

基于双重注意力机制的降噪自编码器推荐算法^①

王倩雯^② 张延华^③ 付琼霄 李 萌 李 庆

(北京工业大学信息学部信息与通信工程学院 北京 100124)

摘 要 为了解决传统协同过滤算法的准确度因评分缺失产生的剧烈变化以及冷启动问题,本文提出了一种新的降噪自编码器推荐算法。该方法将注意力机制与辅助信息共同融入降噪自编码器中对评分与交互数据进行处理。首先针对用户交互项目动态分配注意力以学习用户偏好,然后再次通过注意力机制学习隐藏层向量、用户偏好、辅助信息的权重以获得完整评分矩阵。在公开数据集上对该算法进行实验仿真,观察算法性能。结果表明,该算法有效利用了辅助信息且准确度有明显提高。

关键词 降噪自编码器;注意力机制;推荐算法;辅助信息;深度学习

0 引 言

随着技术的不断发展,信息量持续增长,用户身处信息爆炸化环境,无法快速寻找所需内容。搜索引擎虽然能满足某些用户的需求,但并不能学习用户偏好从而主动帮助用户搜寻合适信息^[1]。而推荐系统在解决信息过载问题的同时可完成用户个性化推荐。

推荐系统主要分为三类,即基于内容的推荐、基于协同的推荐和混合推荐^[2]。其中基于协同的推荐^[3]可有效利用其他相似用户的反馈信息并推荐新鲜项目,现已经成为应用最广的推荐系统。但基于协同的推荐算法仍有明显不足,无法为零历史用户和新物品进行推荐,这就是推荐系统中常见的冷启动问题^[4]。为了缓解这些问题,研究人员通常将内容信息融入协同算法中进行混合推荐。

近年来深度学习因特征提取能力强,在多个领域表现优异^[5-6]。2007 年,在 Nexflx Prize 推荐大赛上提出将深度学习应用于推荐系统。此后,基于深

度学习的推荐系统研究受到国际学术界越来越多的关注,ACM 推荐系统年会于 2016 年召开第 1 届基于深度学习的推荐系统研究专题研讨会,国内外许多研究机构也对基于深度学习的推荐系统开展了广泛研究。Du 等人^[7]将隐因子模型与多层感知机相结合,提出了一种新的群体推荐策略。利用多层感知机获取用户与项目之间的潜在联系,并根据推荐分数设计均衡推荐策略,确保群体用户的满意程度,有效提高了群体推荐准确率。Guo 等人^[8]结合因子化机和深度学习,提出了一种基于深度网络的因子化机模型进行点击率预测,该模型无需人工特征处理,省去了繁琐操作。He 和 Chua^[9]基于因子化模型提出一种神经因子化模型,通过进行特征之间的高层次非线性交互提高预测能力,其参数少、隐层少、训练更加容易。Cui 等人^[10]则设计了一种模型特征交互图神经网络模型,可提升对特征间的复杂交互预测精度。进行建模自编码器作为深度学习模型中重要的一种结构,其学习能力已被大家认可。最早将自编码器应用于推荐领域的算法为基于编码器的协同过滤算法^[11],但它只能解决整数评分预

① 国家自然科学基金(61571021,61901011),中国博士后科学基金面上项目(2018M640032),北京市博士后科研活动经费(ZZ2019-73),朝阳区博士后科研活动经费(2019ZZ-4)和北京工业大学基础研究基金资助项目。
② 女,1996 年生,硕士生;研究方向:深度学习,推荐算法;E-mail: wangqw82196@163.com
③ 通信作者,E-mail: zhangyh@bjut.edu.cn
(收稿日期:2019-12-18)

测问题,且增加了评分矩阵的稀疏程度。

现有的方法部分已经取得了较为满意的效果,但仍有几个因素需要考虑以再度提高算法性能。之前的研究主要依靠单一的用户评分学习隐藏偏好,忽略了每个历史评分项目对预测结果的影响;将用户辅助信息直接加入推荐网络,没有考虑不同用户辅助信息的差异,无法充分利用该信息。

本文针对上述问题提出了一种基于双重注意力机制的降噪自编码器模型,主要贡献如下。

(1) 将用户与项目辅助信息融入降噪自编码器模型,扩充可利用的数据来源,解决冷启动问题。

(2) 利用注意力机制从用户历史交互项目中学习用户长期偏好以构建用户画像。

(3) 再次使用注意力机制学习用户偏好、用户辅助信息与隐藏层向量的权重,训练得到更为准确的评分。

本文在公开数据集上进行了仿真实验,结果表明本文方法准确度更高,证明了注意力机制的有效性。

1 相关工作

1.1 基于自编码器的推荐

在现有基于深度学习的研究中,大部分推荐算法是以自编码器为基础进行的。自编码器为一种无监督自学习,拥有强大的特征学习能力,可以捕捉到用户-项目间的潜在关系,这一特点使它深受研究人员欢迎。2015年,Sedhain等人^[11]提出基于自编码器的协同过滤算法,它将基于用户或基于项目的评分作为输入,预测完整评分。这项研究对两种输入模型都进行了性能评估,发现基于项目的模型表现较好,因项目评分方差较小。Wu等人^[12]在此基础上提出协同降噪自编码器模型,在模型中加入噪声数据与用户因子,提高了模型鲁棒性和推荐准确度。Chen等人^[13]提出边缘去噪模型,用更少的训练时间获得更高的推荐性能。HRC^[14]模型将自编码器与时间感知的隐因子模型相结合,利用堆栈自编码器学习项目特征解决冷启动问题。基于自编码器的混合协同过滤^[15]模型根据内容信息和隐式用户反

馈将自编码器与矩阵分解模型结合从而进行推荐。在文献[16]中,利用栈式降噪自编码器学习隐式特征,并在学习过程中加入用户与项目的辅助信息。上述方法仅是简单的将辅助信息加入模型,未考虑每个用户打分为受各因素的影响程度不同。本文模型则将注意力机制融入降噪自编码器模型中,动态学习辅助信息与用户偏好的权重,更准确地构造最终隐藏层向量,达到了更佳的预测效果。

1.2 注意力机制

注意力机制被广泛用于自然语言处理以及图像领域中,用于推荐领域的相关文章较少,直至2017年Google mind团队发表相关论文,引起了大家对注意力机制的广泛关注,相关文章也大量涌现^[17]。Xiao等人^[18]将注意力机制引入神经因子分解模型,提高了系统解释性和表示能力。Kang等人^[19]提出一种基于注意力机制的序列预测模型,从用户交互中选出少量历史数据即可预测下一项目。Chen等人^[20]利用注意力机制对多媒体推荐的隐形反馈进行建模。Seo等人^[21]提出一种基于双重注意力机制的卷积神经网络模型学习用户偏好。Zhu等人^[22]利用注意力机制学习用户动态特征。Cui等人^[23]提出一种节点图神经网络推荐模型,通过注意力机制学习各个节点的兼容性得分,从而更准确地推荐合适项目。这些方法虽然取得了不错的效果,但是仅考虑了用户交互项目,无法解决冷启动问题。本文不仅利用注意力模型学习用户偏好,还融入了用户辅助向量,无论用户是否有历史数据,均可实现准确推荐。这使得本文算法在保证精度的同时解决了冷启动问题。

2 降噪自编码器推荐算法

2.1 算法架构

基于双重注意力机制的降噪自编码器是在自编码推荐算法的基础上提出的。推荐架构如图1所示。用户辅助信息、交互信息、历史评分作为该模型的输入,通过多层网络学习用户偏好与隐藏层向量,最终补全评分矩阵。

因普通降噪自编码器只利用了评分信息,忽略

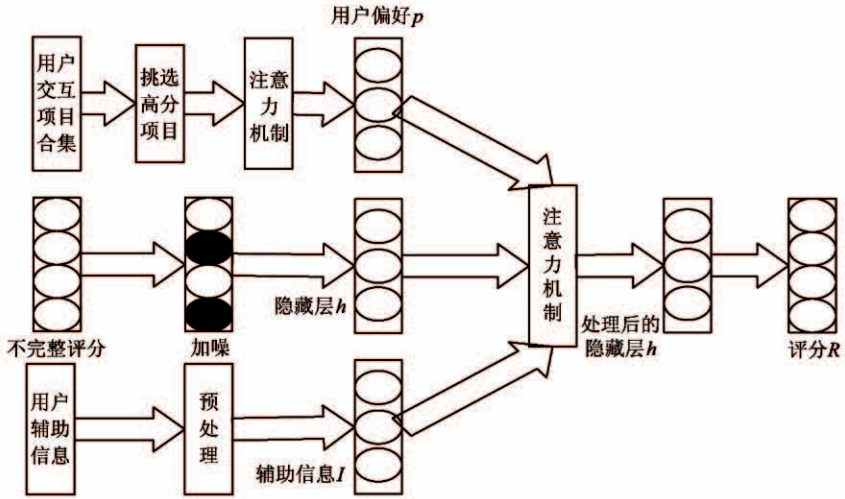


图 1 降噪自编码器架构

了用户辅助信息的影响,本文利用 word2vec 和卷积神经网络模型处理用户辅助信息。为了得到较准确的用户偏好,使用注意力机制从用户高分交互项目提取用户偏好向量。将用户偏好向量、辅助信息向量与降噪自编码器的隐藏层向量再次输入注意力机制模型,最终获得更精确的预测评分。算法伪代码如算法 1 所示。

```

算法 1 推荐算法

Input: Rating matrix  $R$ , user set  $User$ , item set  $Item$ ,
rating constant  $a$ , user auxiliary information  $X$ , item auxiliary
information  $Y$ .

Output: Prediction rating matrix  $R'$ .

1: Processing  $X$ ,  $Y$  by word2vec to obtain user and item
auxiliary information matrix  $U$ ,  $I$ 
2: function History ( $R$ ,  $User$ ,  $Item$ ,  $I$ ,  $a$ )
3: for  $u$  in  $User$  do
4:   for  $i$  in  $Item$  do
5:     if  $R(u, i) > a$  then
6:       preference.append ( $I[i]$ )
7:     end if
8:   end for
9: end for
10: return preference
11: end function
12: function Encoder attention ( $R$ ,  $U$ ,  $Pf$ , preference)
13: Repeat
14:   compute  $Pf$  according to Eqns. (1,2,3,4)
15:   compute  $h1$  according to Eqns. (9)
16:   compute  $h^*$  according to Eqns. (10,11,12)
```

```

17:   compute  $R'$  according to Eqns. (13)
18: for each parameter in  $Pf, h1, h^*, R'$  do
19:   update parameter;
20: end for
21: return  $R'$ 
22: end function
```

2.2 信息预处理

用户评分受到诸多因素的影响,例如用户年龄、性别、地区以及电影名称、类别等,因此辅助信息在算法中显得尤为重要。为了获得用户与项目的辅助信息向量,需要对文本信息进行处理。用户与项目辅助信息如表 1 和表 2 所示。

表 1 用户信息			
User ID	Gender	Age	Occupation
1	F	1	10
2	M	56	16
3	M	25	15
...	...	...	...

表 2 电影信息		
Movie ID	Title	Genres
1	Toy Story	Animation/Children's/Comedy
2	Jumanji	Adventure/Children's/Fantasy
3	Grumpier Old Men	Comedy/Romance
...	...	...

将以上辅助文本信息均利用 word2vec 方法转化为向量,但电影标题需先除去无关信息,如年份。并将处理后的电影标题矩阵输入卷积神经网络,提取标题特征向量。若一个电影有多个类型向量则将类型向量相加获得最终类型向量表示。用户年龄、性别、职位等数字信息则转化为相应个数的向量,如

年龄共分为 7 类,转化为 7 个 a 维向量。将处理后的用户与项目的多个辅助信息向量分别输入全连接层获得最终用户与项目辅助向量,由 U 和 I 表示,过程如图 2 所示。图 2(a)为用户信息处理流程图,图 2(b)为项目信息处理流程图。

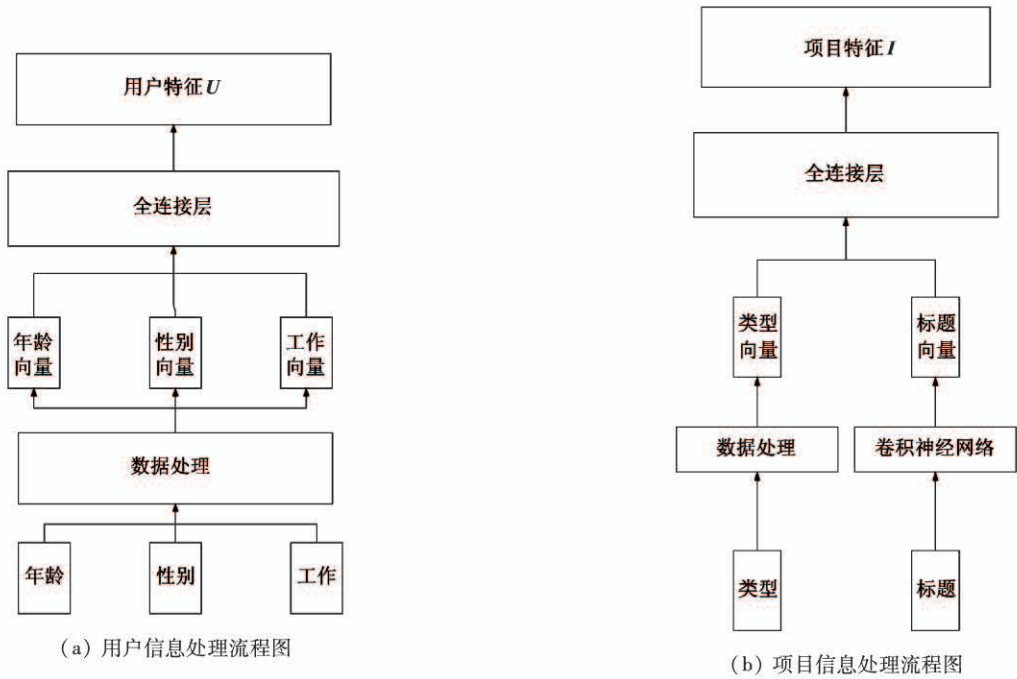
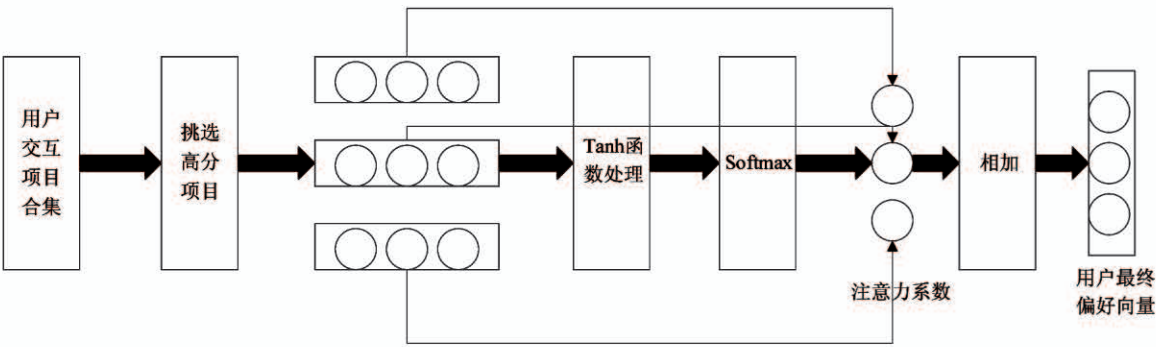


图 2 用户与项目信息处理过程



2.3 获取用户偏好

为了获取准确的用户偏好,利用注意力机制处理用户交互项目,如图 3 所示。根据用户历史评分挑选出用户偏爱的 n 个项目,将项目矩阵输入注意力机制模型中,首先将项目矩阵经过激活函数处理,计算注意力系数并进行归一化处理,将得到的注意力系数与对应项目向量相乘,最终将获得不同注意

力的项目向量相加得到用户偏好矩阵。公式如下:

$$P_i^o = f(W^o P_i + b^o) \tag{1}$$
$$A_o = \text{softmax}(P_i^o W^A) \tag{2}$$
$$P_i^f = A_o P_i^o \tag{3}$$
$$P^f = \sum_{i=0}^n P_i^f \tag{4}$$

其中, P_i 为用户偏爱的第 i 个项目, W^o 为项目权重,

b^o 为项目偏置, P_i^o 为第 i 个经过处理的用户偏爱项目, A_o 为经过 softmax 计算的归一化注意力系数, P_i^f 为获得不同注意力系数的项目偏好, P^f 为用户最终项目偏好。

2.4 基于注意力机制的降噪自编码器

2.4.1 自编码器

自动编码器(autoencoder, AE)采用前馈神经网络算法,由编码器和解码器两部分组成,包括输入层、隐藏层和输出层。自动编码器拥有 3 层全连接神经网络结构,输入层将 n 维输入向量 x 通过编码函数 f_θ 映射到隐藏层 h , 具体计算过程如式(5)所示:

$$h = f_\theta(x) = s(Wx + b) \tag{5}$$

其中, W 为输入层到隐藏层的权重, b 为隐藏层神经元的偏置, s 为 sigmoid 激活函数。

输出层则将 h 反向变换,得到原始输入 x 的重构 y , 如式(6)所示:

$$y = g_\theta(h) = s(W'h + b') \tag{6}$$

其中, W' 和 b' 为输出层的权重与偏置。通过不断地调整各层参数,可得到最小重构误差 J , 即输入输

出尽可能接近。

$$J = \sum_{x \in D} L(x, y) \tag{7}$$

其中, D 为输入的向量合集, L 为重构误差函数。

2.4.2 加入注意力机制

如图 4 所示,将用户评分 R_u 加入高斯噪声后输入编码器得到隐藏层 h_1 , 将其与用户辅助向量 I 、用户最终偏好向量 P^f 输入注意力机制,计算其注意力系数并将数值映射到 $[0, 1]$ 区间内,将处理后的注意力系数与对应向量相乘,并将获得不同注意力系数的向量相加得到新的隐藏层向量 h^* , 输入解码器获得完整预测评分 R , 公式如下:

$$R'_u = R_u + N \tag{8}$$

$$h_1 = f(W^h R'_u + b^h) \tag{9}$$

$$M = f\left(W^m \begin{bmatrix} P^f \\ h_1 \\ I \end{bmatrix} + b^m\right) \tag{10}$$

$$A_m = \text{softmax}(MW^{A_m}) \tag{11}$$

$$h^* = A_m \cdot M \tag{12}$$

$$R = f(W^* h^* + b^*) \tag{13}$$

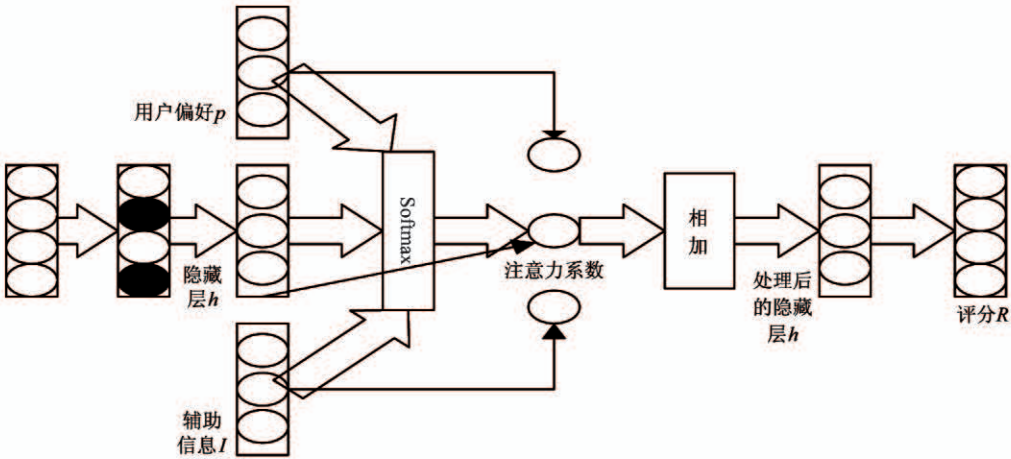


图 4 融入注意力机制的隐藏层模型

R'_u 为用户评分 R_u 加噪声 N 后的输入, W^h 为输入到隐藏层的权重, b^h 为输入到隐藏层的偏置, M 为处理后的注意层输入, W^m 为输入权重, b^m 为输入偏置, A_m 为归一化后的注意力系数, h^* 为新的隐藏层, W^* 和 b^* 则为隐藏层到输出层的权重与偏置。上述参数利用 Adam 算法优化,为了防止过拟合,在重构误差函数里加入了正则化参数。

3 实验数据

3.1 数据集

实验使用的数据集为 movielens, 来源于 movielens 电影视频网站上用户对电影的评分,由 grouplens 小组整理得来,被广泛用于测试推荐算法的性能中。具体情况如表 3,共有近 10 万条评分信息。

表 3 数据集信息

数据集	用户数	电影数	评分数
Movielens 1 M	6 040	3 544	993 482

3.2 评价标准

推荐算法性能的重要度量标准是评分预测准确度。平均绝对误差(MAE)和均方根误差(RMSE)两个标准都是用于观察预测评分与真实评分的偏离程度,数值越小表明算法性能越好。定义如下:

$$MAE = \frac{\sum_{j=1}^N (r_j - r_j^*)^2}{N}$$
 (14)

$$RMSE = \sqrt{\frac{\sum_{j=1}^N (r_j - r_j^*)^2}{N}}$$
 (15)

其中, N 为预测的评分数量, r^* 为算法评分预测值, r 为用户实际评分值。为了使算法性能对比更全面,本文同时记录了 MAE 与 RMSE。

3.3 实验环境

本文实验均在处理器为 Intel(R) Core(TM) i3-4170 CPU @ 3.70 GHz, 内存为 4 GB, 操作系统为 Ubuntu 17.10 的台式计算机上进行, 编程语言为 python。

3.4 实验设计

在数据集上随机抽取 80% 作为训练集, 剩余数据作为测试集, 观察调整隐藏层节点数、学习率、batch size、误差优化方法等参数下的算法性能, 结果如图 5 和图 6 所示。隐藏层节点过少时, 网络训练性能过差, 但节点过多时, 网络训练时间延长, 且容易陷入局部极小点。学习率是模型较为重要的参数, 过小时学习速率太低, 收敛速度慢, 学习率过大则会导致模型无法收敛, 因此本文测试了学习率在 0.1 ~ 0.00001 范围内的模型性能来判断最佳学习率。batch size 数值过小时花费时间长, 震荡范围大, 反

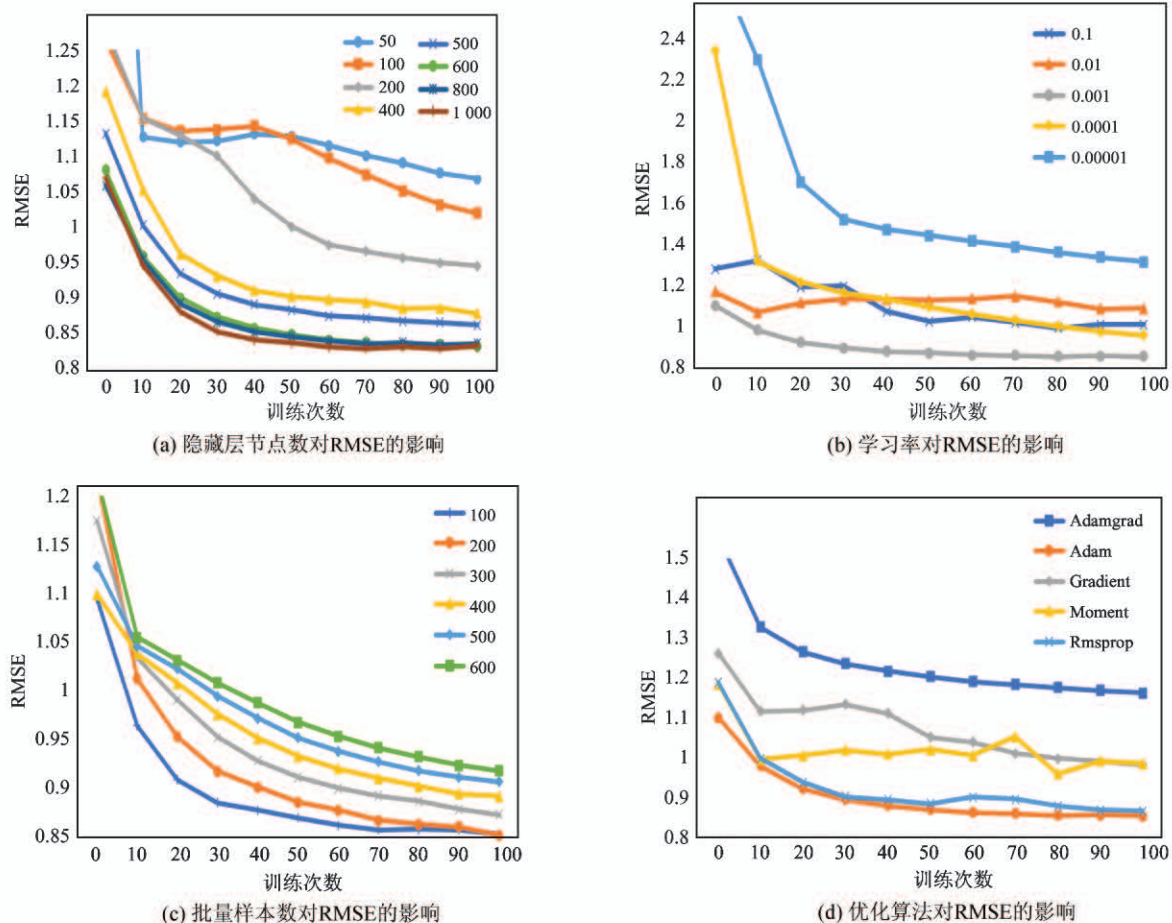


图 5 不同参数对 RMSE 的影响

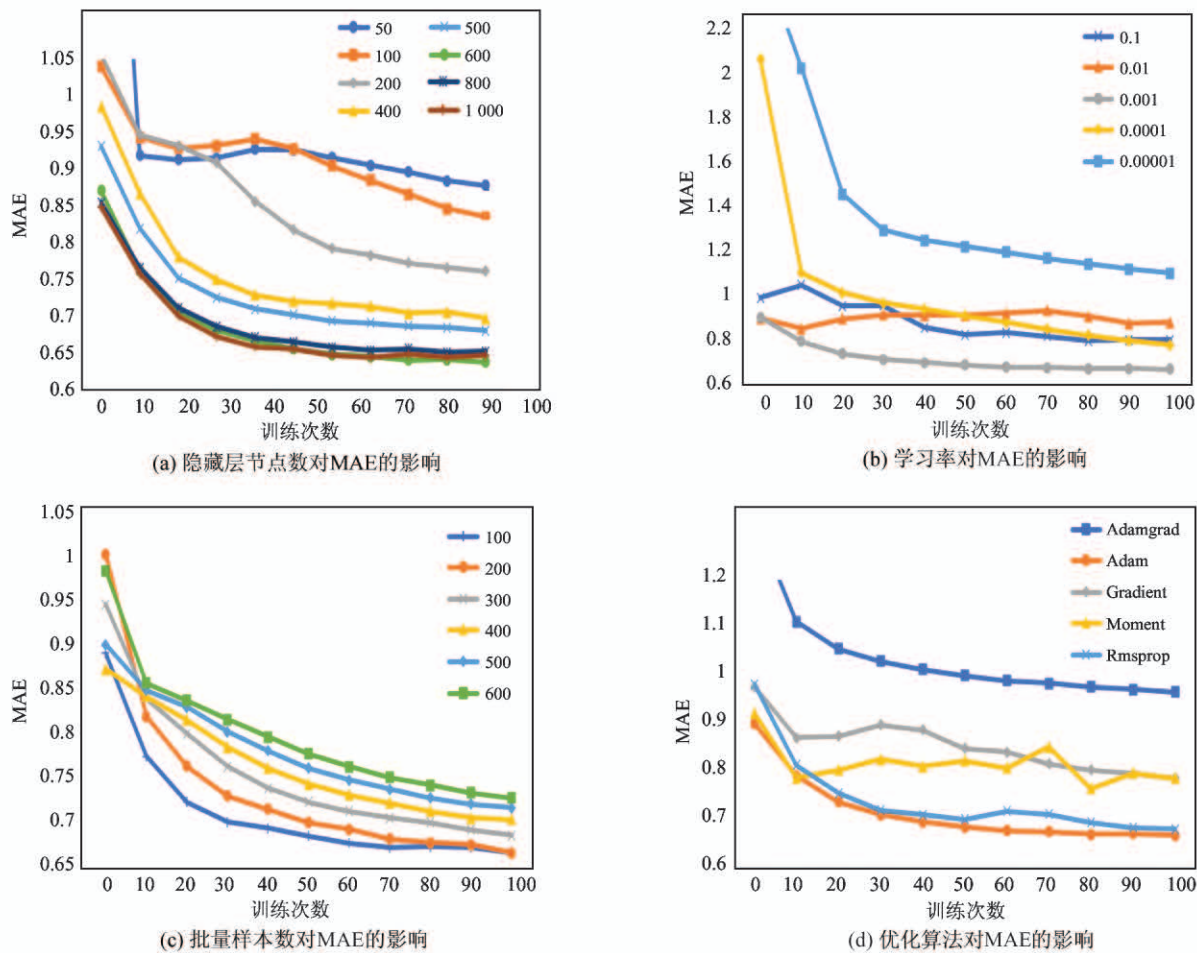


图 6 不同参数对 MAE 的影响

之易陷入局部最小值。而不同的优化算法对性能也有着较大影响,因此本文采用了常用的几种优化算法进行测试。从下图可以看出隐藏层为 600,学习率为 0.001,批样本数为 100,优化算法为 Adam 时,算法性能最优。

同样随机抽取 20% 的数据作为测试集,使用 Autorec-U、Autorec-I、NRR、LSTM-A 作对照方法,并

根据参考文献分别设定各个算法的最优参数,进行对照实验,如图 7 和图 8 所示。Autorec-U 是用户端的自动编码器算法,将用户评分集合输入自动编码器以补全用户评分矩阵从而进行推荐。Autorec-I 为项目端的自动编码器算法,将项目评分集合输入自动编码器以补全项目评分矩阵从而进行推荐。NRR 是基于门控递归神经网络的推荐算法,通过提取用

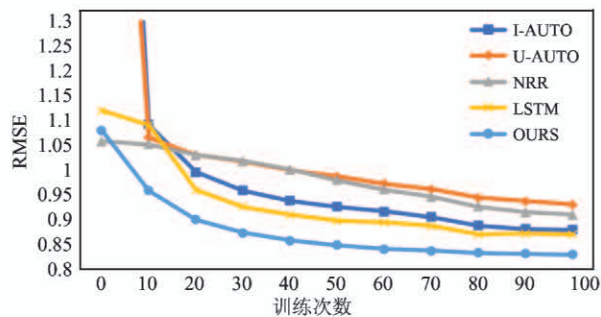


图 7 不同算法 RMSE 对比图

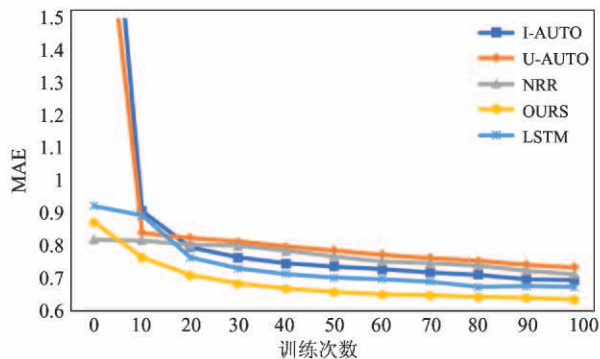


图 8 不同算法 MAE 对比图

户与项目的特征进行评分预测。LSTM-A 是融合注意力 LSTM 的协同过滤推荐算法,用自编码器与长短期记忆网络提取用户与项目特征,融入概率矩阵分解模型中预测评分。

从图 7 和图 8 可以看出,无论是 RMSE 还是 MAE,本文算法均优于其他算法。

4 结 论

传统推荐方法精度较低,忽略了用户对不同项目的偏好差异,针对这些问题,本文提出了基于双重注意力机制的降噪自编码器推荐算法。利用双重注意力机制获得用户潜在的项目偏好与用户辅助信息、评分隐藏层、用户偏好对评分的不同影响,最终获得更准确的评分矩阵。为了测试本文方法的推荐性能,通过实验演示了方法本身的调优过程,并从预测准确度方面同其他方法进行对比。实验结果显示,本文方法在预测精度方面优于其他方法。未来可考虑增加其他因素,例如调整用户近期评分权重可能会进一步增加评分精度。

参考文献

- [1] 代丽,樊粤湘. 个性化推荐系统综述[J]. 计算机时代, 2019(6):9-11,15
- [2] Guo K, Xu T, Tang Y, et al. An incremental collaborative filtering based recommendation framework for personalized websites[C]//2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 2018: 181-186
- [3] Deldjoo Y, Schedl M, Elahi M. Movie genome recommender: a novel recommender system based on multimedia content[C]//2019 International Conference on Content-Based Multimedia Indexing (CBMI), Dublin, Ireland, 2019: 1-4
- [4] Yang Y, Xu Y, Wang E, et al. Improving existing collaborative filtering recommendations via serendipity-based algorithm[J]. *IEEE Transactions on Multimedia*, 2018, 20(7):1888-1900
- [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*, 2016, 40(4):834-848
- [6] Li H, Ota K, Dong M. Learning IoT in edge: deep learning for the Internet of things with edge computing[J]. *IEEE Network*, 2018, 32(1):96-101
- [7] Du J, Li L, Gu P, et al. A group recommendation approach based on neural network collaborative filtering[C]//2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), Macao, China, 2019: 148-154
- [8] Guo H, Tang R, Ye Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction[C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 2017: 1725-1731
- [9] He X, Chua T S. Neural factorization machines for sparse predictive analytics[C]//Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 2017:355-364
- [10] Cui Z, Li Z, Wu S, et al. Fi-GNN: modeling feature interactions via graph neural networks for CTR prediction[C]//ACM International Conference on Information and Knowledge Management (CIKM), Beijing, China, 2019:539-548
- [11] Sedhain S, Menon A K, Sanner S, et al. Autorec: autoencoders meet collaborative filtering[C]//Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 2015:111-112
- [12] Wu Y, Dubois C, Zheng A X, et al. Collaborative denoising auto-encoders for Top-N recommender systems[C]//Proceedings of the 9th ACM International Conference, New York, USA, 2016:153-162
- [13] Chen M, Weinberger K, Sha F, et al. Marginalized denoising auto-encoders for nonlinear representations[C]//International Conference on Machine Learning, Edinburgh, UK, 2014:1476-1484
- [14] Wei J, He J, Chen K, et al. Collaborative filtering and deep learning based recommendation system for cold start

- items[J]. *Expert Systems with Applications*, 2017(69): 29-39
- [15] 王东, 薛峰, 刘凯, 等. 基于物品融合自编码器的协同过滤推荐算法[J]. 计算机应用, 2019, 39(z1): 84-87
- [16] Dong X, Yu L, Wu Z H, et al. A hybrid collaborative filtering model with deep structure for recommender systems[C] // Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017: 1309-1315
- [17] Li H, Li J. Recognition of robot based on attention mechanism and convolutional neural network[C] // 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019: 2578-2584
- [18] Xiao J, Ye H, He X, et al. Attentional factorization machines: learning the weight of feature interactions via attention networks[C] // Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 2017: 3119-3125
- [19] Kang W, Mc Auley J. Self-attentive sequential recommendation[C] // 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 2018: 197-206
- [20] Chen J, Zhang H, He X, et al. Attentive collaborative filtering: multimedia recommendation with item-and component-level attention[C] // Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 2017: 335-344
- [21] Seo S Y, Huang J, Yang H, et al. Interpretable convolutional neural networks with dual local and global attention for review rating prediction[C] // Proceedings of the 11th ACM Conference on Recommender Systems, Como, Italy, 2017: 297-305
- [22] Zhu J, Guo X, Ma S. Attentive neural network recommendation for E-commerce based on various implicit feedback[C] // 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018: 889-892
- [23] Cui Z, Li Z, Wu S, et al. Dressing as a whole: outfit compatibility learning based on node-wise graph neural networks[C] // World Wide Web Conference, San Francisco, USA, 2019: 307-317

A denoising autoencoder algorithm for recommendation based on dual attention mechanism

Wang Qianwen, Zhang Yanhua, Fu Qiongxiao, Li Meng, Li Qing

(Faculty of Information Technology, Beijing University of Technology, Beijing 100124)

Abstract

In order to solve the problem that the changes in accuracy of the traditional collaborative filtering algorithm due to the lack of ratings and cold start, a new denoising autoencoder algorithm is proposed for recommendation. In this algorithm, attention mechanism and auxiliary information are integrated into the denoising autoencoder model to process the ratings and interactive data. Attention is dynamically allocated to user interaction items to learn user preferences, and then the weights of hidden layer vector, user preferences, and auxiliary information are learned again through the attention mechanism to obtain the complete ratings matrix. The algorithm is simulated on the open data set to observe its performance. The results show that the algorithm uses the auxiliary information effectively and the accuracy is improved obviously.

Key words: denoising autoencoder, attention mechanism, recommendation algorithm, auxiliary information, deep learning