

基于评分数值分析的用户项目质量测度及其在深度推荐模型中的应用*

郑 潇 李树青 张志旺

(南京财经大学信息工程学院 南京 210023)

摘要:【目的】利用用户项目评分信息和时间段信息,设计一种反映用户项目质量的有效特征,并结合深度推荐模型改善推荐效果。【方法】从用户评分和项目质量一致性、评分数值分布和基于时间段的评分数值分布三个方面,系统地提出对用户和项目质量特征进行评估的有效方法。【结果】在 MovieLens 数据集上进行测试,与传统的深度推荐模型相比,本文所提模型的 MAE 和 MSE 分别最高提升达到 3.71% 和 4.24%。【局限】现有改进主要结合评分信息,没有探索包括用户项目更多属性特征的质量指标评定方法,在一定程度上影响了推荐效果。【结论】本文所提推荐模型评分预测效果更加准确,缓解了深度推荐模型中数据有效特征缺乏和表现力不足等问题,提高了推荐质量。

关键词: 推荐系统 深度学习 用户项目质量 有效特征

分类号: TP393 G250

DOI: 10.11925/infotech.2096-3467.2021.0683

引用本文: 郑潇, 李树青, 张志旺. 基于评分数值分析的用户项目质量测度及其在深度推荐模型中的应用[J]. 数据分析与知识发现, 2022, 6(4): 39-48. (Zheng Xiao, Li Shuqing, Zhang Zhiwang. Measuring User Item Quality with Rating Analysis for Deep Recommendation Model[J]. Data Analysis and Knowledge Discovery, 2022, 6(4): 39-48.)

1 引言

近年来,信息技术迅猛发展,互联网将人们带进了一个信息爆炸的时代,用户如何从海量信息中选取有价值的信息成为新的难题。为解决这个问题,推荐系统通过个性化的推荐算法,将不同的人 and 物品联系起来,分析用户的行为属性,挖掘不同用户间的共同爱好,为其精准推送个性化信息或物品,从而节省用户大量时间。

在大量推荐算法中,协同过滤算法的应用最为普遍。协同过滤依赖于偏好或兴趣与目标用户相似的用户,并推荐用户可能感兴趣的项目。由于传统

的协同过滤算法的实现非常依赖物品和用户的评分信息,在用户评分数据稀疏的情况下推荐效果往往不够好,所以越来越多的研究针对用户和项目自身的信息并挖掘出其隐含的特征信息进行数据增强。为更加充分地利用用户和项目的信息,提高向用户推荐物品的准确率,本文提出融合直接特征信息和隐含特征信息的深度学习方法,考虑了用户和项目质量特征信息的作用。本文相关工作如下:

(1) 提出一种将用户和项目的直接和隐含特征信息相融合的模式,使用卷积神经网络提取用户和项目的特征信息,用不同的方式拟合出用户的预测评分;

通讯作者(Corresponding author): 李树青(Li Shuqing), ORCID: 0000-0001-9814-5766, E-mail: leeshuqing@163.com。

*本文系江苏省高等学校自然科学研究重大项目(项目编号: 19KJA510011)的研究成果之一。

The work is supported by Major Natural Science Research Projects of Colleges and Universities in Jiangsu Province of China (Grant No.19KJA510011).

(2)提出一种基于用户评分和项目质量一致性的研究方法,利用用户评分和项目质量的一致性来定义质量;提出一种基于评分数值分布的质量测度方法,利用Beta概率分布分配权重,对用户和项目的质量进行评估;提出一种基于时间段的评分数值分布质量测度方法,在评分数值分布的质量研究方法的基础上引入评分稠密时间段,提高推荐算法的效率和准确性。通过实验验证了以上三种方法对比已有的质量研究方法都有不同程度的提高。

2 相关研究

2.1 深度学习在推荐系统中的应用

传统推荐算法无法完全掌握用户每个方面的兴趣和需求,因为用户之前没有对足够多类别的对象进行评价,导致推荐系统缺乏可扩展性。但是深度学习技术可以有效地处理大规模数据,很好地缓解了推荐系统的可扩展性问题^[1]。刘凯等^[2]对深度学习在推荐系统的应用进行系统化的综述,指出深度学习与传统信息推荐算法的结合,主要是利用深度神经网络所提取的高层特征对项目或用户进行聚类,缩小邻域空间范围,或者将特征作为推荐算法训练目标函数的正则化因子,参与到模型训练过程,以此提高信息推荐算法建模效率。近年,使用深度学习的方法将特征信息融入推荐系统的研究逐渐增多,如文献[3]将一种新的压缩交互网络(Compressed Interaction Network, CIN)和经典的深度神经网络(Deep Neural Networks, DNN)结合为xDeepFM模型,该模型可以自动生成直接特征和隐含特征向量,并进行交互,不仅可以直接生成一定阶数内的特征交互关系,而且可以隐式地生成任意低阶和高阶的特征交互,缓解了由于数据多样性导致的手动生成交互特征成本比较高的问题,但由于CIN模块的时间复杂度很高,影响了模型的性能。Kim等^[4]提出卷积矩阵分解模型(ConvMF),该模型将能够提取文档上下文信息的卷积神经网络(Convolutional Neural Networks, CNN)与概率矩阵分解(Probabilistic Matrix Factorization, PMF)结合,对比传统模型,性能得到了显著提升。Filho等^[5]提出DeepRecVis模型,该模型使用卷积神经网络从电影预告片的关键帧中提取特征来表示项目,并在基

于内容的推荐系统中利用这些特征,实验表明该模型的性能优于基于低层次特征表示的系统。Messina等^[6]利用深度神经网络(DNN)提取的视觉特征来推荐数字艺术,通过对神经视觉特征以及人工设计的视觉特征(如自然度、亮度和对比度)的研究,发现基于深层神经视觉特征和人工设计视觉特征的混合推荐模型效果最好,为基于内容的艺术作品推荐领域做出了贡献。可见,通过融合直接特征和隐含特征信息的深度神经网络在不同领域下进行推荐的模型的研究价值不断被发掘,相关技术在不断完善。

2.2 融合特征信息的推荐方法

当前推荐系统通常可以分为三类:基于内容的推荐算法、基于协同过滤的推荐和混合推荐^[7]。基于内容的推荐算法往往通过用户或项目的特征信息进行推荐,基于协同过滤技术的推荐算法大多使用用户对项目的评分数据作为数据源进行推荐,而混合推荐算法结合其他两种方法的优势,往往能够有效解决评分数据稀疏和冷启动问题。使用用户和项目的特征信息的深度混合推荐算法来缓解评分数据稀疏和冷启动问题也是近年来推荐系统领域研究的热点之一。通过深度学习方法和特征信息结合进行推荐的研究,主要分为以下两类:

(1)关于利用卷积神经网络获取特征的方法。如邓存彬等^[8]考虑用户特征和电影特征,利用深度学习模型中的卷积神经网络(CNN)和多层感知机(Multilayer Perceptron, MLP)获取深层次的表示特征,在提高电影评分预测的准确性上取得了不错的效果,但是也存在算法复杂、可解释性差以及参数较多等问题。在基于文档信息的推荐任务中,针对评分数据变得庞大且复杂时,其推荐性能往往不太理想的问题,胡朝举等^[9]提出一种深度融合模型,使用两个并行的神经网络分别获取用户和项目的潜在特征向量,实验结果表明该模型能够在完全捕获文本信息的同时很好地处理复杂且稀疏的评分数据,但是该模型只使用了直接数据信息而未使用隐含数据信息。

(2)关于将深度学习技术和协同过滤算法结合的方法。如Wang等^[7]将项目评价文本通过深度学习引入矩阵分解中,设计协同深度学习

(Collaborative Deep Learning, CDL)模型,利用降噪自动编码器,将矩阵分解中的项目隐因子变为项目个性化偏置项与项目评价文本编码的线性和,最后通过最大化评分后验概率实现矩阵分解(Matrix Factorization, MF)与去噪自动编码器(Denoising AutoEncoder, DAE)参数的调节。但CDL模型不像卷积神经网络可以显式地考虑上下文和单词的顺序,影响了其性能。李涛等^[10]提出基于协同过滤算法的自动化评分音乐双重推荐系统,将隐含特征信息提取器与推荐引擎结合,设立双重推荐机制实现抗人为影响的音乐双重推荐系统,该系统对于大规模音乐推荐具有良好可扩展性,但是没有考虑硬件交互协议的不对称性问题。

上述研究虽然从不同角度利用深度学习和特征信息结合的方法缓解数据稀疏性问题,以此改进推荐的准确率,不过仍存在一些问题。首先,在特征提取时没有将特征信息完全提取,只是简单地提取部分直接特征然后将特征向量化输入模型。其次,关于直接或隐含特征对推荐效果有何影响的研究很少,忽略了各种特征信息对推荐效果提升程度方面的研究。

用户对一个物品的评分是多种因素的结果,并不一定代表用户对物品真正感兴趣,很可能是对质量较高的物品做出的客观评价,或是用户习惯了给出高分,只有遇到质量较差的物品才给出低分。用户本身存在信誉度这一反映用户质量的属性,即用户是否能够客观地对项目进行评价,是否一律给出高分或低分;项目本身也存在质量这一属性,项目质量高低直接或间接影响用户对该项目的评分。有学者通过迭代计算用户的历史评分与项目内在质量的相关性来获得用户的信誉度,有效地去除用户评分不严格带来的自然噪声,加强高信誉度用户影响的效果,提高了推荐的准确性^[11],但是仅使用项目的平均分反映项目的内在质量;还有学者提出一种信誉感知推荐系统,通过信誉感知的方法过滤低质量项目,提高推荐的准确性^[12],但在处理大规模数据集时也表现出局限性。

针对上述问题,在已有对用户项目质量研究的基础上,本文利用深度学习方法,将直接特征信息和使用不同方法评估的用户项目的隐含质量特征融入

推荐模型中,提高预测评分的准确度。

3 深度融合模型

推荐系统中,在没有给项目评分之前,用户完全凭借自己的兴趣爱好选择的项目就能代表用户的潜在兴趣。比如在生活中,人们看一部电影往往不是因为听别人说这部电影非常好看才去看,也不只是因为这部电影的评分很高就去看,更多情况下是这部电影是自己喜欢的类型才去选择性观看,用户给出的低评分往往是由于不喜欢这部电影的某一方面,如电影的演员或者叙事方式等,但是对电影本身的类型依然可以认为属于喜欢。根据这种思想,本文提取用户和电影的直接和隐含特征信息,把这些特征用深度学习的方法来拟合预测出用户的评分。

3.1 模型结构

本文所提深度融合用户项目质量特征信息模型主要通过使用不同的深层网络来提取用户和项目直接和隐含的特征信息,输出用户预测评分,从而达到提升预测评分准确率的效果。模型结构如图1所示,为获得用户的特征,将用户的直接特征信息输入嵌入层映射为低维的用户嵌入向量,用户嵌入向量经过多个全连接层从而层层映射到用户特征向量。在获取项目特征时,将项目ID、类型以及用户和项目的质量特征输入嵌入层映射为低维的项目嵌入向量,再经过多个全连接层从而层层映射到项目特征向量。得到这两个特征以后,再输入融合层,最后则可以拟合预测的评分。

本文使用图1中的方式,将两个特征做向量乘法,结果与真实评分做回归,再通过构建融合层进一步将用户特征和项目特征进行融合,最后通过不同的评价指标来评估模型。

在一个推荐系统中,用户的直接特征指用户本身存在的属性,包括用户的ID、性别、年龄、职业、邮编以及评分的时间等。项目的直接特征指项目本身存在的属性,包括项目的ID、名称及其类型。用户和项目的隐含特征不能直接从数据集中获取,但是可以通过不同方式去获取和量化。用户的隐含特征可以是用户自身的信誉度即用户的质量,由于部分用户存在不良的评分习惯,在评价项目时只给出最高分或者最低分,但是这并不能说明该用户是否喜

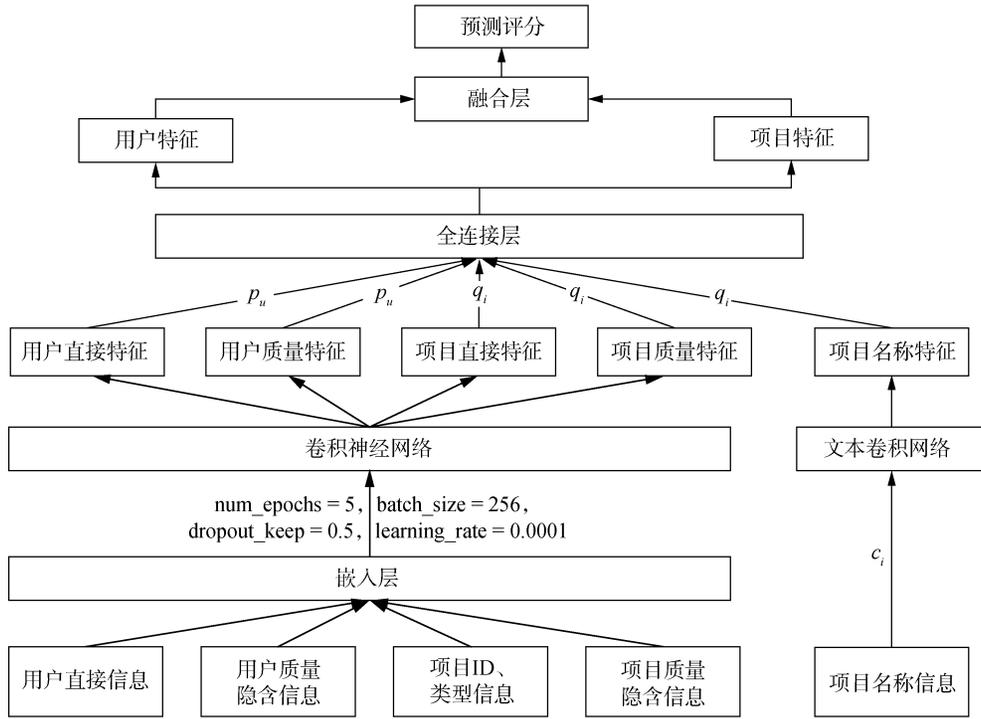


图1 融合用户项目质量特征的深度推荐模型

Fig.1 Deep Recommendation Model Based on User and Item Quality Features

欢该项目。为考虑用户评分的有效性,应当考虑用户和项目的质量这一隐含特征信息。

(1) 用户特征提取

假设有 M 位用户 $U = \{u_1, u_2, \dots, u_M\}$ 和 N 个项目 $I = \{i_1, i_2, \dots, i_N\}$, 模型使用多层的神经网络结构建模用户的潜在特征向量 p_u , 可以更好地学习用户的行为偏好。如图1所示, 下一层的信息输出到上一层作为输入, 因此使用用户-项目交互矩阵作为输入, 该输入通过嵌入层映射为低维的用户嵌入向量, 然后经过多个全连接层, 包括嵌入层和感知器层, 映射到用户特征向量 p_u 。

(2) 项目特征提取

项目偏好提取结构如图1右半部分所示, 其中提取项目直接特征和隐含的质量特征的过程与用户特征提取过程类似。使用CNN^[13]对项目名称进行特征提取操作。假设 $X_i \in R^k$ 是句中第 i 个词对应的 k 维词向量, 那么长度为 n 的句子表示如公式(1)所示。

$$X_{1:n} = X_1 \oplus X_2 \oplus \dots \oplus X_n \quad (1)$$

经过拼接、卷积运算和特征映射等操作之后, 得

到的特征词 C_i 可以由单词 X_{i+h-1} 产生, 操作过程如公式(2)和公式(3)所示。

$$C_i = f(W \cdot X_{i+h-1} + b) \quad (2)$$

$$c = [c_1, c_2, \dots, c_{n-h+1}] \quad (3)$$

对特征映射进行最大池化操作, 取最大值 $q_i = \max \{c\}$ 作为与特定过滤器相匹配的特征。在网络的第三层使用Dropout正则化, 防止过拟合。

(3) 构建特征融合层

用户特征向量 p_u 和项目特征向量 q_i 均已得到, 构建融合层对两种特征向量进行融合从而输出预测值, 融合层的输出定义如公式(4)所示。

$$s_{ui} = \omega(W^T(p_u \circ q_i)) \quad (4)$$

其中, $\omega(\cdot)$ 表示 Sigmoid 激活函数, W 表示权重矩阵, 通过激活函数增加非线性因素, 使得模型能更好地学习用户之间的非线性联系, 提高模型的准确率。

3.2 用户项目质量特征分析

(1) 基于用户评分和项目质量一致性的用户质量测度方法

本文提出一种 RAUIQ_CORR (Recommendation

Algorithm based on User and Item Quality_CORR)方法,即基于用户评分和项目质量一致性的深度融合用户项目质量特征的算法,将提取出的用户和电影特征做向量乘法,得到的预测评分与真实评分做回归。结合文献[14]方法,用户质量由用户的评分与对应项目的内在质量的相关系数决定,通过迭代计算用户的历史评分与项目自身质量的相关性获得用户质量。用户 u 的质量最初被定义如公式(5)所示。

$$R_u = |r(u)/n| \quad (5)$$

其中, $r(u)$ 表示用户 u 给出评分的项目集合。

在实际应用中,项目质量通常以用户对项目的评分来反映,常见的物品质量分析方法有平均分、贝叶斯估计法^[15]以及 HITS 算法^[16]等。其中最简单的方式就是利用用户对项目的平均评分,平均评分可以反映用户对物品评分的总体情况,但是这种方法在面向评分较少的项目时存在项目真实质量往往难以衡量、相关推荐算法易于被攻击等问题。由于要进行大量迭代计算,导致 HITS 算法性能较低,并且机制上容易被作弊者操纵。

本文采用评分标准差^[17]来反映项目质量。评分标准差即离散性反映了项目自身质量属性对用户评分的影响程度,这比仅用平均分反映项目的内在质量更加准确。用户评分的离散程度越高,项目自身质量属性对用户的评分影响越大,越能反映用户的兴趣偏好;反之,项目自身质量属性对用户的评分影响越小,能够反映出用户兴趣偏好的程度越低。因此,为消除用户评分的离散性和项目自身质量属性带来的误差,使用用户评分标准差反映项目的质量,如公式(6)所示。

$$Q_i = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (I_n - \bar{\sigma}_i)^2} \quad (6)$$

其中, N 表示项目 i 的评分个数, $N-1$ 表示自由度, I_n 表示项目 i 的第 n 个评分, $\bar{\sigma}_i$ 表示项目 i 的平均评分。用户 u 和项目 i 内在质量相关性如公式(7)所示。

$$Corr_{ui} = \frac{\sum_{i \in r(u)} Q_i Q_u}{\sum_{i \in r(u)} Q_i \sum_{i \in r(u)} Q_u} \quad (7)$$

其中, Q_i 表示项目 i 的评分标准差, Q_u 表示用户 u 给出评分的标准差。

从公式(7)可以看出,如果用户对项目的评分接

近其内在质量,则用户与项目内在质量之间的相关性很高。在某种程度上,高相关性表明用户专业严谨,否则,代表该用户并不能够客观公正地给出评分。在考虑用户评分质量时,用户评分的数量对推荐模型的支持度分析很有必要,根据文献[17]中数据数量敏感的推荐模型的思想,因为推荐算法需要充分考虑用户行为数据在数据数量方面的差异性,即推荐方法对数据量具有敏感性,所以无须将用户的所有行为数据用于推荐建模,并且在推荐建模的同时应该充分考虑数据数量的差异性。

(2) 基于评分数值分布的用户质量研究方法

本文提出一种 RAUQ_Beta (Recommendation Algorithm based on User Quality_Beta Distribution)方法,即基于评分值分布的用户质量算法。由于每个用户评分习惯不同,给出的评分在不同程度上反映该用户的质量,本文在原有的使用标准差反映项目质量的方法基础上进行改进,使用 Beta 分布的概率密度函数为一个用户的评分生成权重,然后通过加权平均聚集评分生成用户的质量分数。若该用户打出的某项评分频繁极端,那么该用户的质量分数则会偏低。将生成的用户质量分数作为反映用户质量的属性特征。Beta 分布的概率如公式(8)所示。

$$Beta(x_i) = \frac{\gamma(\alpha + \beta)}{\gamma(\alpha)\gamma(\beta)} x_i^{\alpha-1} (1 - x_i)^{\beta-1} \quad (8)$$

其中, α 和 β 是决定分布形状的两个参数, x_i 定义如公式(9)所示。

$$x_i = \frac{p_1 \times i}{n - 1} + p_2 \quad (9)$$

其中, $i = 0, 1, \dots, n - 1$, 当 p_1 取 0.98, p_2 取 0.01 时,可保证公式(9)的 $x_0 = 0.01, \dots, x_{n-1} = 0.99$,使得累加和为 1。生成归一化权重,最后生成用户质量分数如公式(10)和公式(11)所示。

$$BetaDQ_u = \sum_{l=1}^k (l \times LW^l) \quad (10)$$

$$LW^l = \sum_{j=0}^{|R^l|-1} w_j^l \quad (11)$$

为确定形状分布参数 α 和 β ,结合文献[18]使用 IRR (Item Rating Relative Count, 即一个项目的评分计数和一个集合中这个项目所有的平均评分计数之间的比率),引入用户评分相对计数 URRC (User

Rating Relative Count, 即一个用户的评分计数和一个集合中这个用户所有的平均评分计数之间的比率), 以及评分标准差来确定分布的形状。URRC 定义如公式(12)和公式(13)所示, α 和 β 计算方法如公式(14)所示。

$$URRC = \frac{n_i}{\bar{n}} \quad (12)$$

$$\bar{n} = \frac{\sum_{u_i \in M} n_i}{|M|} \quad (13)$$

$$\alpha = \beta = \begin{cases} (\frac{\mu}{\sigma})^2 \times URRC, & \sigma \neq 0 \\ URRC, & \sigma = 0 \end{cases} \quad (14)$$

其中, n_i 是用户 u_i 的评分计数, \bar{n} 表示域中用户的平均评分计数, M 是域中所有的用户集。 μ 和 σ 表示用户的平均评分和评分的标准差。

由于每个用户评分数量不同, 本文探究在不同的评分计数和评分数值的情况下, 采取的分佈策略。不同的评分计数和不同评分数值下得到的用户质量分数和平均绝对误差 (Mean Absolute Error, MAE) 的示例如表 1-表 6 所示。

表 1 低评分计数和不同数值的示例

Table 1 Example of Low Ratings Count and Variant Ratings

评分数值	1	2	3	4	5
频率	3	1	2	4	1

表 2 低评分计数和不同数值下的案例结果

Table 2 Result of Example of Low Ratings Count and Variant Ratings

Beta 分布策略	用户质量分数	MAE
U形 $\alpha=\beta=0.5$	2.90	1.21
钟形 $\alpha=\beta=5$	3.12	1.23

表 3 低评分计数和类似数值的示例

Table 3 Example of Low Ratings Count and Similar Ratings

评分数值	1	2	3	4	5
频率	0	0	3	7	1

表 4 低评分计数和类似数值下的案例结果

Table 4 Result of Example of Low Ratings Count and Similar Ratings

Beta 分布策略	用户质量分数	MAE
U形 $\alpha=\beta=0.5$	1.89	1.46
钟形 $\alpha=\beta=5$	1.95	1.42

表 5 高评分计数的示例

Table 5 Example of High Ratings Count

评分数值	1	2	3	4	5
频率	125	75	25	75	50

表 6 高评分计数下的案例结果

Table 6 Result of High Ratings Count

Beta 分布策略	用户质量分数	MAE
U形 $\alpha=\beta=0.5$	2.86	1.02
钟形 $\alpha=\beta=5$	3.03	1.01

不难看出, 在不同评分数量和分布的情况下, 选择不同 Beta 分布策略得到的结果存在差异。在评分数量较少且评分数值分布较均匀时, 选择 U 形分布得到的 MAE 较低; 在用户评分数量较少且评分数值分布不均匀时, 选择钟形分布更好; 在评分计数较高时, 选择钟形分布得到的结果较好。总之, MAE 随着评分计数的增长和评分数值均匀分布而降低, 评分计数少、评分数值分布不均匀会导致用户的质量下降。

不同分布方法在两个不同评分频率的例子下分配权重的结果如表 7 所示, 其中 Mean 表示平均分配权重, NDR^[18] 表示使用正态分布分配权重的方法。

表 7 使用 Mean、NDR 和 RAUQ_Beta 生成的每层权重示例

Table 7 Example of Weights per Level Generated Using Mean, NDR and RAUQ_Beta

案例	评分数值	1	2	3	4	5
		评分频率	5	1	1	1
案例 1	Mean Weights	0.566	0.111	0.111	0.111	0.111
	NDR Weights	0.578	0.147	0.123	0.191	0.060
	RAUQ_Beta Weights	0.511	0.022	0.027	0.043	0.396
案例 2	评分频率	125	25	25	25	25
	Mean Weights	0.566	0.111	0.111	0.111	0.111
	NDR Weights	0.573	0.140	0.122	0.096	0.069
	RAUQ_Beta Weights	0.590	0.168	0.134	0.083	0.025

在案例 1 中, RAUQ_Beta 为极端评分分配高权重, 为数值居中的评分分配较低权重。在案例 2 中, RAUQ_Beta 生成了一个钟形分布, 为数值居中的评分分配较高的权重, 相同级别的重复评分的非线性权重增量导致该评分级别获得更高的总权重。根据结果来看, RAUQ_Beta 的优势在于它强调了与 NDR

所用频率类似的额定频率,而当每个项目的可用评分没有反映可信的分布时,RAUQ_Beta 对用户质量的评估比 NDR 算法更准确。

(3) 基于时间段的评分数值分布的项目质量研究方法

由于项目评分存在随着评分时间的变化而改变的情况,本文结合时间段的评分分布来改进上述基于评分分布的质量算法,提出一种 RAIQ_TIME (Recommendation Algorithm based on Item Quality_Time Distribution) 方法,即基于时间段的评分数值分布的项目质量算法。将项目的评分稠密时间段定义为某时间段的评分数量占比大于该项目总评分数量 10%^[19] 的时间段。从数据分析的角度来看,评分稠密时间段大多分布在靠前的时间段,其平均评分更加接近于此项目的平均分,且集中在平均划分时间后的排在前两名的时间段内。以 MovieLens-1M 数据集中编号为 1 的项目为例,项目 1 的评分数量为 1 456,平均分为 4.16。该项目评分时间按照平均和稠密划分的结果如表 8、表 9 所示。

表 8 平均划分项目 1 的评分时间的结果
Table 8 Result of Average Rating Time for Item 1

评分时间	1	2	3	4	5
评分数量	799	620	21	8	8
平均评分	4.176	4.140	4.095	4	4.375
分配权重	0.549	0.426	0.015	0.005	0.005

表 9 项目 1 评分稠密时间段划分的结果
Table 9 Result of Dense Rating Time Division for Item 1

稠密时间段评分数量	207	439	145	426	160
平均评分	4.169	4.202	4.134	4.117	4.206
分配权重	0.290	0.415	0.105	0.155	0.035

平均划分方法是按照项目评分数据集平均划分为 5 个相同长度的时间段,由于时间靠前的评分稠密时间段更加接近平均值,采用 WBC^[18] 投票的方法分配给靠前稠密评分时间段更多权重。具体地,将最后一个评分稠密时间段权重的 40% 分配给第一个稠密时间段,其 30% 权重分配给第二个稠密时间段;将倒数第二个评分稠密时间段权重的 30% 分配给第一个稠密时间段,其 20% 权重分配给第二个稠密时间段。评分稠密时间段和评分分布算法结合的过程

如下:

输入:稠密时间段评分数据

输出:使用稠密时间段评分得到的项目权重

过程:

①利用公式(10)得到项目 i 分布(使用 IRRC)的形状参数;

②利用公式(8)得到项目 i 的质量分数;

③利用公式(5)得到项目 i 的评分稠密时间分布;

④采用 WBC 算法重新分配稠密时间段权重。

从时间划分和分配权重的结果来看,使用评分稠密时间段的方法,只需使用部分而无需考虑全部的评分,这显然优于平均划分评分时间的方法。

4 实验

4.1 数据集

本文使用明尼苏达大学 GroupLens 小组发布的 MovieLens-1M 和 MovieLens-100k 数据集进行实验,MovieLens-1M 数据集包含 6 040 个用户,3 952 部电影,共计 1 000 209 条评分记录;MovieLens-100k 数据集包含 943 个用户,1 682 部电影,共计 100 000 条评分记录,每一条评分记录为数字 1~5。数据集包含每个用户的年龄、性别、职业等信息。以 MovieLens-1M 数据集为例,用户信息包含用户 ID、年龄、性别、职业等,其中用户的年龄如果小于 18 则为 1。电影信息数据包括电影 ID、电影名、电影类别等。首先对电影数据集进行预处理:

(1)Age 字段:转换为 7 个连续的数字 0~6。

(2)Genres 字段:是分类字段,转换为数字。首先将 Genres 中的类别转为字符串到数字的字典,然后再将每个电影的 Genres 字段转为数字列表。

(3)Title 字段:处理方式与 Genres 字段一样,首先创建文本到数字的字典,然后将 Title 中的描述转为数字的列表。

(4)Timestamps 字段:转换为当地时间。

4.2 实验设计

为验证本文所提模型的性能,设计对比实验,分别如下:

(1)ConvMF^[4]:传统卷积矩阵分解的推荐模型;

(2)CDL^[7]:为解决评分稀疏问题,结合文本和

评分信息,将堆叠降噪自编码器和概率矩阵分解融合在一起的深度学习模型;

(3)NDRU^[20]:基于正态分布的不确定性用户质量模型;

(4)EPIR^[12]:根据用户兴趣生成项目质量的个性化推荐方法;

(5)WS^[21]:用于计算用户初始质量的算法;

(6)Correlation^[11]:使用用户的历史评分和项目内在质量的相关性来决定用户质量;

(7)LRMF^[22]:融合用户地理位置信息来计算用户质量的方法;

(8)RAUIQ_CORR:本文提出的基于用户评分和项目质量一致性的用户质量测度模型;

(9)RAUQ_Beta:本文提出的基于评分数值分布的用户质量研究算法;

(10)RAIQ_TIME:本文提出的基于时间段的评分数值分布的项目质量算法。

4.3 实验环境及参数设置

本实验的运行环境为:Windows10操作系统、16GB内存、Intel Core i5 CPU。在PyCharm编译软件上使用TensorFlow1.8版本实现了本模型。具体地,本模型所有的训练参数均使用正态分布来初始化,正则化参数设为0.5,学习率设为0.0001。训练集和测试集按照8:2划分,评价指标使用MAE和MSE。

4.4 基于用户评分和项目质量一致性的用户质量测度方法的实验结果

迭代10次时,RAUIQ_CORR与ConvMF、NDRU、EPIR模型对比的实验结果如表10所示。可以看出,本文模型比其他基于用户质量的推荐模型都有提升,对比ConvMF、NDRU和EPIR模型分别提升平均2.35%、1.68%和1.05%,预测精度有所提高。迭代10次时,不同模型分别在项目质量上使用Average、HITS算法和RAUIQ_CORR的运行结果如表11所示。不难看出,利用用户评分和项目质量一致性反映项目质量的效果最好,对比平均分和HITS分别平均提升了3.98%和3.09%。

由于RAUIQ_CORR能反映项目或用户质量信息对评分的影响程度,且充分考虑了项目用户直接和隐含特征信息,这比仅用平均分或其他传统方法

反映项目用户的质量更加准确,本文模型比其他用户质量推荐模型的效果更好。

表10 针对用户质量的实验结果

Table 10 Results of User Quality

算法	MAE	MSE
ConvMF	0.747	0.842
NDRU	0.740	0.839
EPIR	0.735	0.834
RAUIQ_CORR	0.728	0.825

表11 针对项目质量的实验结果

Table 11 Results of Item Quality

算法	MAE	MSE
Average	0.755	0.860
HITS	0.750	0.851
ConvMF	0.745	0.842
RAUIQ_CORR	0.728	0.825

4.5 基于评分分值分布的用户质量算法的实验结果

使用评分分布的用户质量深度算法和其他用户质量算法在两个不同稀疏程度的电影评分数据集上相比较的实验结果如表12所示,评价指标为MAE。从结果来看,RAUQ_Beta算法和最新的利用用户质量进行推荐的算法相比效果更好。从评分分布的角度来看,RAUQ_Beta算法在不同项目的可用评分数没有反映可信的分布时,采用最优的分布形状来生成用户质量的权重,更加灵活。并且在不同用户和评分数量的情况下对用户的质量属性判断更加准确,更加有价值与可行性,有效性也比RAUIQ_CORR更加突出。

表12 各用户质量算法的实验结果

Table 12 Results of Each User Quality Algorithm

算法	ML-100k	ML-1M
Mean	0.805	0.776
WS	0.799	0.768
Correlation	0.781	0.755
LRMF	0.772	0.741
ConvMF	0.778	0.746
CDL	0.766	0.737
RAUIQ_CORR	0.761	0.728
RAUQ_Beta	0.757	0.725

4.6 基于时间段的评分数值分布的项目质量研究方法的实验结果

使用不同项目质量算法和 RAIQ_TIME 算法的实验结果如表 13 所示,评价指标为 MAE。从数据分析角度来看,时间对项目质量和用户兴趣的影响很大,并且 RAIQ_TIME 能够很好地利用时间因素对项目质量进行处理,还能够不使用全部评分信息对项目质量进行有效计算,节省了运算时间,提高了推荐准确率,其结果优于其他对比算法,比 RAUIQ_CORR 也更具有效性。根据 4.5 节和 4.6 节的实验结果来看,从评分分值分布和时间分布的研究角度都可以更好地了解用户和项目的质量信息对于推荐的有效性,这些方法同时也适用于其他各种类型的推荐。

表 13 各项目质量算法的实验结果

Table 13 Results of Each Item Quality Algorithm

算法	ML-100k	ML-1M
Average	0.790	0.755
HITS	0.781	0.750
ConvMF	0.778	0.746
RAUIQ_CORR	0.766	0.728
RAIQ_TIME	0.758	0.724

5 结 语

本文提出深度融合用户项目质量特征模型,从用户评分和项目质量一致性、评分数值分布和基于时间段的评分数值分布三个方面,系统地提出对用户和项目质量特征进行评估的有效手段。作为对传统深度推荐系统模型的拓展,评分预测效果更加准确,有效缓解了深度推荐模型中数据有效特征缺乏和表现力不足等问题。现有改进主要结合评分信息,没有探索用户项目等更多属性特征的质量指标评定方法,在一定程度上影响了推荐效果。下一步将针对其他数据集测试模型的性能,并且进一步完善质量指标的设计,以达到更好的推荐效果。

参考文献:

- [1] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann Machines for Collaborative Filtering[C]//Proceedings of the 24th International Conference on Machine Learning. 2007: 791-798.
- [2] 刘凯, 张立民, 周立军. 深度学习在信息推荐系统的应用综述[J]. 小型微型计算机系统, 2019, 40(4): 738-743.(Liu Kai, Zhang Limin, Zhou Lijun. Survey of Deep Learning Applied in Information Recommendation System[J]. Journal of Chinese Computer Systems, 2019, 40(4): 738-743.)
- [3] Lian J X, Zhou X H, Zhang F Z, et al. XDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 1754-1763.
- [4] Kim D, Park C, Oh J, et al. Convolutional Matrix Factorization for Document Context-Aware Recommendation[C]//Proceedings of the 10th ACM Conference on Recommender Systems. 2016: 233-240.
- [5] Filho R J R, Wehrmann J, Barros R C. Leveraging Deep Visual Features for Content-Based Movie Recommender Systems[C]//Proceedings of the 2017 International Joint Conference on Neural Networks. IEEE, 2017: 604-611.
- [6] Messina P, Dominguez V, Parra D, et al. Content-Based Artwork Recommendation: Integrating Painting Metadata with Neural and Manually-Engineered Visual Features[J]. User Modeling and User-Adapted Interaction, 2019, 29(2): 251-290.
- [7] Wang H, Wang N Y, Yeung D Y. Collaborative Deep Learning for Recommender Systems[C]//Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015: 1235-1244.
- [8] 邓存彬, 虞慧群, 范贵生. 融合动态协同过滤和深度学习的推荐算法[J]. 计算机科学, 2019, 46(8): 28-34.(Deng Cunbin, Yu Huiqun, Fan Guisheng. Integrating Dynamic Collaborative Filtering and Deep Learning for Recommendation[J]. Computer Science, 2019, 46(8): 28-34.)
- [9] 胡朝举, 郑浩. 融合上下文信息的深度推荐模型[J]. 计算机应用研究, 2021, 38(4): 1074-1078.(Hu Chaoju, Zheng Hao. Deep Recommendation Model with Context Information[J]. Application Research of Computers, 2021, 38(4): 1074-1078.)
- [10] 李涛, 符丁. 基于协同过滤算法的自动化隐式评分音乐双重推荐系统[J]. 计算机测量与控制, 2018, 26(11): 171-175.(Li Tao, Fu Ding. An Automated Implicit Scoring Music Dual Recommendation System Based on Collaborative Filtering Algorithm[J]. Computer Measurement & Control, 2018, 26(11): 171-175.)
- [11] Qian F L, Zhao S, Tang J, et al. SoRS: Social Recommendation Using Global Rating Reputation and Local Rating Similarity[J]. Physica A: Statistical Mechanics and Its Applications, 2016, 461: 61-72.
- [12] Abdel-Hafez A, Xu Y, Tian N. Item Reputation-Aware Recommender Systems[C]//Proceedings of the 16th International Conference on Information Integration and Web-Based Applications & Services. ACM, 2014: 79-86.

- [13] Kim Y. Convolutional Neural Networks for Sentence Classification[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2014: 1746-1751.
- [14] Zhou Y B, Lei T, Zhou T. A Robust Ranking Algorithm to Spamming[J]. Europhysics Letters, 2011, 94(4): 48002.
- [15] Radde S, Freitag B. Using Bayesian Networks to Infer Product Rankings from User Needs[C]//Proceedings of the UMAP 2010 Workshop on Intelligent Techniques for Web Personalization and Recommender Systems. 2010.
- [16] Kleinberg J M. Authoritative Sources in a Hyperlinked Environment[J]. Journal of the ACM, 1999, 46(5): 604-632.
- [17] 王岩, 张杰, 许合利. 结合用户兴趣和改进的协同过滤推荐算法[J]. 小型微型计算机系统, 2020, 41(8): 1665-1669. (Wang Yan, Zhang Jie, Xu Heli. Combining User Interests with Improved Collaborative Filtering Recommendation Algorithm[J]. Journal of Chinese Computer Systems, 2020, 41(8): 1665-1669.)
- [18] Abdel-Hafez A J. Reputation Model Based on Rating Data and Application in Recommender Systems[D]. Brisbane: Queensland University of Technology, 2016.
- [19] 于鹏华. 数据数量与质量敏感的推荐系统若干问题研究[D]. 杭州: 浙江大学, 2016. (Yu Penghua. Research on Data Quantity and Quality Sensitive Recommender Systems[D]. Hangzhou: Zhejiang University, 2016.)
- [20] Abdel-Hafez A, Xu Y, Josang A. A Rating Aggregation Method for Generating Product Reputations[C]//Proceedings of the 25th ACM Conference on Hypertext and Social Media. 2014: 291-293.
- [21] McNally K, O' Mahony M P, Smyth B, et al. Towards a Reputation-Based Model of Social Web Search[C]// Proceedings of the 15th International Conference on Intelligent User Interfaces. 2010: 179-188.
- [22] Li S, Wen J H, Luo F J, et al. A Location and Reputation Aware Matrix Factorization Approach for Personalized Quality of Service Prediction[C]//Proceedings of the 2017 IEEE International Conference on Web Services. IEEE, 2017: 652-659.

作者贡献声明:

郑潇: 进行实验, 采集、清洗和分析数据, 起草论文;
李树青: 整体实验方案设计, 论文最终版本修订;
张志旺: 算法验证, 论文修订。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据[1-2]为公开数据集; 支撑数据[3]由作者自存储, E-mail: 1395787037@qq.com。
[1] ml-latest-small.ratings.csv. MovieLens 实验数据集 1. <https://files.grouplens.org/datasets/movielens/ml-latest-small.zip>.
[2] ml-100k.ratings.csv. MovieLens 实验数据集 2. <https://files.grouplens.org/datasets/movielens/ml-100k.zip>.
[3] 郑潇. 中间数据_迭代次数.xlsx. ML-100k 和 ML-1M 数据集实验结果。

收稿日期: 2021-07-07
收修改稿日期: 2021-11-12

Measuring User Item Quality with Rating Analysis for Deep Recommendation Model

Zheng Xiao Li Shuqing Zhang Zhiwang

(College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China)

Abstract: [Objective] This paper designs a new deep learning algorithm to improve the recommendation results. [Methods] Our model evaluated user and item quality features from user ratings and item quality consistency, numerical distribution of ratings and time-period-based numerical distribution of ratings. [Results] We examined our model with the MovieLens dataset, and found the MAE and MSE were improved by up to 3.71% and 4.24%, respectively. [Limitations] More research is needed to explore a quality index evaluation method including attribute features of user and items. [Conclusions] The proposed model generates more accurate scoring prediction, and effectively improves the quality of recommendation.

Keywords: Recommendation System Deep Learning User and Item Quality Effective Features