

MKR模型在电影推荐系统中的应用研究

操淑敏, 秦 学, 鲜翠琼

贵州大学, 贵州 贵阳

Email: csm_1996@163.com

收稿日期: 2021年2月26日; 录用日期: 2021年4月1日; 发布日期: 2021年4月9日

摘 要

针对电影推荐系统中推荐结果的准确性和可解释性不高的问题, 研究了基于多任务特征学习的知识图谱增强推荐(multi-task learning for knowledge graph enhanced recommendation, MKR)。通过构建知识图谱, 将其作为辅助信息构建了MKR模型, 并将其应用到电影推荐系统中。采用预测用户满意度评分的方法根据评分结果来判定用户对电影的喜好程度, 并将合适的电影类型推荐给用户。最后将MKR模型与几种常见的推荐模型进行比较, 使用不同的评价指标进行预测, 并在top-K场景中比较了不同K值下各个模型的推荐性能。实验结果表明, MKR模型在电影推荐系统中有良好的表现, 在准确率和推荐结果上均优于其他模型, 显著提升了推荐的性能。

关键词

深度学习, 推荐系统, 多任务特征学习, 知识图谱

Research on the Application of MKR Model in Movie Recommendation System

Shumin Cao, Xue Qin, Cuiqiong Xian

Guizhou University, Guiyang Guizhou

Email: csm_1996@163.com

Received: Feb. 26th, 2021; accepted: Apr. 1st, 2021; published: Apr. 9th, 2021

Abstract

Aiming at the low accuracy and interpretability of recommendation results in movie recommendation system, a multi task learning for knowledge graph enhanced recommendation (MKR) based on multi task feature learning is studied. By constructing a knowledge graph, and using it as aux-

iliary information to construct a MKR model, and applied to the movie recommendation system. Using the method of predicting user satisfaction score, according to the score results to determine the user's preference for movies, and recommend the appropriate movie types to users. Finally, MKR model is compared with several common recommendation models, and different evaluation indexes are used for prediction. The recommendation performance of each model under different K values is compared in the Top-k scenario. The experimental results show that MKR model has a good performance in the movie recommendation system, and is superior to other models in accuracy and recommendation results, which significantly improves the performance of the recommendation.

Keywords

Deep Learning, Recommendation System, Multi-Task Feature Learning, Knowledge Graph

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

近年来,推荐系统受到国内外研究者的广泛研究,推荐系统能有效解决信息过载和泛滥问题,用户能在海量的产品和服务中做出正确的选择。目前,主流的推荐系统可分为三种[1]:一是基于协同过滤(collaborative filtering)的推荐算法,协同过滤考虑了用户的历史交互,并根据用户潜在的共同偏好进行推荐[2];二是基于内容的推荐算法,利用项目的内容特征进行推荐;三是混合推荐算法。基于协同过滤的推荐系统能很好的捕获用户偏好,而且易于实现,但基于协同过滤的推荐算法通常会遇到用户-物品交互稀少和冷启动问题,为解决这些局限性,研究者提出混合推荐系统,将边信息整合到协同过滤中,如项目属性[3]、项目评论[4]、用户社交网络[5]等。研究者注意到知识图谱通常包含更丰富的事实和实体之间的联系,项目端的知识图谱拓宽了项目的信息,有利于发现项目之间的潜在关联,提高推荐的精度,增加推荐的多样性。知识图谱还可以将用户的历史记录和推荐记录联系起来,从而为推荐系统带来可解释性。

基于知识图谱辅助的推荐算法的一个主要方法是知识图谱特征学习(knowledge graph embedding, KGE) [6]。知识图谱由三元组构成,由于其网络结构特殊性,使用特征学习可以得到实体和关系的低维向量表示,从而与推荐系统进行交互。MKR 框架的目的就是利用知识图谱任务和推荐任务之间的交互性,用知识图谱来辅助推荐任务[7]。通过框架中的交叉压缩单元模块,对项目 and 实体中有交叉的特征进行建模,优化两个任务之间的交替学习任务,使 MKR 框架充分挖掘了用户和项目之间的特征交互,从而使 MKR 推荐系统的整体性能更优,在图书推荐、新闻推荐等推荐场景中都有良好的表现。本文对基于知识图谱特征学习构建电影推荐系统进行研究探讨,构建了基于多任务学习的知识图谱辅助深度推荐框架(MKR),进行整体框架的训练,提升了电影推荐系统的性能。

2. 电影知识图谱的构建

本文利用知识图谱生成推荐结果、构建电影推荐系统(数据源:豆瓣电影)。首先要构建电影知识图谱,共要经历四个过程,包括电影知识抽取,电影知识表示,以及知识融合和知识推理。构建图谱的数据利用 Python 网络爬虫,使用 requests、bs4 等工具进行爬取。数据爬取流程如图 1 所示,根据对豆瓣电影网

页结构的分析, 针对其动态网页结构, 设计 URL 循环函数, 利用上述爬虫工具爬取数据, 并最终得到初始数据, 并将爬取到的数据存储到数据库中。

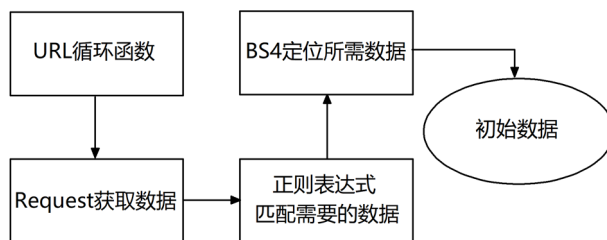


Figure 1. Movie data crawling process
图 1. 电影数据爬取流程

本文共抓取了豆瓣电影中的 20 个类型共 93956 条影片数据, 每个影片包含电影名称、演员、导演、评分、类型等详细信息。对数据进行预处理后, 从数据中抽取所需信息, 将数据导出为 CSV 格式, 并批量创建节点和节点之间的关系。本文共创建了 4 个节点, 分别为电影、演员、导演、电影类型, 一个节点就相当于一个实体。节点间共生成四种关系, 如图 2 所示。

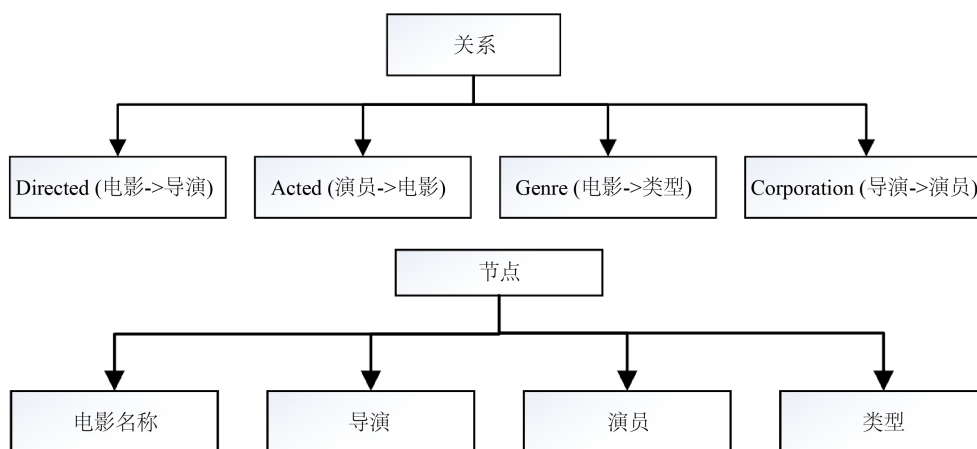


Figure 2. Node and relational data structure
图 2. 节点、关系数据结构

对数据进行初步处理后, 将其导入图数据库 neo4j 中。在 neo4j 的浏览器窗口执行 load CSV 命令, 将节点、关系批量导入图数据库中。部分结果如图 3 所示。

3. 基于知识图谱特征学习的电影推荐模型

3.1. 知识图谱特征学习

将知识图谱特征学习辅助推荐系统学习有三种方式: 依次学习、联合学习、交替学习。依次学习法学习知识图谱中的实体向量和关系向量。联合学习结合推荐算法和知识图谱特征学习, 优化了端到端的训练。交替学习利用知识图谱特征学习任务来辅助推荐任务, 因为推荐系统中的项目可能与知识图谱中的实体相关联, 因此项目及其对应的实体可能具有相似的邻近结构, 在底层和非任务特异性潜在特征空间上可能具有相似的特征[8]。知识图谱特征学习是为了在保留原有的结构信息的同时, 让每个实体和向量都得到低维的特征表示, 其中基于翻译的方法广为应用, 主要代表有 TransE、TransH、TransR 模型。

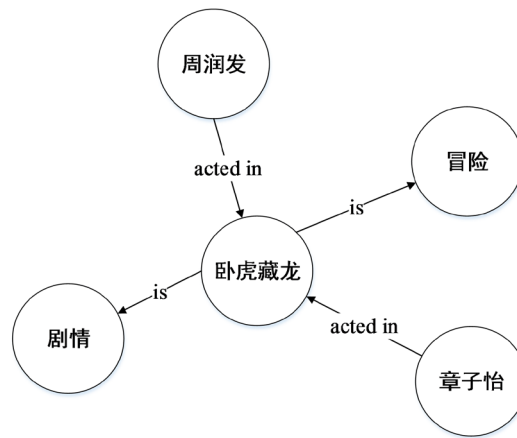


Figure 3. Partial display of the movie knowledge graph
图 3. 电影知识图谱部分展示

在本文的电影推荐系统构建中，记 M 个用户 $U = \{u_1, u_2, \dots, u_M\}$ 和 N 个项目集合 $V = \{v_1, v_2, \dots, v_N\}$ ，记 U 和 N 的交互矩阵为 $Y \in R^{M \times N}$ ，用户的隐式反馈数据构成交互矩阵， $y=1$ 为有交互记录，例如观看，评论行为，否则 $y=0$ 。记知识图谱 $G(\mathcal{E}, \mathcal{R})$ 由三元组 $g(h, r, t)$ 构成，其中 h 表示头节点、 r 表示 h 和 t 的关系、 t 表示尾节点，其中 \mathcal{E} 和 \mathcal{R} 表示知识图谱的实体和关系的集合。例如图 3 所展示，三元组(周润发，主演，卧虎藏龙)的映射即为知识图谱中所示的周润发是电影《卧虎藏龙》的主演。得到集合 Y 和图谱 G 后，可以计算出其预测函数为 $\hat{y}_{uv} = F(u, v | \Theta, Y, G)$ ，得到用户 u 与未交集的物品 v 的交互概率。

3.2. MKR 电影推荐模型

通过研究基于多任务学习的知识图谱辅助推荐系统框架(MKR 推荐系统框架)，本文使用交替学习法来交替学习知识图谱特征学习任务 and 推荐系统任务。利用推荐模型中的交叉压缩单元模块，学习并实现两个任务中的实体之间的高阶交互，并抽取两个任务的低层特征进行融合训练，从而达到更优的结果。

图 4 指出了 MKR 模型的 3 个部分：推荐系统部分、知识图谱嵌入部分、交叉压缩单元部分。左边推荐部分将用户和电影作为输入，提供用户对电影的显式评分，使用多层感知机(Multi-Layer Perception, MLP)和交叉压缩单元从用户和物品中学习特征，模型预测结果为用户对该电影的交互预估值，数值为 0~1。右边知识图谱特征学习模块将实体和关系嵌入到低维的连续向量空间里，与此同时保证实体和关系的结构稳定。此模块将 2.1 中三元组 (h, r, t) 的头节点 h 和关系 r 作为输入，使用多层感知机学习头节点和关系的特征，预测尾节点(目标实体)特征，将其作为输出。

图 5 指出了模型中的交叉压缩单元模块，其主要利用知识图谱中的实体和推荐系统中的物品的同一性，使物品和实体的表示可以相互补充，不仅提高了泛化能力，并且避免了噪声拟合。

交叉操作是对物品 v 和与其相关的实体 e 的任一可能的特征交互进行建模，根据特征构建交互矩阵：

$$C_l = v_l e_l^T = \begin{bmatrix} v_l^{(1)} e_l^{(1)} \cdots v_l^{(1)} e_l^{(d)} \\ \cdots \\ v_l^{(d)} e_l^{(d)} \cdots v_l^{(d)} e_l^{(d)} \end{bmatrix} \quad (1)$$

式(1)中 $C_l \in R^{d \times d}$ 表示交叉压缩单元的第 l 层交叉特征矩阵， d 表示交互矩阵的特征维度。 v_l 和 e_l 分别为第 l 层物品和实体的特征。压缩操作中的权值向量重新使交叉特征矩阵由 $R^{d \times d}$ 的空间指向特征空间 R^d ，将特征交叉矩阵投影到下一层的潜在表示空间，输出下一层的项目和实体的特征向量：

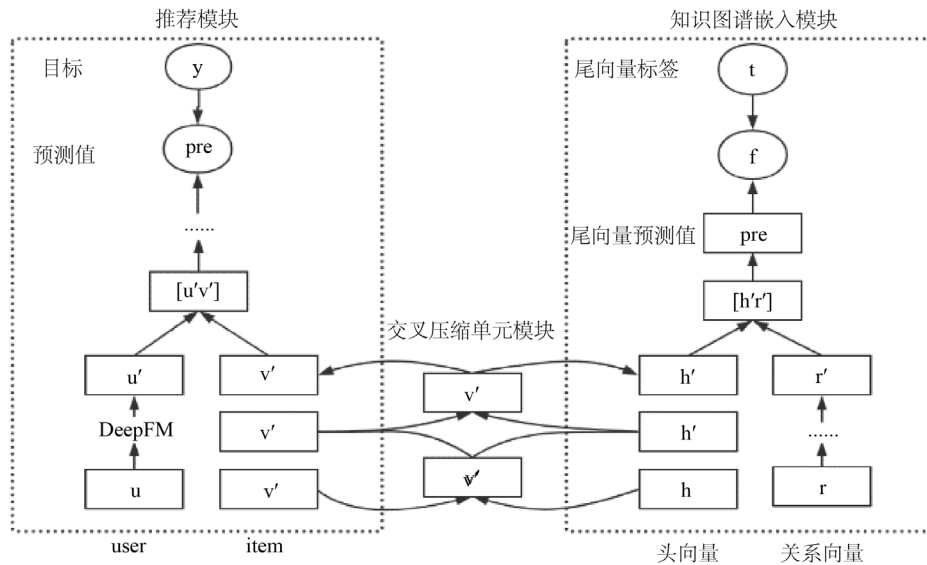


Figure 4. MKR model framework
图 4. MKR 模型框架

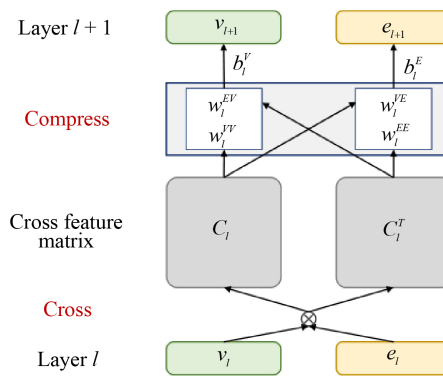


Figure 5. Cross compression unit
图 5. 交叉压缩单元

$$\begin{aligned}
 v_{i+1} &= C_l w_i^{VV} + C_l^T w_i^{EV} + b_i^V = v_i e_i^T w_i^{VV} + e_i v_i^T w_i^{EV} + b_i^V \\
 e_{i+1} &= C_l w_i^{VE} + C_l^T w_i^{EE} + b_i^E = v_i e_i^T w_i^{VE} + e_i v_i^T w_i^{EE} + b_i^E
 \end{aligned}
 \tag{2}$$

式(2)中 w 和 b 分别是训练权重和偏置参数。在本数据集中，根据用户和电影的交互历史数据，按评分对特征向量进行加权处理。评分的范围为 1~5 分，将评分为 4 分以下的赋予负值，然后经过加权得到特征向量的权重。交叉压缩单元只存在于模型的底层，通过在底层将推荐模型和知识图谱特征学习模型连接起来，实现两个任务之间知识的传递，从而有效的改善了信息的稀疏性。

3.3. MKR 模型训练

MKR 模型的训练采用多任务框架交替学习的方法进行特征建模和预测。其完整的损失函数由三部分组成：推荐系统部分、知识图谱特征学习部分、正则化项。

$$\begin{aligned}
 L &= L_{RS} + L_{KG} + L_{REG} \\
 &= \sum_{u \in U, v \in V} J(\hat{y}_{uv}, y_{uv}) - \lambda_1 \left(\sum_{(h,r,t) \in G} score(h,r,t) - \sum_{(h',r',t') \in G} score(h',r',t') \right) + \lambda_2 \|W\|_2^2
 \end{aligned}
 \tag{3}$$

式(3)中 u 和 v 分别遍历用户集和项目集, J 为交叉熵函数, 此函数收敛。知识图谱部分为三元组的得分函数, 选取的计算方式是向量内积, 此函数收敛。正则化项是为了防止过拟合, 其中 λ_1 和 λ_2 为超参数, λ_1 可视作两项任务的两个学习率之比, 数值为负, 从而确保知识图谱模块的特征学习性能。 W 为式(2)描述的训练权重。

MKR 算法训练如算法 1 所示。获取电影数据集和用户评分数据集作为输入, 根据用户物品交互记录, 按评分对物品向量进行加权, 得到对应的用户特征向量, 采用多任务学习框架, 交替训练推荐模块和知识图谱嵌入模块。通过交叉压缩单元模块实现推荐系统中物品和知识图谱中实体之间的信息共享, 输出为物品潜在特征表示。通过计算内积的方法融合电影特征向量和用户特征向量, 得到向量相似度, 并以此为依据对用户行为进行预测, 向用户推荐偏好电影, 实现推荐。

算法 1 MKR 训练算法

输入: 交互矩阵 Y , 知识图谱 G

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

1. 初始化所有参数
 2. for 训练总轮数 do
 3. for t 步 do
 4. 从 Y 中采样正负交互的一个批量(minibatch);
 5. 为该批量中的每个物品 v 采样 $e \sim S(v)$;
 6. 根据交叉压缩单元计算公式(1)-(3), 用梯度下降法更新 \mathcal{F} 的参数;
 7. end for
 8. 从 G 中采样真假三元组的一个批量;
 9. 为该批量中的每个头节点 h 采样 $v \sim S(h)$;
 10. 使用梯度下降法更新 \mathcal{F} 的参数;
 11. end for
-

4. 实验结果与分析

4.1. 数据集处理

本文所选数据集包含 93956 条影片数据, 199814 个用户, 2706200 次用户与电影的交互记录评分。将知识图谱和电影数据集作为输入时, 需保持数据格式的统一, 要对传入的数据进行预处理。对数据的预处理环节中, 对电影 ID、关系、评分等数据进行提取和转化, 处理缺失值, 使数据格式电影符合知识图谱中的数据结构标准。图 6 展示了数据集的整体处理流程。

首先将评分数据集划分为训练集和测试集两类, 其中训练集占 80%, 测试集占 20%, 再针对训练集构建电影评分矩阵, 根据评分公式计算评分 S 。用户对电影的满意程度随评分(1~5 分)的增长而提高。若评分 S 大于等于 4, 此时的数据对应正样本, 则推荐; 若 S 小于 4 此时的数据对应负样本, 则不推荐。利用测试集分别测试几种推荐模型, 对比算法性能并选择性能最佳的算法。

4.2. 评价指标

本文使用准确率(Accuracy, ACC)和 ROC 曲线线下面积(Area Under Curve, AUC)作为评价推荐的性能指标。在推荐准确度上, 将评分推荐分为推荐和不推荐, 即评分 1 到 3 分为不推荐, 4 到 5 分为推荐。

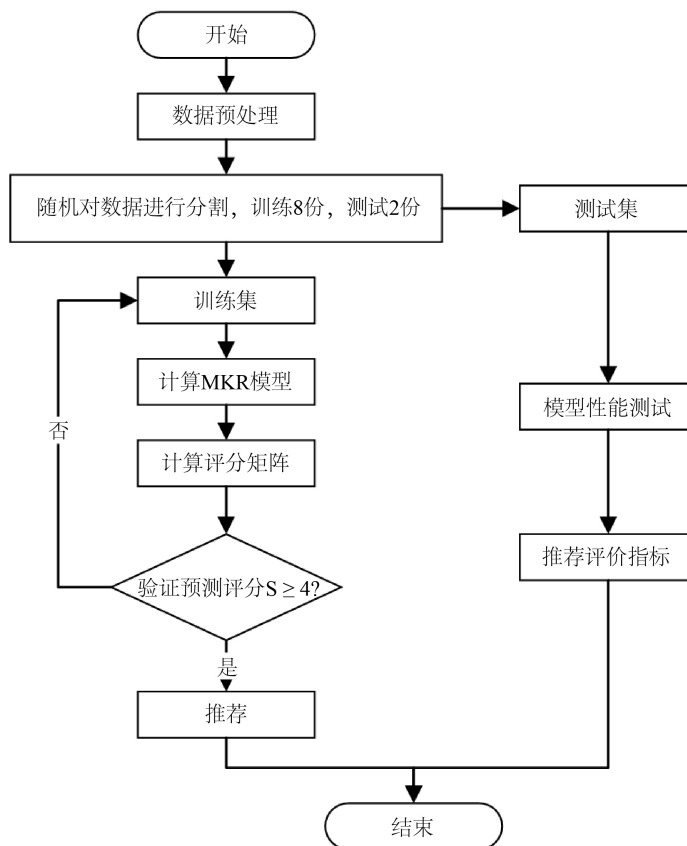


Figure 6. Data set processing flow
图 6. 数据集处理流程

ACC 指标在预测用户是否喜欢时，通过预测正确的分类比例，从而描述推荐模型的推荐性能，基于表 1 的混淆矩阵，计算公式如下：

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \quad (4)$$

AUC 用于二分类模型的评估，作为 ROC 曲线下边的面积，用来评估分类器的分类质量。根据评分数据处理后得到的正样本和负样本，计算正样本的置信度大于负样本置信度的概率，比较置信度以得到最终的 AUC 值。由于阈值变化影响分类结果，使用 AUC 则可以考虑到阈值变动，所以评估的效果更好。

Table 1. Confusion matrix
表 1. 混淆矩阵

系统	用户喜欢	用户不喜欢
推荐	TP	FP
不推荐	FN	TN

4.3. 结果分析

将训练好的用户特征矩阵和电影推荐矩阵计算所有电影的评分，进行电影推荐，图 7 展示了本数据集在 MKR 模型下返回的部分推荐结果。

```

为用户ID为: 975874的用户推荐的电影为:
电影ID: '15124', 推荐分: 32.00000000000014

为用户ID为: 50123的用户推荐的电影为:
电影ID: '11607', 推荐分: 36.129588500101796

为用户ID为: 1709381的用户推荐的电影为:
电影ID: '11521', 推荐分: 33.00000000000014

为用户ID为: 484464的用户推荐的电影为:
电影ID: '14313', 推荐分: 31.00000000000018

为用户ID为: 2133037的用户推荐的电影为:
电影ID: '118123', 推荐分: 26.999999999999996

```

Figure 7. Movie recommendation results

图 7. 电影推荐结果

另外, 为了验证 MKR 模型的高效性, 将此模型与 LibFM [9]、PER [10]、DKN [11]、CKE [12]和 Wide & Deep [13]推荐算法进行对比, 用该数据集分别测试各模型, 计算 ACC 和 AUC 值, 并取 10 次实验所得数据的平均值作为最终结果, 如表 2 所示。

Table 2. Comparison of the recommendation effect of each recommendation algorithm on the movie dataset

表 2. 各推荐算法在电影数据集上推荐效果对比

模型	ACC	AUC
LibFM	0.825	0.903
PER	0.677	0.736
CKE	0.731	0.803
DKN	0.632	0.716
Wide&Deep	0.813	0.860
MKR	0.825	0.903

从表 2 结果可知, 相比其他几种模型, MKR 模型在本数据集中取得了最佳的表现, ACC 和 AUC 的计算结果都超越了对比算法, 由此可知 MKR 模型使推荐效果有了显著提升。

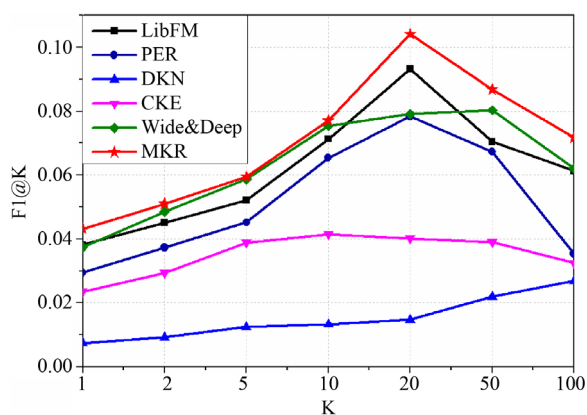


Figure 8. The results of the top-K movie recommendation comparison experiment

图 8. top-K 电影推荐比较实验结果

为进一步验证模型的性能, 本文进行了 top-K 场景实验, 使用训练好的模型在测试集中为用户推荐电影, 设置推荐电影数目为 K 值, 使用 F1@K 值作为 top-K 推荐中的性能指标。分别在不同 K 值场景下计算 F1@K 值, 并取 10 次计算结果的平均值作为最终结果, 如图 8 所示。

从图 8 中总体来看, 随着 K 值的增加, 各模型的 F1@K 值呈先增大后减小的趋势, 当 K = 20 时, MKR、LibFM、DKN 模型的 F1@K 值达到峰值; 并且 MKR 模型在不同 K 值的场景下比其他几种模型的表现结果都更好一些, 证明该模型中的交叉压缩单元能有效学习用户和实体之间的知识交互, 从而有效提升了推荐的整体性能。

5. 结语

本文通过构建电影知识图谱, 将其作为辅助信息, 构建了 MKR 推荐模型, 并将其应用到电影推荐系统。采用预测用户满意度评分的方法优化了推荐性能, 通过评分结果来判定用户对电影的喜好程度, 并将合适的电影类型推荐给用户。通过大量实验对比了 MKR 模型与几种常见推荐模型, 采用 AUC 和 ACC 两种评价指标来预测分值, 并且在 top-K 场景中比较了不同 K 值下各个模型的推荐性能。从实验结果来看, MKR 模型的准确率和推荐效果明显优于其他几种模型, 并且有效提升了推荐的性能。后续将考虑使用注意力机制融合知识图谱, 重新设计交叉压缩单元, 挖掘更多的电影信息; 另外本文是在单机上进行的测试, 如果数据量过大势必会影响推荐性能, 因此下一步考虑采用多 GPU 并行的方式来实现该模型的并行计算, 从而进一步提高推荐性能。

参考文献

- [1] Adomavicius, G. and Tuzhilin, A. (2005) Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, **17**, 734-749. <https://doi.org/10.1109/TKDE.2005.99>
- [2] Gao, Y., Li, Y.F., Lin, Y., Gao, H. and Khan, L. (2020) Deep Learning on Knowledge Graph for Recommender System: A Survey. *Information Retrieval*, arXiv:2004.00387.
- [3] Zhao, H., Yao, Q.M., Li, J.D., Song, Y.Q. 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*, 635-644. <https://doi.org/10.1145/3097983.3098063>
- [4] 宋瑞平. 混合推荐算法的研究[D]: [硕士学位论文]. 兰州: 兰州大学, 2014.
- [5] Koren, Y., Bell, R. and Volinsky, C. (2009) Matrix Factorization Techniques for Recommender Systems. *Computer*, **42**, 30-37. <https://doi.org/10.1109/MC.2009.263>
- [6] Guo, Q., Zhuang, F., Qin, C., et al. (2020) A Survey on Knowledge Graph-Based Recommender Systems. *Scientia Sinica Informationis*, **50**, 937. <https://doi.org/10.1360/SSI-2019-0274>
- [7] Wang, H., Zhang, F., Zhao, M., et al. (2019) Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. *The World Wide Web Conference*, 2000-2010. <https://doi.org/10.1145/3308558.3313411>
- [8] 涂存超, 杨成, 刘知远, 等. 网络表示学习综述[J]. *中国科学: 信息科学*, 2017, 47(8): 980-996.
- [9] Rendle, S. (2012) Factorization Machines with libFM. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **3**. <https://doi.org/10.1145/2168752.2168771>
- [10] Xiao, Y., Xiang, R., Sun, Y., et al. (2014) Personalized Entity Recommendation: a Heterogeneous Information Network Approach. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*.
- [11] Wang, H., Zhang, F., 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*. International World Wide Web Conferences Steering Committee, 1835-1844. <https://doi.org/10.1145/3178876.3186175>
- [12] Zhang, F.Z., Yuan, N.J., Lian, D.F., Xie, X. and Ma, W.Y. (2016) Collaborative Knowledge Base Embedding for Recommender Systems. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 353-362. <https://doi.org/10.1145/2939672.2939673>
- [13] Cheng, H.T., Koc, L., Harmsen, J., et al. (2016) Wide & Deep Learning for Recommender Systems. *Wide & Deep Learning for Recommender Systems*.