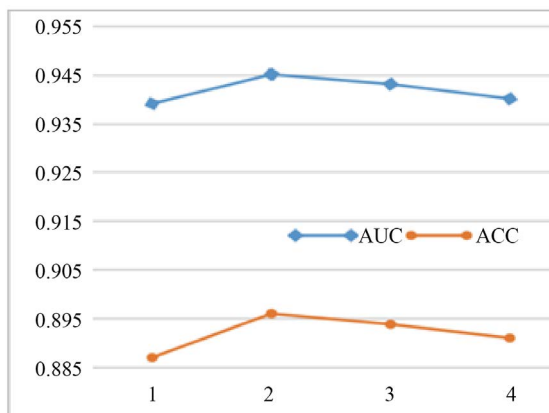


Figure 5. Influence of the number of hidden layers on the results

图 5. 隐藏层的数量对结果的影响

4.4.3. 隐因子维度分析

如图 6, 研究了隐因子的个数对 MKAFG 模型的影响。此处隐藏层数设置为 2。随着隐因子维度的增加, 并不能总给推荐效果带来提升。当隐因子维度为 128 时, 推荐效果最佳。

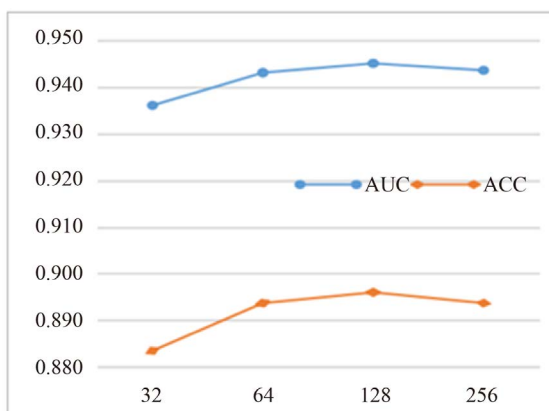


Figure 6. Influence of the number of hidden factors on experimental results

图 6. 隐因子个数对实验结果的影响

4.5. MKAFG 模型参数分析

4.5.1. 训练集稀疏程度分析

实验将 MovieLens-1M 的训练集大小设置为 10%到 100%。测试集和验证集不变。所有模型的 AUC 全部升高; 同时 $r = 100\%$ 与 $r = 10\%$ 时, 各个模型 AUC 差分别为: PER (11.2%)、CKE (12.7%)、DKN (7.6%)、RippleNet (7.7%)、MKR (5.5%)、MKAFG (5.1%)。如图 7, MKAFG 面对数据稀疏性问题时, 依然保持良好的性能。

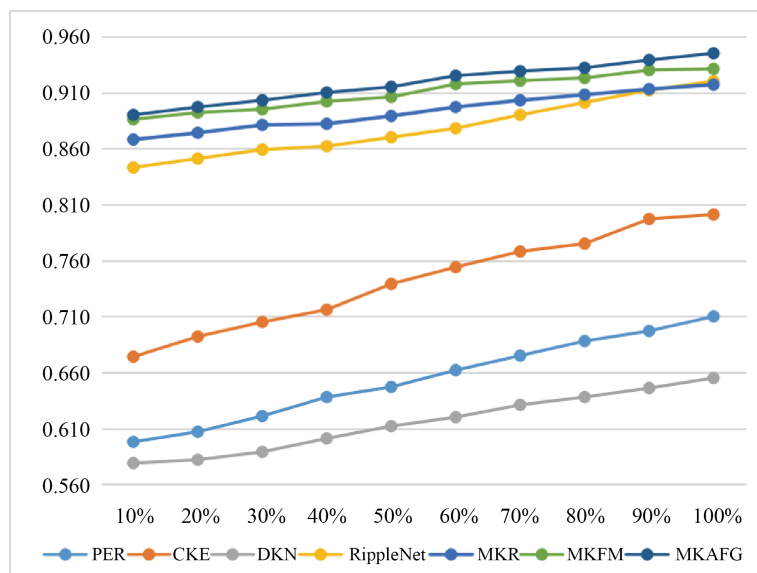


Figure 7. Analysis of sparsity of training set
图 7. 训练集稀疏程度分析

4.5.2. 知识图谱 KG 大小分析

本部分研究辅助信息数据集的大小对推荐效果的影响。将知识图谱的大小设置为原来的 0.1 倍到 1 倍。由图 8, 随着知识图谱数据集增大, 推荐效果不断上升。且 AUC 和 ACC 分别增加了 14.1%、12.8%。说明引入辅助信息可以提高推荐效果, 同时辅助信息数据集越大, 推荐效果会越好。

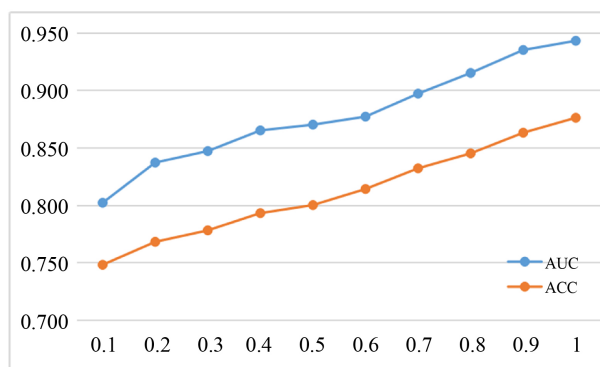


Figure 8. Analysis of KG size
图 8. 知识图谱大小分析

4.5.3. 推荐模块训练频次分析

每个 epoch, 推荐部分训练 t 次, 知识嵌入部分训练 1 次。本文设计 $t(1-10)$, 同时保持其他参数不变。如图 9, 当 $t=4$ 时, MKAFG 性能最佳, 当 $t>4$ 时, 随着 t 的增大时, MKAFG 性能降低。因为如果知识图谱嵌入模块训练频次太少, 不能充分利用知识图谱的知识; 训练频次过高, 则会影响 MKAFG 的目标函数。

4.5.4. 嵌入维度分析

在嵌入层, 用户、物品、实体、关系的嵌入维度会影响 MKAFG 的表现。当嵌入维度增加时, 会带来更多的编码信息。如图 10, 当 $\text{dim} = 8$ 时, MKAFG 的推荐结果达到最优。原因是, 随着嵌入维度的

增加, 会引入噪声, 这会误导预测函数的预测。

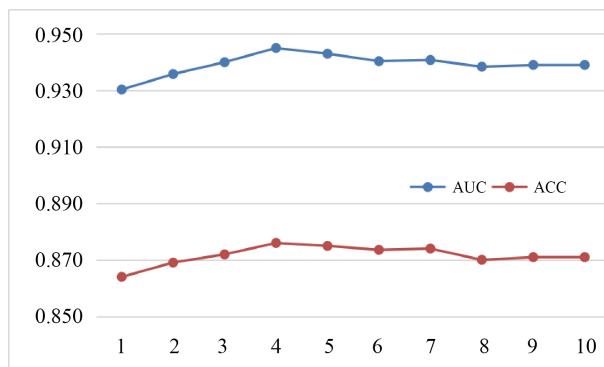


Figure 9. Analysis of training frequency of recommendation module

图 9. 推荐模块训练频次分析

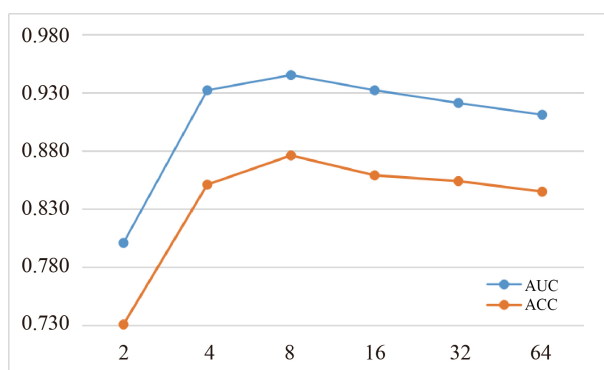


Figure 10. Embedded dimension analysis

图 10. 嵌入维度分析

5. 结束语

本文提出了融合知识图谱和用户行为信息的深度算法推荐模型。在推荐模块加入具有注意力机制的 FM, 通过注意力网络区分不同特征交互的重要性, 使 FM 提取到对目标预测起到重要作用的一阶、二阶线性交互特征。在知识嵌入模块通过 GCN 学习知识图谱自身的关系和特征, 还可以通过不断学习让每个节点接受邻居节点的信息从而更新自身的节点表示。通过实验表明, 该模型的性能相比目前主流推荐模型有所提升。由于本文数据只在 Movielens-1M 电影数据集上进行研究, 未来将应用到图书、新闻、音乐等领域进行推荐研究, 以能在所有数据集上获取较好的效果。

参考文献

- [1] Wang, H., Wang, J., Wang, J., et al. (2017) GraphGAN: Graph Representation Learning with Generative Adversarial Nets. *IEEE Transactions on Knowledge and Data Engineering*.
- [2] Zhang, F.Z., Yuan, N.J., Lian, D.F., et al. (2016) Collaborative Knowledge Base Embedding for Recommender Systems. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, 353-362. <https://doi.org/10.1145/2939672.2939673>
- [3] Bahdanau, D., Cho, K. and Bengio, Y. (2016) Neural Machine Translation by Jointly Learning to Align and Translate. *Proceedings of the 3rd International Conference on Learning Representations*, Banff, September 2014, 353-362.
- [4] Wang, H.W., Zhang, F.Z., Xie, X., et al. (2018) DKN: Deep Knowledge-Aware Network for News Recommendation.

Proceedings of the 2018 World Wide Web Conference on World Wide Web, Lyon, 23-27 April 2018, 1835-1844.

<https://doi.org/10.1145/3178876.3186175>

- [5] Wang, H.W., Zhang, F.Z., Wang, J.L., *et al.* (2018) RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ACM Press, Torino, 417-426. <https://doi.org/10.1145/3269206.3271739>
- [6] Wang, H.W., Zhang, F.Z., Zhao, M., *et al.* (2019) Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. *Proceedings of the 2019 World Wide Web Conference on World Wide Web (WWW 2019)*, New York, May 2019, 2000-2010. <https://doi.org/10.1145/3308558.3313411>
- [7] Xiao, J., Ye, H. and He, X.N. (2017) Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. <https://doi.org/10.24963/ijcai.2017/435>
- [8] Kipf, T.N. and Welling, M. (2017) Semi-Supervised Classification with Graph Convolutional Networks. *International Conference on Learning Representations*, Toulon, 24-26 April 2017, 14. <https://arxiv.org/abs/1609.02907>
- [9] Rendle, S. (2010) Factorization Machines. *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, Sydney, 13-17 December 2010, 995-1000. <https://doi.org/10.1109/ICDM.2010.127>
- [10] Rendle, S. (2012) Factorization Machines with LibFM. *ACM Transactions on Intelligent Systems and Technology*, **3**, 57:1-57:22. <https://doi.org/10.1145/2168752.2168771>
- [11] Pal, S.K. and Mitra, S. (1992) Multilayer Perceptron, Fuzzy Sets, and Classification. *IEEE Transactions on Neural Networks*, **3**, 683-697. <https://doi.org/10.1109/72.159058>
- [12] Cheng, H.T., Koc, L., Harmsen, J., *et al.* (2016) Wide & Deep Learning for Recommender Systems. *Workshop on Deep Learning for Recommender Systems*, Boston, 15 September 2016, 7-10. <https://doi.org/10.1145/2988450.2988454>
- [13] He, X.N. and Chua, T.S. (2017) Neural Factorization Machines for Sparse Predictive Analytics. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Tokyo, 7-11 August 2017, 355-364.
- [14] Guo, H.F., Tang, R.M., Ye, Y.M., *et al.* (2017) DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, Melbourne, 19-25 August 2017, 1725-1731. <https://doi.org/10.24963/ijcai.2017/239>
- [15] Zhou, G., Song, C., Zhu, X., *et al.* (2017) Deep Interest Network for Click-Through Rate Prediction. *Proceedings of the 40th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, June 2017, 1058-1068.
- [16] 刘知远, 孙茂松, 林衍凯, 等. 知识表示学习研究进展[J]. 计算机研究与发展, 2016, 53(2): 247-261.
- [17] Bordes, A., Weston, J., Collobert, R., *et al.* (2011) Learning Structured Embeddings of Knowledge Bases. *National Conference on Artificial Intelligence*, Barcelona, 16-22 July 2011, 301-306.
- [18] Bordes, A., Usunier, N., Garciaduran, A., *et al.* (2013) Translating Embeddings for Modeling Multi-Relational Data. *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Volume 2, 2787-2795.
- [19] Wang, Z., Zhang, J., Feng, J., *et al.* (2014) Knowledge Graph Embedding by Translating on Hyperplanes. *National Conference on Artificial Intelligence*, Québec, 27-31 July 2014, 1112-1119.
- [20] Lin, Y., Liu, Z., Sun, M., *et al.* (2015) Learning Entity and Relation Embedding for Knowledge Graph Completion. In: *Proceedings of AAAI, AAAI*, Menlo Park, 2181-2187.
- [21] Yang, S., Korayem, M., Aljadda, K., *et al.* (2017) Combining Content-Based and Collaborative Filtering for Job Recommendation System: A Cost-Sensitive Statistical Relational Learning Approach. *Knowledge-Based Systems*, **136**, 37-45. <https://doi.org/10.1016/j.knosys.2017.08.017>