

融合偏置深度学习的距离分解 Top-N 推荐算法

钱梦薇¹ 过 弋^{1,2,3}

1 华东理工大学信息科学与工程学院 上海 200237

2 大数据流通与交易技术国家工程实验室-商业智能与可视化技术研究中心 上海 200436

3 上海大数据与互联网受众工程技术研究中心 上海 200072

(ww649960358@foxmail.com)

摘 要 针对传统矩阵分解算法大多是浅层的线性模型,难以学习到深层次的用户和物品的隐特征向量,且在数据稀疏的情况下容易产生过拟合的问题,文中提出一种融合偏置深度学习的矩阵分解算法,在解决数据稀疏问题的同时,还能学习到表征能力更强的距离特征向量。首先,通过用户与物品的显式和隐式数据构建用户与物品的交互矩阵,并将交互矩阵转化为相应的距离矩阵;其次,将距离矩阵分别按行和按列输入加入偏置层的深度神经网络,学习得到具有非线性特征的用户和物品的距离特征向量;最后,根据用户和物品的距离特征向量计算用户和物品之间的距离,用距离值对物品按升序排列,生成 Top-N 的推荐列表。在 4 个真实数据集上进行实验,采用 Precision, Recall, MAP, MRR 和 NDCG 指标进行评估,结果表明所提算法在上述指标方面相比其他主流推荐算法有明显提升。

关键词: 偏置层;深度学习;距离分解;物品排序

中图法分类号 TP191

Biased Deep Distance Factorization Algorithm for Top-N Recommendation

QIAN Meng-wei¹ and GUO Yi^{1,2,3}

1 School of Information Science and Engineering, East University of Science and Technology, Shanghai 200237, China

2 Business Intelligence and Visualization Research Center, National Engineering Laboratory for Big Data Distribution and Exchange Technologies, Shanghai 200436, China

3 Shanghai Internet Big Data Engineering Technology Research Center, Shanghai 200072, China

Abstract Since traditional matrix factorization algorithms are mostly based on shallow linear models, it is difficult to learn latent factors of users and items at a deep level. When the dataset is sparse, it is inclined to overfitting. To deal with the problem, this paper proposes a biased deep distance factorization algorithm, which can not only solve the data sparse problem, but also learn the distance feature vectors with stronger characterization capabilities. Firstly, the interaction matrix is constructed through the explicit and implicit data of the user and the item. Then the interaction matrix is converted into the corresponding distance matrix. Secondly, the distance matrix is input into the depth of the bias layer by row and column respectively. The neural network learns the distance feature vectors of users and items with non-linear features. Finally, the distance between the user and the item is calculated according to the distance feature vectors. Top-N item recommended list is generated according to the distance value. The experimental results show that Precision, Recall, MAP, MRR and NDCG of this algorithm are significantly improved compared to other mainstream recommendation algorithms on four different datasets.

Keywords Biased layer, Deep learning, Distance factorization, Item ranking

1 引言

近年来,随着推荐算法研究的不断推进,推荐系统越来越多地被应用在电商、文娱、社交等领域的平台上,有效解决了

用户在面对种类丰富的物品时无从挑选的问题,极大地缓解了信息过载现象给用户带来的压力。其中,物品排序就是推荐系统研究的重要问题之一,它能够根据不同用户的喜好为用户生成一个可供挑选的个性化推荐列表,一方面降低了用

到稿日期:2020-08-20 返修日期:2020-11-01

基金项目:国家重点研发计划(2018YFC0807105);国家自然科学基金项目(61462073);上海市科学技术委员会科研计划项目(17DZ1101003, 18511106602, 18DZ2252300)

This work was supported by the National Key Research and Development Program of China(2018YFC0807105), National Natural Science Foundation of China(61462073) and Science and Technology Committee of Shanghai Municipality(17DZ1101003, 18511106602, 18DZ2252300).

通信作者:过弋(guoyi@ecust.edu.cn)

户从海量物品中检索的时间成本,另一方面推荐列表中实时生成的商品也可能是用户未曾接触过的,给用户带来了一定的新奇感。

矩阵分解是传统推荐算法中比较经典的算法,它擅长根据已知的观测值来获得用户和物品的潜在特征向量,但是在数据稀疏的情况下,用矩阵分解单一的线性模型对特征向量进行交互,进而预测矩阵的缺失项时,往往无法达到预期效果。并且,矩阵分解的点积运算也不满足三角不等式,无法准确衡量不同用户和物品之间的相似度关系。为了克服矩阵分解的这些弊端,Zhang 等^[1]提出了将矩阵分解转化为距离分解的新思路,将用户和物品的评分矩阵转化为距离矩阵,对距离矩阵进行分解,在降低矩阵稀疏度的同时,对用户和物品的距离特征矩阵计算欧氏距离,完美解决了矩阵分解中点积运算无法克服的矛盾。

随着机器学习的发展,深度学习作为其中的一个重要分支,已经在自然语言处理^[2]、图像处理^[3]和机器翻译^[4]中取得了一定的成效。越来越多的专家学者开始尝试在推荐任务中使用深度学习来提高推荐系统的推荐性能,如 Cheng 等^[5]为 Google Play 设计的浅层和深层联合训练的框架,结合了浅层模型的记忆能力和深层模型的泛化能力,保证了单模型对推荐结果的准确性和扩展性。Yang 等^[6]用增加了注意力机制和闸门机制的神经网络来分别训练用户、物品和推荐列表的特征向量,同时采用二维卷积层来训练“用户-物品”对的联合特征,极大地提高了对物品推荐得分预测的准确度。由此可见,深度学习在推荐系统中具有良好的特征学习能力,能够学习到数据中隐含的非线性结构特征。但现有的深度学习模型常常忽略用户打分偏好和物品得分倾向特征的学习,且大多数模型需要依靠辅助信息才能达到较好的推荐效果,但在许多领域中,用户和物品的辅助信息往往难以获得。

因此,本文尝试融合距离分解和深度学习,提出了一种基于距离分解的偏置深度学习推荐算法,相比其他推荐算法,该算法既解决了矩阵分解的点积运算不满足三角不等式的问题,又通过深度神经网络学习到了表征能力更强的用户和物品的距离特征向量。此外,本文在传统矩阵分解算法中用户偏置项和物品偏置项的启发下,发现偏置项在噪声较大的隐式反馈数据中能够有效减少假正例,同时在显式评分数据中能够良好地保留用户和物品的偏好信息。基于偏置项的这些优点,本文考虑在深度神经网络结构中单独构建一层偏置层,直接将用户和物品的距离值映射到低维空间,用于保留原始信息,并捕获隐藏的不同用户的评分偏好和不同物品的得分倾向信息,在 4 个真实数据集上证明了该算法在物品排序和生成 Top-N 推荐列表任务中的优越性。

2 相关工作

2.1 距离分解

距离分解最早由 Zhang 等^[1]在 2018 年提出,将用户和物品的评分矩阵转化为距离矩阵,再将距离矩阵分解为用户和物品的 k 维距离特征向量,即每个用户和物品都在 k 维空间里拥有一个坐标,通过计算用户和物品之间的欧氏距离得到距离值,从而预测用户对物品的评分。模型的流

程图解如图 1 所示。

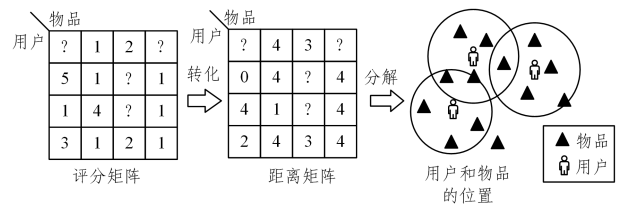


图 1 距离分解图解

Fig. 1 Illustration of distance factorization

距离矩阵中的距离值定义为:

$$Distance(u, i) = \text{Max Sim} - \text{Sim}(u, i) \quad (1)$$

其中, Max Sim 表示评分值或隐式反馈值的最大值, $\text{Sim}(u, i)$ 表示用户 u 对物品 i 的评分或隐式反馈值。对距离矩阵分解后得到的用户距离特征向量 \mathbf{P}_u 和物品距离特征向量 \mathbf{Q}_i 进行距离值计算。

$$D(u, i) = \|\mathbf{P}_u - \mathbf{Q}_i\|_2^2 \quad (2)$$

其中, $\mathbf{P}_u \in \mathbf{R}^k$ 且 $\mathbf{Q}_i \in \mathbf{R}^k$, 其分别表示用户和物品在 k 维空间内的坐标。

由于用户和物品的评分存在一定的偏差,相应地,线性变换之后的距离值也存在偏差,因此,最后的预测距离值也需要考虑偏置项:

$$\hat{Y} = D(u, i) + b_u + b_i + \mu \quad (3)$$

Zhang 等^[7]在 2020 年又重新对文献[1]进行了修改,在不同的数据集上证明了距离分解的推荐效果不仅优于传统的基于用户和基于物品的协同过滤算法,同时还超越了一些经典的基于神经网络的推荐算法。Dai 等^[8]基于距离分解,增加了对物品联合矩阵的学习,物品联合矩阵中的元素表示两个物品同时出现在用户的交互物品列表中的次数,将距离矩阵和物品联合矩阵共同作为距离分解模型的输入,能够更好地学习到用户和物品在 k 维空间内的特征向量。

尽管距离分解已经表现出了超越矩阵分解的推荐性能,但目前对距离分解的研究仍然不多。本文尝试继续研究距离分解,在其基础上致力于训练出表征能力更强的距离特征向量,从而为用户推荐其更感兴趣的物品列表。

2.2 深度推荐算法

随着深度学习的快速发展,近几年将深度学习应用于推荐算法的研究成果也不断涌现,本文按推荐系统的任务,将深度推荐算法划分为 3 类:评分预测、Top-N 推荐和序列推荐。每一类的具体代表算法如表 1 所列。

表 1 深度推荐算法分类汇总

Table 1 Classification of deep recommendation algorithms

类别	典型算法
评分预测	MF, FM, NNMF, U-AutoRec, I-AutoRec, NRR, NFM, AFM
Top-N 推荐	BPRMF, CML, CADE, MLP, GMF, NeuMF, LRML
序列推荐	PRME, LSTM, GRU4Rec, CASER, AttRec

2.2.1 评分预测

在评分预测任务中, MF (Matrix Factorization) 和 FM (Factorization Machine) 都是比较传统经典的算法。NNMF (Neural Network Matrix Factorization)^[9]用一个简单的多层反馈机制的神经网络代替了矩阵分解的点积运算。

U-AutoRec 和 I-AutoRec^[10] 分别是基于用户和物品的协同过滤,用自动编码器从评分矩阵中学习用户或物品的特征向量,但在数据稀疏的情况下,模型的学习能力不太理想。NRR (Neural Rating Regression)^[11] 在从评分矩阵中提取特征的同时,融入了从用户评论中提取的语义特征,丰富了用户特征向量包含的信息。NFM (Neural Factorization Machines)^[12] 融合了分解机和神经网络,能够实现高阶交互和非线性交互。AFM (Attentional Factorization Machines)^[13] 在分解机的基础上增加了注意力机制,在学习不同交叉特征组合的重要性上具有良好的效果。

2.2.2 Top-N 推荐

Top-N 推荐不直接根据预测的评分值对物品进行排序,由于在预测值中,只要有一个值的偏差过大,就会影响整个推荐列表中物品的排序。因此,Top-N 推荐的常规做法是将评分值处理为 1,即每一个评分值都代表用户和物品的一次交互,基于这种交互信息预测用户可能对物品有兴趣的概率。

BPRMF (Bayesian Personalized Ranking Matrix Factorization)^[14] 基于贝叶斯的极大化后验概率,构造了用户对不同物品喜好的偏序关系,能够利用隐式反馈解决推荐排序问题。CML (Collaborative Metric Learning)^[15] 在学习用户和物品直接潜在关系的同时,还学习用户与用户之间,以及物品与物品之间的相似度,有效减小了用户与喜欢的物品之间的距离,增大了用户与不喜欢物品之间的距离。CADE (Collaborative Denoising Auto-Encoders)^[16] 是一种去噪自动编码器,它的训练集是被破坏过的数据,目的是使隐藏层发现更健壮的特性。MLP 则是大多数神经网络中经常用到的多层感知器,GMF (Generalized Matrix Factorization) 是广义矩阵分解,它会对矩阵分解后点积运算的结果进行加权。NeuMF (Neural Matrix Factorization)^[17] 将 MLP 和 GMF 相结合,使两个模型分别从嵌入层中学习特征,最后将两个模型学习到的特征向量以串联的方式结合起来,既保留了 MLP 用非线性核学习交互函数的优势,又融合了 GMF 用线性核建模交互的潜在特征。LRML (Latent Relational Metric Learning)^[18] 在网络中增加了注意力机制和关系特征向量,用于更好地表征用户和物品之间的关系。

2.2.3 序列推荐

为了对不同的用户及其序列信息进行联合建模,PRME (Personalized Ranking Metric Embedding)^[19] 提出了成对排序的距离嵌入,良好地平衡了个人喜好与序列信息。LSTM (Long Short-Term Memory) 对用户的长短期兴趣建模,能动态捕捉用户的兴趣变化。GRU4Rec^[20] 用门控循环单元来模拟基于会话的点击序列推荐。CASER^[21] 将 CNN 学习到的物品的序列特征与用户的特征向量做拼接,最终输入 MLP 完成预测。AttRec^[22] 采用自注意力机制来学习物品的序列信息,结合用户的特征向量建模用户的长期偏好。

本文的主要目标是提升 Top-N 推荐的效果,为用户生成个性化推荐列表。目前的很多深度推荐算法都需要对用户和物品的辅助信息进行特征抽取,而在许多领域的数据集中,这些辅助信息难以获取。并且,这些深度网络基本都忽略了用户的打分偏好和物品的得分倾向。因此,本文仅根据用户对物品的评分信息建模,在构建的深度网络中增加了偏置层,实

现了一个具有较强的通用性和可扩展性的推荐列表生成算法。

3 融合偏置深度学习的距离分解算法

本文的网络结构采用双塔模型,分别对用户和物品进行训练。该网络由输入层、嵌入层、多层感知器层、偏置层和输出层组成,如图 2 所示。用户的距离矩阵输入网络后,由嵌入层分别按行和按列映射成 k 维的潜在距离特征向量,多层感知器对潜在距离特征向量进行非线性映射,结合偏置层捕捉的用户和物品的偏好信息,分别输出同时包含线性特征与非线性特征的用户和物品的距离特征向量 P_u 和 Q_i ,并计算两个距离特征向量之间的距离。

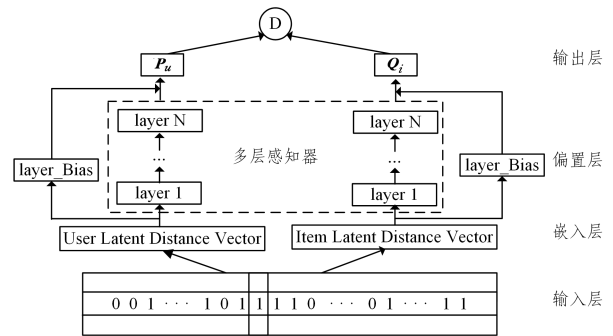


图 2 融合偏置深度学习的矩阵分解网络结构图

Fig. 2 Network structure of biased deep distance factorization

3.1 输入层

网络的输入为用户和物品的距离矩阵。在物品排序中,常常将用户 u 对物品 i 的显式评分 R_{ui} 处理为交互信息 Y_{ui} ,即:

$$Y_{ui} = \begin{cases} 1, & R_{ui} \neq 0 \\ 0, & R_{ui} = 0 \end{cases} \quad (4)$$

将用户未对物品给出评分的项标记为 0,有过评分的项标记为 1,生成一个交互矩阵。再根据图 1 中距离矩阵的转化过程,将交互矩阵转化为对应的距离矩阵。其中,用户 u 与物品 i 的距离值 D_{ui} 为:

$$D_{ui} = 1 - Y_{ui} \quad (5)$$

3.2 嵌入层

将距离矩阵分别按行和按列映射到 k 维空间,得到:

$$P_u' = \sigma(D_u * W_u) \quad (6)$$

$$Q_i' = \sigma(D_i * W_i) \quad (7)$$

其中, P_u' 和 Q_i' 分别是非线性变换之后的用户和物品的潜在距离特征向量, W_u 和 W_i 分别是用户和物品的潜在距离特征映射矩阵,可通过反向传播训练得到。 $\sigma(x)$ 为非线性激活函数,激活函数的选择在第 4 节的参数性能实验中进行了全面分析。

3.3 多层感知器层

嵌入层得到的潜在距离特征向量还需要经过多层感知器的多层非线性映射,才能得到表征能力更强的距离特征向量。

以用户 P_u 的多层感知器训练为例,记隐藏层为 $l_i, i = 2, \dots, N-1$,网络的层数为 N ,每层感知器的权重矩阵为 W_i ,偏置项为 b_i ,得到的隐式特征向量为 P_u^* ,则:

$$l_1 = W_1 P_u' \quad (8)$$

$$l_i = f(\mathbf{W}_{i-1} l_{i-1} + \mathbf{b}_{i-1}), i=2, \dots, N-1 \quad (9)$$

$$\mathbf{P}_u^* = f(\mathbf{W}_N l_{N-1} + \mathbf{b}_N) \quad (10)$$

其中, $f(x)$ 为感知器的激活函数, 感知器输出的距离特征向量为:

$$\mathbf{P}_u^* = f_{UN}(\dots f_{U3}(\mathbf{W}_{U2} f_{U2}(\mathbf{W}_{U1} \mathbf{P}_u')) \quad (11)$$

$$\mathbf{Q}_i^* = f_{IN}(\dots f_{I3}(\mathbf{W}_{I2} f_{I2}(\mathbf{W}_{I1} \mathbf{Q}_i')) \quad (12)$$

其中, \mathbf{P}_u^* 和 \mathbf{Q}_i^* 为多层感知器训练得到的用户和物品的距离特征向量。 \mathbf{W}_{U1} 和 \mathbf{W}_{I1} 是深度网络第一层的权重矩阵, \mathbf{W}_{U2} 和 \mathbf{W}_{I2} 为第二层的权重矩阵。 f_{U2} 和 f_{I2} 是深度网络第二层的激活函数, f_{U3} 和 f_{I3} 为第三层的激活函数, f_{UN} 和 f_{IN} 则对应第 N 层的激活函数。

3.4 偏置层

考虑到式(3)中的偏置项能够一定程度地获取用户的打分习惯和物品的得分倾向。而目前几乎所有的主流深度推荐模型都忽略了对偏置信息的学习。因此,为了在网络中进一步深入学习用户和物品隐藏的偏置信息,本文引入偏置层,即图2中的 layer_Bias,用偏置层直接将用户和物品的距离特征向量映射到一个低维空间,考虑到网络的输入是距离值,偏置层能够有效保留用户和物品的原始信息,从而保留用户和物品的偏置信息。

$$\bar{\mathbf{P}}_u = (\bar{\mathbf{W}}_u \mathbf{P}_u' + \bar{\mathbf{b}}_u) \quad (13)$$

$$\bar{\mathbf{Q}}_i = (\bar{\mathbf{W}}_i \mathbf{Q}_i' + \bar{\mathbf{b}}_i) \quad (14)$$

其中, $\bar{\mathbf{P}}_u$ 和 $\bar{\mathbf{Q}}_i$ 分别表示用户和物品的偏置向量,用来获取用户和物品的偏置信息。 $\bar{\mathbf{W}}_u$ 和 $\bar{\mathbf{W}}_i$ 为偏置层的权重矩阵, $\bar{\mathbf{b}}_u$ 和 $\bar{\mathbf{b}}_i$ 为偏置层的偏置项。

3.5 损失函数

网络最终输出的用户和物品的距离特征向量 \mathbf{P}_u 和 \mathbf{Q}_i , 为多层感知器层和偏置层输出的和,即:

$$\mathbf{P}_u = \mathbf{P}_u^* + \bar{\mathbf{P}}_u \quad (15)$$

$$\mathbf{Q}_i = \mathbf{Q}_i^* + \bar{\mathbf{Q}}_i \quad (16)$$

用户与物品之间预测的距离值根据式(3)可得:

$$\mathbf{D}_{ui}' = \|\mathbf{P}_u - \mathbf{Q}_i\|_2^2 + \mathbf{b}_u + \mathbf{b}_i + \mu \quad (17)$$

将损失函数定义为:

$$L = \sum_{(u,i) \in N} c_{ui} (\mathbf{D}_{ui} - \mathbf{D}_{ui}') \quad (18)$$

其中, c_{ui} 为置信度。

$$c_{ui} = 1 + \alpha \cdot t_{ui} \quad (19)$$

其中, t_{ui} 表示用户 u 与物品 i 交互的次数。 α 用来控制置信度的大小,在实验中, α 取值为 0.1。

为了防止过拟合现象,在计算预测距离值时,从 k 维的距离特征向量 \mathbf{P}_u 和 \mathbf{Q}_i 中随机丢弃一部分维度,例如:

$$\|\mathbf{P}_u - \mathbf{Q}_i\|_2^2 = |\mathbf{P}_{u1} - \mathbf{Q}_{i1}|^2 + \underbrace{|\mathbf{P}_{u2} - \mathbf{Q}_{i2}|^2}_{\text{drop}} + \underbrace{|\mathbf{P}_{u3} - \mathbf{Q}_{i3}|^2 + \dots + |\mathbf{P}_{uk} - \mathbf{Q}_{ik}|^2}_{\text{drop}} \quad (20)$$

式(20)随机丢弃了第3维的特征值。模型按一定的丢弃率随机选择丢弃某些维度的特征值。

4 实验与分析

本文提出的算法基于 TensorFlow 2.0 搭建深度网络结构,在 Windows 10 64 位操作系统, Inter Core i5 CPU, 16 GB

内存, Python 3.5 的环境下进行。

4.1 数据集

本文选取 MovieLens 100K、MovieLens 1M、mazon、微博这4个具有代表性的数据集进行性能实验,涵盖了电影、电商和文娱领域,兼顾了不同数量级和稀疏度的数据,体现了算法的普适性。其中, MovieLens 100K, MovieLens 1M, Amazon 为公共数据集,而微博数据集为本文自行爬取的新冠疫情期间微博相关热搜话题的所有微博,用户在不同的热搜话题下发表微博,即被视为用户与该热搜话题有交互,用户与话题的交互次数为用户在该话题下发表的微博总数。模型会根据用户历史交互的热搜话题,预测该用户可能感兴趣的其他热搜话题。用户 i 对话题 j 的兴趣度可用下式来衡量:

$$\mathbf{Q}_{ij} = \frac{N}{N_{\max}} \quad (21)$$

其中, N 表示用户 i 在热搜话题 j 下发表的博文总数, N_{\max} 表示在话题 j 下发表微博总数最多的用户发表的微博总数。兴趣度的取值范围为 $[0, 1]$ 。

4个数据集的具体参数如表2所列。

表2 数据集详细统计信息

Table 2 Detailed statistics of datasets

Datasets	Users	Items	Ratings	Density/%
ML-100K	943	1682	10000	6.305
ML-1M	6040	3706	1000209	4.468
Amazon	2292	450	2852	0.277
Weibo	5946	23	7375	5.393

4.2 评价指标

为了全面考量模型的推荐性能,采用准确率(Precision, 其中 Precision@10 和 Precision@5 分别表示推荐列表前10个和前5个物品的准确率)、召回率(Recall, 其中 Recall@10 和 Recall@5 分别表示前10个和前5个物品的召回率)、平均精度均值(MAP)、平均倒排序值(MRR)、归一化折损累计增益(NDCG)7个评价指标对本文模型和对比模型进行评价。将所有数据集都按 8:2 划分,随机抽取 80% 作为训练集,剩下 20% 作为测试集。

记 $S(u)$ 为模型生成的推荐列表, $T(u)$ 为用户真实的物品交互列表,则上述 Precision, Recall, MAP, MRR, NDCG 分别按下式计算:

$$Precision = \frac{1}{M_u \in N} \sum \frac{S(u) \cap T(u)}{S(u)} \quad (21)$$

$$Recall = \frac{1}{M_u \in N} \sum \frac{S(u) \cap T(u)}{T(u)} \quad (22)$$

$$MAP = \frac{1}{M} \sum_{i=1}^M AP_i \quad (23)$$

$$AP = \int_0^1 P(r) dr \quad (24)$$

$$MRR = \frac{1}{M} \sum_{i=0}^M \frac{1}{rank(i)} \quad (25)$$

$$NDCG = \frac{1}{M_u \in N} \sum \frac{DCG(u)}{IDCG(u)} \quad (25)$$

$$DCG(u) = \sum_{i=0}^k \frac{2^{rel(i)} - 1}{\log_2(i+1)}$$

其中, M 表示用户总数, k 为推荐列表中的物品总数。

4.3 参数设置

实验的主要参数包括网络层数、特征维度、丢弃率、学习

率,不同数据集的参数设置如表 3 所列。

表 3 不同数据集的参数设置

Table 3 Hyper-parameter settings for different datasets

数据集	网络层数	特征维度	丢弃率	学习率
ML-100K	3	350	0.05	0.0005
ML-1M	2	400	0.05	0.0001
Amazon	3	150	0.05	0.0005
Weibo	3	250	0.03	0.0008

表 4 MovieLens 100K 数据集实验结果的对比

Table 4 Experimental results of different models on MovieLens 100K

	<i>Precision@10</i>	<i>Recall@10</i>	<i>Precision@5</i>	<i>Recall@5</i>	<i>MAP</i>	<i>MRR</i>	<i>NDCG</i>
CDAE	0.191	0.123	0.221	0.074	0.141	0.43	0.486
CML	0.064	0.03	0.062	0.012	0.054	0.201	0.366
GMF	0.193	0.095	0.215	0.052	0.133	0.439	0.472
JRL	0.063	0.021	0.061	0.012	0.054	0.209	0.366
LRML	0.222	0.151	0.256	0.088	0.176	0.505	0.517
NeuMF	0.184	0.085	0.211	0.048	0.128	0.431	0.468
MetricF	0.206	0.123	0.234	0.075	0.143	0.452	0.488
本文算法	0.238	0.142	0.283	0.087	0.172	0.518	0.521

表 5 MovieLens 1M 数据集实验结果的对比

Table 5 Experimental results of different models on MovieLens 1M

	<i>Precision@10</i>	<i>Recall@10</i>	<i>Precision@5</i>	<i>Recall@5</i>	<i>MAP</i>	<i>MRR</i>	<i>NDCG</i>
CDAE	0.025	0.007	0.031	0.005	0.016	0.121	0.305
CML	0.024	0.007	0.03	0.005	0.016	0.118	0.305
GMF	0.184	0.066	0.214	0.04	0.101	0.387	0.455
JRL	0.024	0.007	0.030	0.04	0.016	0.115	0.305
LRML	0.220	0.086	0.251	0.050	0.134	0.457	0.497
NeuMF	0.183	0.066	0.205	0.038	0.100	0.386	0.454
MetricF	0.179	0.066	0.196	0.037	0.096	0.366	0.451
本文算法	0.223	0.088	0.252	0.051	0.142	0.464	0.499

表 6 Amazon 数据集实验结果的对比

Table 6 Experimental results of different models on Amazon

	<i>Precision@10</i>	<i>Recall@10</i>	<i>Precision@5</i>	<i>Recall@5</i>	<i>MAP</i>	<i>MRR</i>	<i>NDCG</i>
CDAE	0.026	0.264	0.034	0.168	0.125	0.125	0.250
CML	0.003	0.033	0.004	0.021	0.020	0.020	0.15
GMF	0.011	0.11	0.011	0.056	0.045	0.045	0.179
JRL	0.005	0.047	0.007	0.033	0.026	0.026	0.157
LRML	0.024	0.243	0.031	0.154	0.105	0.105	0.251
NeuMF	0.014	0.136	0.018	0.089	0.061	0.061	0.192
MetricF	0.016	0.156	0.025	0.124	0.086	0.086	0.219
本文算法	0.031	0.273	0.037	0.183	0.126	0.126	0.257

表 7 微博数据集实验结果的对比

Table 7 Experimental results of different models on Weibo

	<i>Precision@10</i>	<i>Recall@10</i>	<i>Precision@5</i>	<i>Recall@5</i>	<i>MAP</i>	<i>MRR</i>	<i>NDCG</i>
CDAE	0.051	0.483	0.060	0.282	0.203	0.206	0.371
CML	0.055	0.516	0.047	0.225	0.177	0.178	0.351
GMF	0.061	0.564	0.061	0.279	0.214	0.217	0.383
JRL	0.056	0.528	0.048	0.227	0.184	0.185	0.357
LRML	0.072	0.685	0.086	0.412	0.259	0.261	0.422
NeuMF	0.058	0.551	0.059	0.279	0.212	0.216	0.381
MetricF	0.075	0.710	0.090	0.424	0.275	0.280	0.436
本文算法	0.078	0.744	0.093	0.434	0.281	0.285	0.442

从表中加粗的数据可以看出,本文提出的模型在 4 个数据集上均优于其他深度推荐模型和距离分解模型。LRML 模型和 MetricF 模型次优,LRML 的结果说明了深度学习对于潜在特征向量的训练效果明显,获取的特征向量具有较强的表征能力,而 MetricF 模型的优越性证明了距离分解比传统矩阵分解的推荐性能更好,预测距离值一定程度上能够解

模型在 4 个数据集上均采用梯度下降 (gradient descent) 优化器,批量大小 (batch size) 为 1024,激活函数为 relu 函数,且所有模型均不使用预训练。

4.4 实验结果与分析

本文选取 CDAE, CML, GMF, JRL, LRML, NeuMF 和 MetricF 算法作为对比模型。所有模型在 4 个数据集上的实验结果如表 4—表 7 所列。

决交互矩阵稀疏的问题。本文模型结合了这两个模型的长处,表现出了更好的推荐效果,且在不同领域、不同数量级、不同稀疏度的数据集上均能保持稳定的实验结果,良好地反映了模型的通用性。而 CDAE 模型只在 Amazon 数据集上有较为突出的推荐效果,虽然证明了该模型对于稀疏矩阵特征提取的效果较好,但同时也反映了模型的不稳定性。NeuMF 在

4 个数据集上的物品排序效果都较为稳定,但性能不如本文模型,且在训练过程中时间复杂度较高。此外,本文模型只依赖用户的评分值,不需要其他辅助信息,就能够提高模型提取特征的效率,同时也使模型具备了良好的可扩展性。

为了深入探究本文模型中对物品排序结果起关键性作用的因素,在 MovieLens 100K 数据集上,进一步对深度网络是否增加偏置层、网络层数、特征维数,以及激活函数进行了性能分析实验。

(1) 偏置层性能实验

如图 3 所示,黑色柱形表示增加了偏置层的网络,其实验效果相较于不加偏置层,在评价指标上提高了至少 7.2%,有力地证明了偏置层能够有效获取用户和物品中隐藏的偏置信息,其对于整个网络训练距离特征向量具有重要作用。

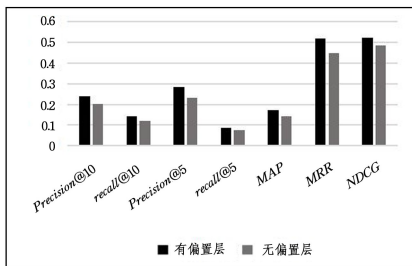


图 3 有无偏置层实验的对比

Fig. 3 Performance of model with/without biased layer

(2) 不同网络层数性能比较实验

对网络层数分别为 2, 3, 4 进行实验,如图 4 所示,当网络层数为 3 时,生成的物品推荐列表的准确度和相关度最高,超过 3 层后开始出现过拟合现象,模型性能有所下降。

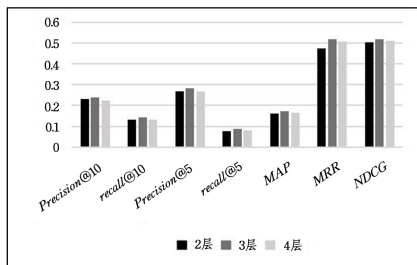


图 4 网络层数实验的对比

Fig. 4 Performance of model with different number of deep layers

(3) 不同特征维数性能比较实验

特征维度在 250 到 400 之间,以 50 为步长递增,从图 5 中可以看出,特征维数为 350 时,模型的实验结果达到最优,维数超过 350 后效果开始下降。

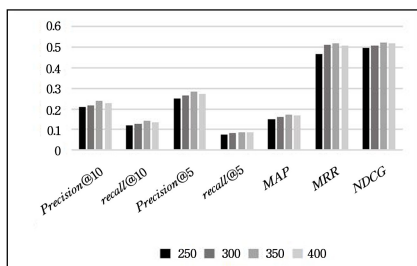


图 5 特征维度实验的对比

Fig. 5 Performance of model with different number of factors

(4) 不同激活函数性能比较实验

选取 4 个经典的激活函数 relu, sigmoid, softplus 和 tanh, 以及最新提出的深度学习激活函数 mish^[23] 进行性能对比实验。其中, mish 是最新研究出的激活函数,其函数图像向上无边界的特性避免了因封顶导致的饱和,并且允许比较小的负梯度流入。如图 6 所示, relu 的实验结果较为突出, sigmoid, softplus, tanh 和 mish 的结果差不多,容易出现梯度消失现象。

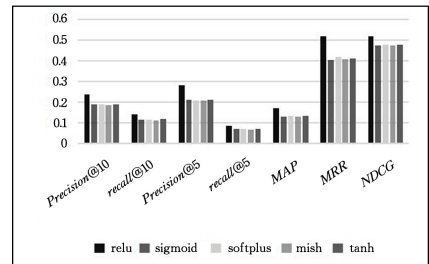


图 6 不同激活函数实验的对比

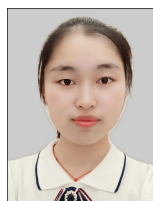
Fig. 6 Performance of model with different activation functions

结束语 在 Top-N 推荐任务中,本文基于距离分解模型,结合深度学习,并通过加入偏置层有效改进了深度网络模型,解决了传统矩阵分解不满足三角不等式的问题,同时利用改进后的深度网络训练出了表征能力更强的距离特征向量。本文结合了线性特征和非线性特征的距离特征向量,有效地在不同领域和稀疏度的数据集上,生成了优于其他深度推荐模型和距离分解模型的推荐列表,证明了本文算法的普适性和可扩展性。考虑到数量级较高的数据集需要保证一定的训练效率,而本文算法在较大的数据集上需要消耗的计算资源也相应地增多。未来,本文会在提升系统性能的同时,致力于降低模型的计算复杂度,研究出整体性能更高的推荐算法。

参考文献

- [1] ZHANG S, YAO L, HUANG C, et al. Metric Factorization: Recommendation beyond Matrix Factorization[EB/OL]. [2018-06-04]. <http://xxx.itp.ac.cn/pdf/1802.04606v2>.
- [2] DOU Z Y, WANG X, SHI S M, et al. Exploiting deep representations for natural language processing[J]. Neurocomputing, 2020, 386: 1-7.
- [3] LI B, ZHAO J J, FU H. DLT-Net: deep learning transmittance network for single image haze removal[J]. Signal, Image and Video Processing, 2020, 14(6): 1245-1253.
- [4] XIA Y. Research on statistical machine translation model based on deep neural network[J]. Computing, 2020, 102(3): 643-661.
- [5] CHENG H T, KOC L, HARMSEN J, et al. Wide & Deep Learning for Recommender Systems[EB/OL]. [2016-06-24]. <https://arxiv.org/pdf/1606.07792.pdf>.
- [6] YANG C, MIAO L H, JIANG B, et al. Gated and attentive neural collaborative filtering for user generated list recommendation[J]. Knowledge-Based System, 2020, 187(8): 78252-78264.
- [7] ZHANG S, YAO L, WU B, et al. Unraveling Metric Vector Spaces with Factorization for Recommendation[J]. IEEE Transactions on Industrial Informatics, 2020, 16(2): 732-742.
- [8] DAI H, WANG L J, QIN J W. Metric Factorization with Item

- Cooccurrence for Recommendation[J]. *Symmetry*, 2020, 12(4): 512.
- [9] DZIUGAITE G K, DANIEL M R. Neural Network Matrix Factorization [EB/OL]. [2015-12-15]. <https://arxiv.org/abs/1511.06443>.
- [10] SEDHAIN S, MENON A K, SANNER S, et al. AutoRec: Autoencoders Meet Collaborative Filtering[C]// *International Conference on World Wide Web*. 2015:111-112.
- [11] LI P J, WANG Z H, REN Z C, et al. Neural rating regression with abstractive tips generation for recommendation[C]// *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2017:345-354.
- [12] HE X N, 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*. 2017:355-364.
- [13] XIAO J, YE H, HE X N, 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*. 2017:3119-3125.
- [14] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]// *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. 2009:452-461.
- [15] HSIEH C K, YANG L Q, CUI Y, et al. Collaborative metric learning[C]// *Proceedings of the 26th International Conference on World Wide Web*. 2017:193-201.
- [16] WU Y, DUBOIS C, ZHENG A X, et al. Collaborative denoising auto-encoders for top-n recommender systems[C]// *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 2016:153-162.
- [17] HE X N, LIAO L Z, ZHANG H W, et al. Neural Collaborative Filtering[C]// *Proceedings of the 26th International Conference on World Wide Web*. 2017:173-182.
- [18] TAY Y, TUAN L A, HUI S C. Latent relational metric learning via memory-based attention for collaborative ranking[C]// *Proceedings of the 27th International Conference on World Wide Web*. 2018:729-739.
- [19] FENG S S, LI X, ZENG Y F, et al. Personalized ranking metric embedding for next new poi recommendation[C]// *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 2015:2069-2075.
- [20] LIANG D, ALTOSAAR J, CHARLIN L, et al. Factorization Meets the Item Embedding; Regularizing Matrix Factorization with Item Co-occurrence[C]// *Proceedings of the 10th ACM Conference on Recommender Systems*. 2016:59-66.
- [21] TANG J X, WANG K. Personalized top-n sequential recommendation via convolutional sequence embedding[C]// *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 2018:565-573.
- [22] ZHANG S, TAY Y, YAO L, et al. Next Item Recommendation with Self-Attentive Metric Learning[C]// *Proceedings of the 33th AAAI Conference on Artificial Intelligence*. 2019:9-19.
- [23] MISRA D. Mish: A Self Regularized Non-Monotonic Neural Activation Function[EB/OL]. [2020-08-13]. <https://arxiv.org/abs/1908.08681>.



QIAN Meng-wei, born in 1996, post-graduate. Her main research interests include data mining and recommender system.



GUO Yi, born in 1975, Ph.D, professor. His main research interests include text mining, knowledge discovery and business intelligence.