

DOI: 10.7652/xjtuxb201909011

QPR-NN:一种结合二次多项式回归与神经网络的推荐算法

廖彬^{1,2}, 张陶^{3,4}, 于炯³, 国冰磊³, 李敏², 刘炎⁵

(1. 新疆财经大学丝路经济与管理研究院, 830012, 乌鲁木齐; 2. 新疆财经大学统计与数据科学学院, 830012, 乌鲁木齐; 3. 新疆大学信息科学与工程学院, 830012, 乌鲁木齐; 4. 新疆医科大学医学工程技术学院, 830012, 乌鲁木齐; 5. 清华大学软件学院, 100084, 北京)

摘要: 针对传统推荐算法不能很好地适应数据高规模及高稀疏性的问题,结合深度学习数据建模的方法,提出了一种结合二次多项式回归与神经网络(QPR-NN)的推荐算法。在对已有特征提取方法缺陷分析的基础上,利用二次多项式回归模型将用户对物品的评分数据进行特征提取及降维,充分挖掘了用户与物品之间的相关性。将特征提取后的数据作为深度学习训练模型的输入,增加输入数据与训练模型之间的匹配度,并将训练得到的模型用于推荐评分预测。在 MovieLens 与 Epinions 两组数据集上的实验结果表明:QPR 特征提取方法与 QPR-NN 推荐算法在平分绝对误差与均方根误差评价指标上均优于现有的主流算法,QPR-NN 推荐算法可以有效提升推荐准确率。

关键词: 推荐算法;深度学习;特征提取;二次多项式回归

中图分类号: TP391 **文献标志码:** A **文章编号:** 0253-987X(2019)09-0079-09

QPR-NN: A New Recommendation Algorithm Combining Quadric Polynomial Regression and Neural Network

LIAO Bin^{1,2}, ZHANG Tao^{3,4}, YU Jiong³, GUO Binglei³, LI Min², LIU Yan⁵

(1. Institute of Silk Road Economy and Management, Xinjiang University of Finance and Economics, Urumqi 830012, China;
2. College of Statistics and Data Science, Xinjiang University of Finance and Economics, Urumqi 830012, China;
3. School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China;
4. Department of Medical Engineering and Technology, Xinjiang Medical University, Urumqi 830011, China;
5. School of Software, Tsinghua University, Beijing 100084, China)

Abstract: The traditional recommendation algorithms cannot adapt to the problem of high-scale and sparse data. A recommendation algorithm combining quadric polynomial regression and neural network (QPR-NN) is proposed in view of the generality of deep learning data modeling, where quadratic regression is combined with neural network. Following the analysis on the defects in the existing feature extraction methods, the algorithm chooses quadratic regression model to extract feature and reduces the dimensions for the user rating data to fully explore the correlation between the user and item data. The data after feature extraction are taken as the input of deep learning training model to increase the matching degree between the input data and the training model, then this model is used for recommending score prediction. For the two datasets of MovieLens and Epinions, the experimental results show that the QPR feature

收稿日期: 2019-02-23。 作者简介: 廖彬(1986—),男,副教授,硕士生导师。 基金项目: 国家自然科学基金资助项目(61562078,61462079);新疆维吾尔自治区自然科学基金资助项目(2016D01B014)。

网络出版时间: 2019-06-10

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1069.T.20190606.1710.012.html>

extraction method and the QPR-NN recommendation algorithm are superior to the existing mainstream algorithms in the evaluation indexes mean absolute error and root mean square error, and effectively improve the recommendation accuracy.

Keywords: recommendation algorithm; deep learning; feature extraction; quadric polynomial regression

据文献[1]统计,2017年全球数据总量为21.6 ZB,按照每年40%左右的增长率计算,预计到2020年全球的数据总量将达到40 ZB,如何更好地管理并利用这些海量数据,是大数据研究的核心内容。特别是在互联网领域,面对规模庞大的数据,如何进行精准地选择成为困扰用户的一大难题。推荐系统为了解决这个问题,一方面从用户的角度出发,高效地从海量数据中选择出用户可能感兴趣的物品(如:电影、书籍、音乐、商品、新闻、微博等);另一方面从商家的角度出发,向用户提供更为个性化的服务,提高用户的黏性及信任度,从而提高经济收益。

根据数据源的不同,当前主流的推荐系统所采用的核心算法主要分为基于人口统计学的推荐^[2]、基于内容的推荐^[3]以及基于协同过滤的推荐^[4-5]这3种。协同过滤算法在早期取得了不错的应用效果,但是随着数据量以指数级增加,数据稀疏性、冷启动以及算法的可扩展性等问题不断出现。因此,研究能同时适应数据高规模及高稀疏性特点的推荐算法成为了当前的研究热点。

近年来,深度学习研究得益于海量数据的积累、硬件计算能力的提升以及算法研究者的努力而不断发展,特别是深度神经网络^[6]及其变种形式的卷积神经网络^[7]、循环神经网络^[8]等,在机器视觉、自然语言处理等领域大放异彩。深度学习的本质是对数据特征进行深层次地抽象挖掘,通过大规模数据来学习有效的特征表示以及复杂的映射机制,从而建立起数据模型。深度学习作为一种适应度较高的数据建模方法,理论上可以广泛地应用到不同的领域,其中就包括推荐系统。但是,目前深度学习应用到推荐系统的工作相对较少。

为了解决现有推荐算法不能很好适应数据高规模及高稀疏性的现状,本文结合深度学习建模方法通用性的特点,提出一种结合二次多项式回归与神经网络(QPR-NN)的推荐算法。QPR-NN首先利用QPR模型来对用户物品的评分数据进行特征提取及降维,经过QPR模型处理后的数据作为深度学习训练模型的输入,训练后的模型用于预测用户对物品的评分。与主流推荐算法进行的平分绝对

误差 E_{MA} 以及均方根误差 E_{RMS} 这2个评价指标的对比实验表明,QPR-NN算法能够有效地提高推荐质量。

1 相关工作

由于本文的关注点是利用二次多项式回归及深度学习算法来解决现有推荐算法不能很好地适应数据高规模及高稀疏性的问题,所以本节首先对该问题的已有相关研究进行阐述,其次对深度学习在推荐系统上的应用研究进行介绍。

1.1 解决数据高规模及高稀疏性问题的研究

随着分布式计算模型 MapReduce 的普及,推荐算法逐渐从单机向分布式模式发展,进而很好地解决了数据的高规模问题。Guo 等和 Zhao 等将协同过滤算法移植到 Hadoop 平台,提高了算法的并行计算能力^[9-10]。Schelter 等面对用户数据量快速增长的问题,提出了基于 MapReduce 的近邻协同过滤算法,并通过 Yahoo 的 7 亿条音乐数据证明了算法在效率上的明显提升^[11]。廖彬等对 MapReduce 架构下的协同过滤算法存在的性能问题进行了分析,并提出利用 Spark 内存计算^[12]、相似度计算效率优化^[13]、Spark DAG 调度^[14]等方法来进一步提高分布式推荐算法的执行效率。

针对高稀疏性问题,一般采用数据填充、聚类、矩阵分解等方法解决。数据填充是在进行相似性计算之前,对用户及评分矩阵数据进行填充,从而降低其稀疏性。聚类是一种有效的数据降维方法,能够在有效地降低算法计算量的同时提高推荐质量。矩阵分解中最具有代表性的工作是奇异值分解(SVD)^[15-16],文献[17]提出了带正则化的基于迭代最小二乘法的矩阵分解方法,文献[18]通过矩阵分解实现对原始数据的降维及数据填充,并引入了时间衰减函数预处理用户评分,提升了推荐算法的准确度。

1.2 深度学习在推荐算法上的应用研究

Salakhutdinov 等最早在 NIPS 2007 上提出了使用限制玻尔兹曼机进行评分预测,该模型是2层的类二部图结构,通过非线性关系关联隐含层与可见层的数据信息,但该模型最大的问题是链接隐含

层与评分层的权重参数(矩阵)规模太大^[19]。在后期工作中,部分工作利用深度神经网络模型来作为信息变换模型,例如:文献[20]使用多层降噪自动编码器将文本特征与评分预测特征相融合;文献[21]利用卷积神经网络解决音乐推荐系统中的冷启动问题。除此之外:文献[22]将卷积神经网络与协同过滤算法进行结合,提高了音乐推荐的质量;微软公司提出了深度结构化语义模型(DSSM),该模型是基于多层神经网络模型搭建的广义语义匹配模型,通过级联的深度神经网络模型的映射与变换,最终在同一个隐含空间内表示了推荐系统中的用户和物品,并可以使用余弦相似度进行计算^[23];文献[24]使用循环神经网络进行基于 session 的推荐,充分考虑了物品的时序关系;Brebisson 等使用神经网络模型解决 ECML/PKDD 会议的数据挑战题目“出租车下一地点预测”,取得了比赛第一名的成绩,并且对多种多层感知器模型以及循环神经网络模型进行了对比,发现基于改进后的多层感知器模型会取得最好的效果^[25]。

已有工作大部分是利用深度学习算法作为训练模型去处理推荐系统的数据,而在输入数据的特征提取方面,往往采用传统的几种处理方法(如 RaF^[26]、SVD^[15-16]及 PMF^[27-28]等),并没有更多地考虑到用户和物品之间的适配度。本文与已有工作的最大不同之处在于:在分析已有特征提取方法缺陷的基础上,为了更好地挖掘用户与物品之间的相关

性,提出 QPR 模型对数据进行特征提取及降维,然后将处理后的数据作为深度学习训练模型的输入,最后将训练后得到的模型用于推荐系统的评分预测,提高了算法的推荐质量。

2 基于 QPR 模型的特征表示模型

2.1 已有特征表示模型的优缺点分析

若用 m 表示用户数量、 n 表示物品数量,协同过滤推荐算法的输入可以表示为 $m \times n$ 的用户项目评分矩阵 $\mathbf{R}$,具体如图 1 所示,其中 R_{ij} 表示第 i 个用户对第 j 个物品的评分, $R_{ij} \in \mathbf{R}^{m \times n}, i \in [1, m], j \in [1, n]$ 。 R_{ij} 的取值范围可以根据实际应用系统的特点进行设定,连续或离散的情况都有,大多数应用系统采用数值为 1~5 的离散值来表示评分,而 $R_{ij} = 0$ 则表示用户尚未对该项目进行评分。

	物品 1	物品 2	...	物品 n
用户 1	R_{11}	R_{12}	...	R_{1n}
用户 2	R_{21}	R_{22}	...	R_{2n}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
用户 m	R_{m1}	R_{m2}	...	R_{mn}

图 1 用户项目评分矩阵 $\mathbf{R}$

本文需要使用用户项目评分矩阵 $\mathbf{R}$ 获得用户及项目的特征表示,目前主流的特征表示方法有 RaF、SVD 及 PMF,这 3 种方法的原理及优缺点分析如表 1 所示。

表 1 几种特征表示方法的对比分析

方法	原理	优点	缺点
RaF	将用户对全部物品的打分直接作为用户的特征表示。	简单,效率较高。	当评分矩阵高度稀疏时,容易造成特征失真,导致推荐结果不准确。
SVD	利用 SVD 将评分矩阵分解为 3 个矩阵的乘积,分别获得用户及物品的特征。	将高维的评分矩阵映射为 2 个低维的用户和物品矩阵,较好地解决了数据的稀疏性问题。	需要对缺失值进行预处理,并且高复杂度的 SVD 计算容易导致算法性能不佳。
PMF	使用用户的特征矩阵和物品的特征矩阵的乘积去拟合评分矩阵。	有效地避免了对缺失值的处理。	不能有效挖掘特征之间的关联关系。

2.2 利用 QPR 模型提取用户和物品的特征

2.1 节对已有的特征分析方法进行了分析,为了解决这些特征分析方法在特征提取时存在的问题,本文利用 QPR 模型提取得到用户和物品的特征。在 QPR 模型中,设特征向量为 $\mathbf{x} = (x_1, \dots, x_w)$,其对应的监督值 z 的拟合值 $\hat{z}$ 为

$$\hat{z} = h + \sum_{i=1}^w k_i x_i + \sum_{i=1}^{w-1} \sum_{j=i+1}^w K_{ij} x_i x_j \tag{1}$$

式中: h 是常数项系数; k 与 K 分别表示一次与二次

项的系数; w 表示特征向量 $\mathbf{x}$ 的维度。对于用户项目评分矩阵 $\mathbf{R}$,设待解的用户特征矩阵为 $\mathbf{U}$,物品特征矩阵为 $\mathbf{T}$ 。设 a 为用户特征维度, b 为物品特征维度;对于 $\mathbf{U} \in \mathbf{R}^{m \times a}$,第 u 行的 a 维行向量 $\mathbf{U}_u$ 表示第 u 个用户的特征向量, U_{ui} 表示 $\mathbf{U}_u$ 中的第 i 个元素;对于 $\mathbf{T} \in (\mathbf{R}^{n \times b})^T$,第 t 行的 b 维行向量 $\mathbf{T}_t$ 表示第 t 个物品的特征向量, T_{tj} 表示 $\mathbf{T}_t$ 中的第 j 个元素。用 R_{ut} 和 $\hat{R}_{ut}$ 分别表示评分矩阵 $\mathbf{R}$ 中第 u 行第 t 列元素的实际值和估计值,将 $\mathbf{x} = (\mathbf{U}_u, \mathbf{T}_t)$ 以及 $z = R_{ut}$ 代入

式(1),得到

$$\begin{aligned} \hat{R}_{ut} = & h + \sum_{i=1}^a k_i U_{ui} + \sum_{j=1}^b k_{j+a} T_{tj} + \\ & \sum_{i=1}^{a-1} \sum_{j=i+1}^a K_{ij} U_{ui} U_{uj} + \sum_{i=1}^{b-1} \sum_{j=i+1}^b K_{i+a, j+a} T_{ti} T_{tj} + \\ & \sum_{i=1}^a \sum_{j=1}^b K_{i, j+a} U_{ui} T_{tj} \end{aligned} \quad (2)$$

为了简化式(2),设

$$e_u = \sum_{i=1}^a k_i U_{ui} + \sum_{i=1}^{a-1} \sum_{j=i+1}^a K_{ij} U_{ui} U_{uj} \quad (3)$$

$$f_t = \sum_{j=1}^b k_{j+a} T_{tj} + \sum_{i=1}^{b-1} \sum_{j=i+1}^b K_{i+a, j+a} T_{ti} T_{tj} \quad (4)$$

将式(3)与式(4)代入式(2),可以得到

$$\hat{R}_{ut} = h + e_u + f_t + \sum_{i=1}^a \sum_{j=1}^b K_{ij} U_{ui} T_{tj} \quad (5)$$

可以看出,式(5)中的 e_u 与 f_t 分别与其对应的用户 u 与物品 t 相关。使用最小二乘损失函数,可以得到模型的拟合误差 Q 为

$$Q = \frac{1}{2} \sum_{u,t} (R_{ut} - \hat{R}_{ut})^2 \quad (6)$$

展开式(6),得到

$$Q = \frac{1}{2} \sum_{u,t} (R_{ut} - h - e_u - f_t - \sum_{i=1}^a \sum_{j=1}^b K_{ij} U_{ui} T_{tj})^2 \quad (7)$$

当式(7)达到最优最小值解时,即计算出了所需的用户特征矩阵 U 与物品特征矩阵 T 。

3 QPR-NN 算法

3.1 QPR 与 NN 模型的连接

在 2.2 节得到用户特征矩阵 U 与物品特征矩阵 T 的基础上,本节探讨怎样利用神经网络模型,对这些输入数据进行训练,并利用训练后的模型拟合评分数据,从而实现推荐功能。

(1)神经网络输入层。设输入向量为 v_0 , v_0 由用户特征 U_i 与物品特征 T_j 通过链接函数 f_c (神经网络中采用 concatenate 函数)链接而成,公式为

$$v_0 = f_c(U_i, T_j) \quad (8)$$

(2)隐藏层。输入向量 v_0 经过第 1 层隐藏层后,需要引入一些非线性特性,此时需要使用激活函数 f_c (神经网络中采用 activation 函数)进行激活,公式为

$$v_1 = f_a(w_1 v_0 + b_1) \quad (9)$$

式中: v_1 为输出向量; w_1 为输入层与隐藏层之间的权值矩阵; b_1 为偏移向量。本文采用线性整流(ReLU)函数作为激活函数,这是因为相比 sigmoid 函

数,ReLU 函数能够在节省计算量的同时减少参数之间的相互依赖关系,缓解过拟合问题的发生。

基于式(9),可以计算出在第 i 层隐藏层的输出 v_i 为

$$v_i = f_a(w_i v_{i-1} + b_i) \quad (10)$$

(3)输出层。输出层的目标是拟合评分数据 R_{ij} ,采用独热编码函数 f_e (神经网络中采用 One HotEncode 函数)将连续值转化为离散值,公式为

$$y = f_e(R_{ij}) \quad (11)$$

(4)softmax 变换。在输出层之前,对输出数据进行 softmax 变换可以获得神经网络模型的输出结果 $\hat{y}$,公式为

$$\hat{y} = f_s(w_o x_l + b_o) \quad (12)$$

式中: f_s 为 softmax 变换函数,神经网络中采用 softmax 函数; w_o 表示输出层的权值矩阵; x_l 表示最后一层隐藏层的输出,其中变量 l 表示隐藏层的层数; b_o 表示输出层的偏移量。

当通过式(12)计算出 $\hat{y}$ 后,可以利用香农信息论中的交叉熵来评估与输出值 $\hat{y}$ 与监督值 y 的差异,设差异值为 C ,可以得出

$$C = -\frac{1}{n} \sum_{i=1}^c [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)] \quad (13)$$

式中: y_i 和 $\hat{y}_i$ 表示神经网络输出层编号为 i 的神经元的监督值和输出值; c 表示输出层的神经元个数; n 表示样本参数,当 $n=1$ 时,式(13)可简化为

$$C = -\sum_{i=1}^c [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)] \quad (14)$$

最终得到模型预测用户 i 对物品 j 的评分值 $\hat{R}_{ij}$ 为

$$\hat{R}_{ij} = f_g(\hat{y}_c) \quad (15)$$

式中 f_g 函数表示从神经网络中输出的 c 个 $\hat{y}(\hat{y}_c)$ 中取出最大值,神经网络中采用 getmax 函数。

3.2 计算用户与物品特征矩阵

神经网络的训练主要在于权值矩阵和偏移向量的学习,即对 3.1 节中 w_i (第 i 层神经网络的连接权值)与 b_i (第 i 层神经网络的偏移向量)进行学习。在模型训练前,需要计算用户物特征矩阵 U 与物品特征矩阵 T ,利用梯度下降法来计算各特征,算法步骤如下。

- ① $K = \text{random}()$ // 随机初始化二次项系数 K
- ② $U = \text{randomMatrix}()$ // 随机初始化矩阵 U
- ③ $T = \text{randomMatrix}()$ // 随机初始化矩阵 T
- ④ while $\Delta_{ut} = \hat{R}_{ut} - R_{ut} > \varphi$ // 收敛退出条件

```
⑤  $\Delta_{ui} \leftarrow \hat{R}_{ui} - R_{ui}$  // 计算  $\Delta_{ui}$ 
⑥  $h \leftarrow h - \theta \sum_{u,t} \Delta_{ui}$  // 更新常数项参数  $h$ 
⑦ for  $u=1$  to  $m$  do
⑧  $e_u \leftarrow e_u - \theta \sum_t \Delta_{ui}$  // 更新  $e_u$ 
⑨  $U_{ui} \leftarrow U_{ui} - \theta \sum_t (\Delta_{ui} \sum_{j=1}^b K_{ij} T_{ij})$  // 更新  $U_{ui}$ 
⑩ end for
⑪ for  $t=1$  to  $n$  do
⑫  $f_t \leftarrow f_t - \theta \sum_u \Delta_{ui}$  // 更新  $f_t$ 
⑬  $T_{ij} \leftarrow T_{ij} - \theta \sum_u (\Delta_{ui} \sum_{i=1}^a K_{ij} U_{ui})$  // 更新  $T_{ij}$ 
⑭ end for
⑮  $K_{ij} \leftarrow K_{ij} - \theta \sum_{u,t} \Delta_{ui} U_{ui} T_{ij}$  // 更新  $K$ 
⑯ end while
⑰ return  $\mathbf{U}$  and  $\mathbf{T}$ 
```

算法的输入参数为:用户项目评分矩阵 $\mathbf{R}$, 用户数量 m , 物品数量 n , 用户特征维度 a , 物品特征维度 b , 收敛阈值 φ , 学习率 θ 。输出参数是用户特征矩阵 $\mathbf{U}$ 与物品特征矩阵 $\mathbf{T}$ 。第①~③行分别对二次项系数值 K 、用户特征矩阵 $\mathbf{U}$ 及物品特征矩阵 $\mathbf{T}$ 进行初始化;第④行判断是否达到收敛条件;第⑥行对常数项参数 h 进行更新;第⑦~⑩行循环 m 次,首先对 e_u 进行更新,第⑨行更新 U_{ui} ;第⑪~⑭行循环 n 次,第⑫行对 f_t 进行更新,第⑬行更新 T_{ij} ;第⑮行对二次项系数进行更新;第⑰行输出用户特征矩阵 $\mathbf{U}$ 与物品特征矩阵 $\mathbf{T}$ 。

算法中需要设置一个较为合适的 θ , θ 太小会导致收敛的速度很慢, θ 太大则会妨碍收敛, 导致损失函数在最小值附近波动甚至偏离最小值。优化学习率 θ 最根本的思想为:在方向导数较大的地方设置较小的 θ ;在方向导数较小的地方设置较大的 θ 。在实践中,学习率 θ 的优化通常有二分线性搜索、回溯线性搜索和 Armijo 准则共 3 种, 本文采用最简单的二分线性搜索,通过设置参数区间 $[\theta_1, \theta_2]$, 不断地将区间切分成两半,参数选择方法如下。

设梯度下降优化函数 h 为

$$h(\theta) = h(x + \theta d), \quad \theta > 0 \tag{16}$$

式中: x 表示当前点输入值; d 表示搜索斜率。

根据式(16)可知,在 $h'(\theta_1) < 0, h'(\theta_2) > 0$ 时:如果 $h'(\frac{\theta_1 + \theta_2}{2}) < 0$, 则 $\theta_1 = \frac{\theta_1 + \theta_2}{2}$; 如果 $h'(\frac{\theta_1 + \theta_2}{2}) > 0$, 则 $\theta_2 = \frac{\theta_1 + \theta_2}{2}$ 。不断重复进行二分线性搜索,

最后找到 $h'(\theta_1) = 0$ 或当搜索区间非常小时,即找到最优的 θ 值。

通过以上算法完成对特征矩阵的计算和神经网络的权值训练后,可以利用训练后的模型去预测用户对未评分项目的评分。对预测结果进行排序后,利用 topN 算法得出预测得分最高的前 N 个物品进行推荐。

4 实验及结果分析

4.1 实验环境及数据集配置

实验的操作系统版本是 Ubuntu 16.04, CPU 版本是 Intel Core i7 6700K, 内存 16 GB, SSD 大小为 250 GB, 显卡是 NVIDIA GeForce GTX 1080, CUDA 版本是 8.0 for Ubuntu 16.04, 深度学习框架是 Caffe(convolutional architecture for fast feature embedding), 具体版本是 Caffe for Ubuntu 16.04。实验的测试数据集采用 2 组数据,具体情况如表 2 所示。

表 2 实验数据集说明

数据集	类型	内容概述	数据量
MovieLens	用户电影评分数据集	包含 movies.dat、ratings.dat 及 user.dat 共 3 个文件。	含 6 040 个用户对约 3 900 部电影的 1 000 209 条评分数据。
Epinions	产品评分数据集	包括用户信任关系、用户对 item 的打分信息、评论信息等。	含 49 290 个用户对 139 738 个物品的 664 824 条评分数据。

4.2 评价指标及参数配置

平分绝对误差 E_{MA} 与均方根误差 E_{RMS} 适用于推荐系统中的评分预测场景, 本文采用 E_{MA} 与 E_{RMS} 来进行算法准确度的评价。 E_{MA} 评价指标表达式为

$$E_{MA} = \frac{\sum_{(i,j) \in \Omega^{\text{test}}} |R_{ij} - \hat{R}_{ij}|}{|\Omega^{\text{test}}|} \tag{17}$$

式中: Ω^{test} 表示测试数据集; $|\Omega^{\text{test}}|$ 表示测试数据集的数据量。

E_{RMS} 评价指标表达式为

$$E_{RMS} = \sqrt{\frac{\sum_{(i,j) \in \Omega^{\text{test}}} (R_{ij} - \hat{R}_{ij})^2}{|\Omega^{\text{test}}|}} \tag{18}$$

E_{MA} 与 E_{RMS} 越小, 表明算法预测结果与用户真实评分结果越相近, 算法预测准确率越高。

在参数设置方面: MovieLens 数据集设置参数

$a=18, b=20, \theta=0.01$;Epinions 数据集设置参数 $a=22, b=20, \theta=0.01$;在神经网络的输入层,神经元的个数为用户维度与物品维度之和,输出层的神经元个数为 5,分别对应 1~5 的评分概率。

4.3 特征分析方法对比

本节将本文提出的利用 QPR 的特征提取方法与 RaF、SVD 及 PMF 这 3 种方法进行对比。实验过程中,将 MovieLens 及 Epinions 数据集随机分成 80%及 20%两份,其中 80%部分是训练集,20%部分是验证数据集。2 组数据集下不同特征提取方法的 E_{MA} 对比结果如表 3 所示。

从表 3 可以看出:在 MovieLens 数据集下,QPR 效果最好,PMF 与 SVD 效果次之,RaF 效果最差,QPR 的 E_{MA} 比 PMF、SVD 及 RaF 分别降低了 1.2%、2.81%及 6.54%;在 Epinions 数据集下,同样是 QPR 效果最好,PMF 与 SVD 效果次之,RaF 效果最差,QPR 的 E_{MA} 比 PMF、SVD 及 RaF 分别降低了 1.3%、5.46%及 6.27%。

分析表 3 可知:RaF 在 2 组数据集上的效果最不理想,这是因为 RaF 直接将用户对全部物品的打

表 3 2 组数据集下不同特征提取方法的 E_{MA}

数据集	E_{MA}			
	RaF	SVD	PMF	QPR
MovieLens	0.703	0.676	0.665	0.657
Epinions	0.813	0.806	0.772	0.762

分直接作为用户的特征表示,当评分矩阵高度稀疏时,造成了较为严重的特征值失真现象;SVD 由于需要使用平均值填充缺失值(缺失值的预处理),使得计算出的特征值与实际值存在一定的误差;PMF 与 QPR 都取得了不错的效果,二者的共同点是都规避了缺失值的影响,但是 QPR 相比 PMF 效果更好,这是因为 QPR 能够通过二次项计算更好地将用户与物品之间的相关性体现出来。

4.4 预测准确度对比

将本文 QPR-NN 算法与较新的主流推荐算法进行对比,评价指标使用 E_{MA} 与 E_{RMS} ,具体地, E_{MA} 与 E_{RMS} 是在 MovieLens 与 Epinions 这 2 组数据集下分别执行算法 5 次后取的平均值。与本文算法进行对比的 7 个算法如表 4 所示,8 种算法在 2 组数据

表 4 QPR-NN 对比算法的信息

算法	原理描述
基于 SVD 的推荐算法 ^[15-16]	利用 SVD 进行矩阵分解,将评分矩阵分解为多个矩阵的乘积,最终利用分解出的矩阵进行乘积,完成对未知值(推荐值)的计算。
PMF 概率矩阵分解推荐算法 ^[27-28]	在矩阵分解的基础上引入概率模型,通过评分矩阵已知值使用最大后验概率和最大似然估计得到用户和物品的特征矩阵,然后用特征矩阵去预测评分矩阵中的未知值。
Item-Based 推荐算法 ^[12,29-30]	在对物品进行相似性计算的基础上,根据用户的历史记录推荐与用户喜欢的物品相似度较高的物品。
贝叶斯概率矩阵分解(BPMF)推荐算法 ^[31-33]	从贝叶斯概率角度而非传统概率角度出发,把参数作为服从某种分布的随机变量,将推荐问题转化为对该分布参数的估算。
DsRec 推荐算法 ^[34]]	在传统推荐方法的基础上,提出领域敏感的推荐算法,从而提高推荐算法的精度。
基于聚类(Cluster)的推荐算法 ^[35-37]	利用聚类方法降低用户及物品数据的维度,从而大大提高推荐系统的运行效率。
最大间距矩阵分解(MMMF)推荐算法 ^[38-39]	利用近似支持向量机代替传统支持向量机,以此来提高推荐算法的精度。

集上的 E_A 和 E_{RMS} 如表 5 所示。

对比 8 种算法在 MovieLens 数据集上的 E_{MA} 可以发现:SVD^[16]、PMF^[27]、ItemBased^[29]与 Cluster^[35]这 4 种算法比 BPMF^[32]、DsRec^[34]、PMMM^[38]及 QPR-NN 的指标差;QPR-NN 取得了最好的结果,相比 SVD、PMF、ItemBased、BPMF、DsRec、Cluster、PMMM^[38], E_{MA} 指标分别降低了 4.34%、3.36%、4.66%、1.54%、1.77%、3.62%及 1.36%。对比 8

种算法在 Epinions 数据集上的 E_{MA} 可以发现:QPR-NN 相比其他 7 种算法在 E_{MA} 指标值上取得了最优成绩;相比 SVD、PMF、ItemBased、BPMF、DsRec、Cluster、PMMM^[38]分别降低了 7.09%、6.79%、5.4%、5.62%、1.98%、4.54%及 3.29%。总体上对比 E_{MA} 可以发现,QPR-NN 在 Epinions 数据集上的性能提高比 MovieLens 多,可能原因是 Epinions 数据量相比 MovieLens 多,算法能够学习到的特征比较准确,

表 5 8 种算法在 2 组数据集上的 E_{MA} 与 E_{RMS}

算法	E_{MA}		E_{RMS}	
	MovieLens	Epinions	MovieLens	Epinions
SVD ^[16]	0.697 5	0.857 5	0.953 5	1.346 2
PMF ^[27]	0.690 4	0.854 7	0.952 0	1.301 8
ItemBased ^[29]	0.699 8	0.842 2	0.964 5	1.433 1
BPMF ^[32]	0.677 6	0.844 1	0.935 9	1.265 7
DsRec ^[34]	0.679 2	0.812 8	0.942 2	1.253 7
Cluster ^[35]	0.692 3	0.834 6	0.944 3	1.247 9
PMMMF ^[38]	0.676 4	0.823 8	0.951 7	1.257 2
QPR-NN	0.667 2	0.796 7	0.938 9	1.243 8

从而预测的精度较高。

对比 8 种算法在 MovieLens 数据集上的 E_{RMS} 可以发现: E_{RMS} 指标性能最好的是 BPMF,次之是 QPR-NN,SVD、ItemBased 及 Cluster 效果较差;QPR-NN 相比 SVD、PMF、ItemBased、BPMF、DsRec、Cluster、PMMMF 分别降低了 1.53%、1.38%、2.65%、-0.32%、0.35%、0.57% 及 1.35%。对比 8 种算法在 Epinions 数据集上的 E_{RMS} 可以发现:QPR-NN 实验结果最好,Itembased 效果最差;QPR-NN 相比 SVD、PMF、ItemBased、BPMF、DsRec、Cluster、PMMMF 分别降低了 7.61%、4.46%、13.21%、1.73%、0.79%、0.33% 及 1.07%。总体上对比 E_{RMS} 可以发现,在 MovieLens 数据集下,相同指标体系下的 BPMF 算法指标比 QPR-NN 好,但在 Epinions 中,QPR-NN 比 BPMF 好,说明随着数据量的增大,QPR-NN 算法性能更为优异。特别地,在数据集与算法相同的条件下, E_{RMS} 都要大于 E_{MA} ,反映出 E_{RMS} 对算法的评价比 E_{MA} 严格一些,这是因为 E_{RMS} 对预测误差较大的情况加大了惩罚因子的值。

4.5 隐藏层参数优化

对于用户特征维度参数 a 、物品特征维度 b 、数据量参数 m 与 n ,已知的是模型准确率会随着特征维度与数据量参数值的增大而提高。但是,神经网络中隐藏层的数量 l 将对算法性能产生巨大影响:过小的 l 将影响算法达不到最优性能,过大的 l 将影响算法的计算效率。表 6 为参数 l 对 E_{MA} 的影响。

表 6 参数 l 对 E_{MA} 的影响

数据集	E_{MA}				
	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$
MovieLens	0.676 9	0.646 4	0.642 8	0.651 5	0.654 2
Epinions	0.821 5	0.777 2	0.770 4	0.791 6	0.802 2

从表 6 可以发现:当 $l=3$ 时,算法的 E_{MA} 指标达到最优状态;当 l 逐渐增大时, E_{MA} 也逐渐增大,计算效率也会随之下降。

5 结 论

大数据时代数据的爆炸式增长使得传统的推荐算法已不能适应数据高规模及高稀疏性的特点,已有工作从移植现有推荐算法到分布式平台,或与深度学习算法进行结合这 2 种思路去解决这个问题。在与深度学习算法进行结合的研究中,大部分工作是利用深度学习算法作为训练模型去处理推荐系统的数据,而在输入数据的特征提取方面,往往采用传统的处理方法(如 RaF、SVD 及 PMF 等),并没有更多考虑到两者之间的适配度问题。所以,本文利用深度学习数据建模方法通用性的特点,提出一种结合二次多项式回归与神经网络的推荐算法 QPR-NN。首先,在对 RaF、SVD 及 PMF 等特征提取方法缺陷分析的基础上,为充分挖掘用户与物品之间的相关性,提出利用二次多项式回归模型对用户与物品的评分数据进行特征提取及降维;其次,将特征提取后的数据作为深度学习训练模型的输入,增加输入数据与训练模型之间的匹配度,并将训练得到的模型用于推荐评分预测;最后,通过与 SVD、PMF、ItemBased、BPMF、DsRec、Cluster 及 PMMMF 这 7 种主流推荐算法对比实验,证明了 QPR-NN 算法的有效性。下一步工作将主要集中在将本文推荐模型应用到更为广泛的推荐场景,从而提高模型的应用适应度。

参考文献:

[1] GANTZ J, CHUTE C, MANFREDIZ A, et al. The diverse and exploding digital universe: an updated

- forecast of worldwide information growth through 2011 [EB/OL]. (2018-02-01)[2018-02-25]. <http://www.ifap.ru/library/book268.pdf>.
- [2] 王媛媛, 李翔. 基于人口统计学的改进聚类模型协同过滤算法 [J]. 计算机科学, 2017, 44(3): 63-69.
WANG Yuanyuan, LI Xiang. Study on improved clustering collaborative filtering algorithm based on demography [J]. Computer Science, 2017, 44(3): 63-69.
 - [3] SHU J B, SHEN X X, LIU H, et al. A content-based recommendation algorithm for learning resources [J]. Multimedia Systems, 2018, 24(2): 163-173.
 - [4] WU X K, CHENG B, CHEN J L. Collaborative filtering service recommendation based on a novel similarity computation method [J]. IEEE Transactions on Services Computing, 2017, 10(3): 352-365.
 - [5] LI Y Y, WANG D, HE H Y, et al. Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems [J]. Neurocomputing, 2017, 249: 48-63.
 - [6] LIU W B, WANG Z D, LIU X H, et al. A survey of deep neural network architectures and their applications [J]. Neurocomputing, 2017, 234: 11-26.
 - [7] YU Y, GONG Z Q, ZHONG P, et al. Unsupervised representation learning with deep convolutional neural network for remote sensing images [M] // Lecture Notes in Computer Science. Cham, Germany: Springer International Publishing, 2017: 97-108.
 - [8] GUO L, LI N P, JIA F, et al. A recurrent neural network based health indicator for remaining useful life prediction of bearings [J]. Neurocomputing, 2017, 240(5): 98-109.
 - [9] ZHAO Z D, SHANG M S. User-based collaborative-filtering recommendation algorithms on Hadoop [C] // Proceedings of the International Conference on Knowledge Discovery and Data Mining. Piscataway, NJ: IEEE, 2010: 478-481.
 - [10] MA M M, WANG S P. Research of user-based collaborative filtering recommendation algorithm based on Hadoop [C] // Proceedings of the International Conference on Computer Information Systems and Industrial Applications. Paris, France: Atlantis Press, 2015: 63-66.
 - [11] SCHELTER S, BODEN C, MARKL V. Scalable similarity-based neighborhood methods with MapReduce [C] // Proceedings of the ACM Conference on Recommender Systems. New York, NY, USA: ACM Press, 2012: 163-170.
 - [12] 廖彬, 张陶, 国冰磊, 等. 基于 Spark 的 ItemBased 推荐算法性能优化 [J]. 计算机应用, 2017, 37(7): 1900-1905.
LIAO Bin, ZHANG Tao, GUO Binglei, et al. Performance optimization of ItemBased recommendation algorithm based on Spark [J]. Journal of Computer Applications, 2017, 37(7): 1900-1905.
 - [13] 廖彬, 张陶, 于炯, 等. 基于 Spark 的 MapReduce 相似度计算效率优化 [J]. 计算机科学, 2017, 44(8): 46-53.
LIAO Bin, ZHANG Tao, YU Jiong, et al. An efficiency optimization method for MapReduce similarity computing based on Spark [J]. Computer Science, 2017, 44(8): 46-53.
 - [14] 廖彬, 张陶, 于炯, 等. Spark DAG 优化 MapReduce 协同过滤算法 [J]. 中山大学学报(自然科学版), 2017, 56(3): 46-56.
LIAO Bin, ZHANG Tao, YU Jiong, et al. Optimization of collaborative filtering algorithm based on DAG Spark scheduling [J]. Acta Scientiarum Naturalium Universitatis Sunyatseni, 2017, 56(3): 46-56.
 - [15] 魏港明, 刘真, 李林峰, 等. 加入用户对项目属性偏好的奇异值分解推荐算法 [J]. 西安交通大学学报, 2018, 52(5): 101-107.
WEI Gangming, LIU Zhen, LI Linfeng, et al. Singular value decomposition recommendation algorithm considering user's preference for item attributes [J]. Journal of Xi'an Jiaotong University, 2018, 52(5): 101-107.
 - [16] 方耀宁, 郭云飞, 丁雪涛, 等. 一种基于局部结构的改进奇异值分解推荐算法 [J]. 电子与信息学报, 2013, 35(6): 1284-1289.
FANG Yaoning, GUO Yunfei, DING Xuetao, et al. An improved singular value decomposition recommender algorithm based on local structures [J]. Journal of Electronics & Information Technology, 2013, 35(6): 1284-1289.
 - [17] 李改, 李磊. 基于矩阵分解的协同过滤算法 [J]. 计算机工程与应用, 2011, 47(30): 4-7.
LI Gai, LI Lei. Collaborative filtering algorithm based on matrix decomposition [J]. Computer Engineering and Applications, 2011, 47(30): 4-7.
 - [18] 王永贵, 宋真真, 肖成龙. 基于改进聚类和矩阵分解的协同过滤推荐算法 [J]. 计算机应用, 2018, 38(4): 1001-1006.
WANG Yonggui, SONG Zhenzhen, XIAO Chenglong. Collaborative filtering recommendation algorithm based on improved clustering and matrix factori-

- zation [J]. *Journal of Computer Applications*, 2018, 38(4): 1001-1006.
- [19] SALAKHUTDINOV R, MNIH A, HINTON G. Restricted Boltzmann machines for collaborative filtering [C]// *International Conference on Machine Learning*. New York, NY, USA: ACM Press, 2007: 791-798.
- [20] WANG H, SHI X J, YEUNG D Y. Relational stacked denoising autoencoder for tag recommendation [C]// *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Reston, VA, USA: AAAI Press, 2015: 3052-3058.
- [21] OORD A V D, DIELEMAN S, SCHRAUWEN B. Deep content-based music recommendation [C]// *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Cham, Germany: Springer International Publishing, 2013: 2643-2651.
- [22] HAMEL P, LEMIEUX S, BENGIO Y, et al. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio [C]// *Proceedings of the 12th International Society for Music Information Retrieval Conference*. Miami, FL, USA: International Society for Music Information Retrieval, 2011: 729-734.
- [23] HUANG P S, HE X, GAO J, et al. Learning deep structured semantic models for web search using click-through data [C]// *ACM International Conference on Conference on Information & Knowledge Management*. New York, NY, USA: ACM Press, 2013: 2333-2338.
- [24] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based recommendations with recurrent neural networks [C] // *International Conference on Learning Representations*. London, UK: ICLR, 2016: 1-10.
- [25] BRÉBISSEON A D, SIMON É, AUVOLAT A, et al. Artificial neural networks applied to taxi destination prediction [C]// *European Conference on Principles of Data Mining and Knowledge Discovery*. Aachen, Germany: CEUR-WS, 2015: 40-51.
- [26] LONG C, ZHANG J, HUANG M L, et al. Estimating feature ratings through an effective review selection approach [J]. *Knowledge and Information Systems*, 2014, 38(2): 419-446.
- [27] AA T V, CHAKROUN I, HABER T. Distributed Bayesian probabilistic matrix factorization [C]// *2016 IEEE International Conference on Cluster Computing*. Piscataway, NJ, USA: IEEE, 2016: 1030-1039.
- [28] 王海艳, 董茂伟. 基于动态卷积概率矩阵分解的潜在群组推荐 [J]. *计算机研究与发展*, 2017, 54(8): 1853-1863.
- WANG Haiyan, DONG Maowei. Latent group recommendation based on dynamic probabilistic matrix factorization model integrated with CNN [J]. *Journal of Computer Research and Development*, 2017, 54(8): 1853-1863.
- [29] JABAKJI A, DAG H. Improving item-based recommendation accuracy with user's preferences on Apache Mahout [C] // *IEEE International Conference on Big Data*. Piscataway, NJ, USA: IEEE, 2016: 1742-1749.
- [30] MA W M, SHI J F, ZHAO R D. Normalizing item-based collaborative filter using context-aware scaled baseline predictor [J]. *Mathematical Problems in Engineering*, 2017(4): 1-9.
- [31] SALAKHUTDINOV R, MNIH A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo [C] // *International Conference on Machine Learning*. New York, NY, USA: ACM Press, 2008: 880-887.
- [32] LIU J T, WU C H, LIU W Y. Bayesian probabilistic matrix factorization with social relations and item contents for recommendation [J]. *Decision Support Systems*, 2013, 55(3): 838-850.
- [33] WU J M, LIU Z, CHENG G Q, et al. Online shopping recommendation with Bayesian probabilistic matrix factorization [M]// *IFIP Advances in Information and Communication Technology*. Cham, Germany: Springer International Publishing, 2017: 445-451.
- [34] LIU J, JIANG Y, LI Z C, et al. Domain-sensitive recommendation with user-item subgroup analysis [C]// *International Conference on Data Engineering*. Piscataway, NJ, USA: IEEE, 2016: 1466-1467.
- [35] LIAO C L, LEE S J. A clustering based approach to improving the efficiency of collaborative filtering recommendation [J]. *Electronic Commerce Research & Applications*, 2016, 18(8): 1-9.
- [36] ROH T H, OH K J, HAN I. The collaborative filtering recommendation based on SOM cluster-indexing CBR [J]. *Expert Systems with Applications*, 2003, 25(3): 413-423.
- [37] TSAI C F, HUNG C. Cluster ensembles in collaborative filtering recommendation [J]. *Applied Soft Computing*, 2012, 12(4): 1417-1425.

(下转第136页)